

Fachhochschule
FH JOANNEUM Gesellschaft mbH

iPhone Development Tools
Analyse von Development Tools für das Apple iPhone

Bachelorarbeit 1
zur Erlangung des akademischen Grades
„Bachelor of Science in Engineering“

eingereicht am
Studiengang Informationsmanagement

Betreuer: Dipl.-Ing. (FH) Robert Strohmaier

eingereicht von: Mario Ranftl
Personenkennzahl: 0810422026

27.09.2010

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Graz, 27. September 2010

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Tabellenverzeichnis	7
Abkürzungsverzeichnis	9
Kurzfassung	10
Abstract.....	11
1. Einleitung.....	12
1.1. Ziel der Bachelorarbeit.....	12
1.2. Aufbau der Arbeit.....	13
2. Methoden der Software Evaluierung	14
2.1. Grundlagen der Evaluierung	14
2.2. Angewandte Evaluierung	15
2.2.1. Festlegen der Evaluierungsobjekte	15
2.2.2. Formulieren des Evaluierungsziels.....	16
2.2.3. Ableiten der Evaluierungskriterien.....	17
2.2.4. Kriterienkatalog.....	17
2.2.4.1. Allgemein	17
2.2.4.2. Code-Editor.....	18
2.2.4.3. User Interface Desinger	19
2.2.4.4. IDE Funktionalität.....	20
2.2.5. Gewichten der Evaluierungskriterien	21
2.2.6. Abbilden der Evaluierungskriterien in Messgrößen	21
2.2.7. Auswählen von Messmethoden.....	22
2.2.8. Messen der Kriterienerträge und Organisation des Evaluierungsprozesses.....	22

3. Evaluierung.....	23
3.1. Apple iOS SDK	23
3.1.1. Beurteilung nach Kriterienkatalog.....	24
3.1.1.1. Allgemein	25
3.1.1.2. Code-Editor.....	26
3.1.1.3. User Interface Designer	27
3.1.1.4. IDE Funktionalität.....	28
3.1.2. Fazit.....	29
3.2. Adobe Flash CS5 mit Adobe Packager for iPhone.....	29
3.2.1. Beurteilung nach Kriterienkatalog.....	31
3.2.1.1. Allgemein	31
3.2.1.2. Code-Editor.....	32
3.2.1.3. User Interface Designer	33
3.2.1.4. IDE Funktionalität.....	34
3.2.2. Fazit.....	35
3.3. Unity iPhone.....	35
3.3.1. Beurteilung nach Kriterienkatalog.....	37
3.3.1.1. Allgemein	37
3.3.1.2. Code-Editor.....	39
3.3.1.3. User Interface Designer	39
3.3.1.4. IDE Funktionalität.....	41
3.3.2. Fazit.....	42
3.4. GameSalad	42
3.4.1. Beurteilung nach Kriterienkatalog.....	44
3.4.1.1. Allgemein	44
3.4.1.2. Code-Editor.....	45

3.4.1.3. User Interface Designer	45
3.4.1.4. IDE Funktionalität.....	46
3.4.2. Fazit.....	47
3.5. PhoneGap.....	47
3.5.1. Beurteilung nach Kriterienkatalog.....	48
3.5.1.1. Allgemein	48
3.5.1.2. Code-Editor.....	49
3.5.1.3. User Interface Designer	49
3.5.1.4. IDE Funktionalität.....	49
3.5.2. Fazit.....	50
4. Zusammenfassung	51
4.1. Übersichtstabelle.....	51
4.2. Ergebnisse und Ausblick.....	53
Literaturverzeichnis	55

Abbildungsverzeichnis

Abbildung 1: iOS SDK Komponenten.....	24
Abbildung 2: Flash CS5 Entwicklungsumgebung.....	29
Abbildung 3: Unity iPhone Entwicklungsumgebung	36
Abbildung 4: GameSalad Entwicklungsumgebung.....	43

Tabellenverzeichnis

Tabelle 1: Allgemeine Kriterien	18
Tabelle 2: Editor Kriterien (vgl. [Wihsböck 2007], S.6f.)	19
Tabelle 3: User Interface Designer Kriterien	20
Tabelle 4: IDE Funktionalität Kriterien	21
Tabelle 5: iOS SDK Basisinformationen.....	24
Tabelle 6: iOS SDK Allgemein	26
Tabelle 7: iOS SDK Code-Editor	27
Tabelle 8: iOS SDK User Interface Designer	28
Tabelle 9: iOS SDK IDE Funktionalität.....	28
Tabelle 10: Flash CS5 Basisinformationen	30
Tabelle 11: Flash CS5 Allgemein	32
Tabelle 12: Flash CS5 Code-Editor.....	33
Tabelle 13: Flash CS5 User Interface Designer	34
Tabelle 14: Flash CS5 IDE Funktionalität	35
Tabelle 15: Unity iPhone Basisinformationen.....	37
Tabelle 16: Unity iPhone Allgemein	38
Tabelle 17: Unity iPhone Code-Editor	39
Tabelle 18: Unity iPhone User Interface Designer	40
Tabelle 19: Unity iPhone IDE Funktionalität	41
Tabelle 20: GameSalad Basisinformationen	43
Tabelle 21: GameSalad Allgemein	45
Tabelle 22: GameSalad User Interface Designer.....	46
Tabelle 23: GameSalad IDE Funktionalität	47

Tabelle 24: PhoneGap Basisinformationen	48
Tabelle 25: PhoneGap Allgemein.....	49
Tabelle 26: PhoneGap IDE Funktionalität	50
Tabelle 27: Übersichtstabelle	53

Abkürzungsverzeichnis

App	Applikation, Programm (meist für mobile Endgeräte)
GUI	Graphical User Interface (grafische Benutzeroberfläche)
IDE	Integrated Development Environment (Entwicklungsumgebung)
RTE	Runtime Environment (Laufzeitumgebung)
SDK	Software Development Kit
VPL	Visual Programming Language
WYSIWYG	What You See Is What You Get

Kurzfassung

In einer Evaluierung werden bestimmte Objekte zielgerichtet und auf der Grundlage eines Systems von Kriterien mit festgelegten Messmethoden beurteilt. Bereits bei der Auswahl der Objekte muss nach fixierten Vorgaben gehandelt werden.

In dieser Bachelorarbeit wird der Begriff Evaluierung definiert und erläutert. In mehreren Phasen wird methodisch vorgegangen um ein Modell aufzustellen, mit dem es möglich ist die wichtigsten Aspekte einer Entwicklungsumgebung einzufangen und diese schlussendlich zu vergleichen.

Ziel dieser Bachelorarbeit ist es, die Stärken und Schwächen ausgewählter Entwicklungsumgebungen, die auf die Entwicklung für das Apple iPhone ausgelegt sind, aufzuzeigen. Es sollten alternative Entwicklungstools zu dem von Apple bereitgestellten Entwicklungskomponenten Xcode und iOS SDK gefunden werden und eine strukturierte Übersicht aufbereitet werden.

Es konnte dabei festgestellt werden, dass die alternativen Entwicklungsumgebungen, zur Entwicklung von bestimmten Projekten, bereits besser geeignet sind als die Apple eigenen Entwicklungstools.

Abstract

An evaluation is a systematic determination of defined objects based on criteria with fixed measuring methods. The selection of objects has to be done with specified guidelines.

This thesis defines the term evaluation and illustrates it. There are multiple phases which have to be methodically completed, to set up a model, that is capable of covering all important facts of an integrated development environment and finally compare them with each other.

The aim of this work is to determine the advantages and disadvantages of selected IDEs, that are able to develop applications for the Apple iPhone. Alternative development-tools should be found and a structured overview should be prepared.

It could be identified that some alternative IDEs are currently better for development of specific type of projects than Apple Xcode and iPhone SDK.

1. Einleitung

Mobile Applikationen (Apps) haben durch den Erfolg des Apple iPhone und der von Apple eingeführten und in Apple iTunes integrierten Verkaufsplattform App Store, an Bedeutung gewonnen. Seit Anfang des Jahres 2010 befinden sich bereits mehr als 140.000 iPhone-Apps im Apple App Store mit rasant steigender Tendenz (vgl. [Franke 2010], S.39).

Der Großteil der Software im App Store stammt von Drittentwicklern. Apple stellt für die Entwicklung von Apps die Entwicklungsumgebung XCode in Verbindung mit dem iOS SDK bereit. Neben dieser Entwicklungsumgebung gibt es jedoch auch zahlreiche andere Tools die speziell für die iPhone Entwicklung ausgelegt sind, beziehungsweise die Möglichkeit für das iPhone zu entwickeln in ihr Produktportfolio aufgenommen haben. Angehende iPhone-EntwicklerInnen haben dadurch die Chance sich über die Restriktionen von Apple, wie die vorgeschriebene Programmiersprache Objective-C und der Apple-Betriebssystem-Pflicht (Mac-OS X) hinwegzusetzen. Allerdings bedeutet der Einsatz einer alternativen Entwicklungsumgebung eventuell auch den Verlust der Möglichkeit, die Software über den Apple eigenen App Store zu vertreiben.

1.1. Ziel der Bachelorarbeit

Ziel dieser Bachelorarbeit ist es, einen strukturierten Überblick der zum Zeitpunkt der Arbeit verfügbaren Entwicklungstools zu geben und die Stärken und Schwächen ausgewählter Entwicklungsumgebungen untereinander und im Vergleich zu den von Apple bereitgestellten Entwicklungstools aufzuzeigen.

Es ist nicht Ziel dieser Arbeit alle verfügbaren Entwicklungsumgebungen und Möglichkeiten einzufangen.

1.2. Aufbau der Arbeit

Die Arbeit gliedert sich in drei Teile. Zu Beginn der Arbeit wird der Begriff Evaluierung definiert und erläutert. Für die Bewertung der Entwicklungsumgebungen wird ein Evaluierungsmodell aufgestellt, durch welches es möglich ist, die wichtigsten Aspekte einzufangen und zu beurteilen. Dazu werden systematisch diverse Phasen abgearbeitet, die sowohl in der Theorie definiert, als auch in die Praxis, in Bezug auf diese Evaluierung, umgesetzt werden. Dies umfasst z. B. die Definition eines Kriterienkatalogs.

Im zweiten Teil werden die einzelnen Entwicklungsumgebungen vorgestellt und anschließend nach den definierten Kriterien beurteilt. Ein besonderes Augenmerk wird dabei auf die Möglichkeit der Programmierung mittels Adobe Flash CS5 geworfen. Den Abschluss jeder einzelnen Evaluierung bildet eine Zusammenfassung der beobachteten Stärken und Schwächen.

Der abschließende dritte Teil stellt die Ergebnisse der Evaluierung in einer Übersichtstabelle dar. Außerdem werden Schlussfolgerungen gezogen und ein Ausblick auf die Entwicklung in der Zukunft gegeben.

2. Methoden der Software Evaluierung

Um die Entwicklungsumgebungen zu beurteilen, ist es essentiell, gezielt und methodisch vorzugehen. Daher ist es sinnvoll die Grundlagen des Begriffes Evaluierung zu beleuchten und ein Modell zu definieren.

2.1. Grundlagen der Evaluierung

"Evaluierung oder Evaluation heißt, bestimmte Objekte auf der Grundlage eines Systems von Kriterien zielbezogen zu beurteilen." ([Lutz 2009], S.392)

Daraus folgt, dass zur Beurteilung der Entwicklungsumgebungen erst ein Modell aufgestellt werden muss, durch welches es möglich ist, die wichtigsten Aspekte einzufangen.

Es ist außerdem wichtig festzuhalten, dass die Ziele bei einer Evaluierung von denen einer Bewertung differieren. Eine Bewertung verfolgt das Ziel den Geldwert bzw. die Kosten von Produkten zu ermitteln. Eine Evaluierung behandelt die Beurteilung von Objekten nach beliebigen systematisch festgelegten Zielen.

Lutz führt dabei nach der Norm EN-1325-1 (Wertanalyse-Arbeitsplan) aus, dass eine Evaluierung nach einem bestimmten und dokumentierten Arbeitsplan zu erfolgen hat, der sich in mehrere Phasen gliedert (vgl. [Lutz 2009], S.393 f. zitiert nach [EN-1325-1 1996]):

1. *Festlegen der Evaluierungsobjekte*
2. *Formulieren des Evaluierungsziels*
3. *Ableiten der Evaluierungskriterien*
4. *Gewichten der Evaluierungskriterien*
5. *Abilden der Evaluierungskriterien in Messgrößen*
6. *Auswählen von Messmethoden*
7. *Messen der Kriterienerträge und Organisation des Evaluierungsprozesses*

2.2. Angewandte Evaluierung

Im Folgenden wird der definierte Arbeitsplan praktisch angewandt, um ein Modell abzuleiten, das für die Gewinnung der Informationen verwendet werden kann. Außerdem ergänzt ein Kriterienkatalog die Phase der Bestimmung der Evaluierungskriterien.

2.2.1. Festlegen der Evaluierungsobjekte

In der ersten Phase werden die Evaluierungsobjekte festgelegt. Dies wird als eine Art der Vorauswahl bezeichnet, da nur Alternativen die die festgelegten Kriterien erfüllen, in die Evaluierung einbezogen werden (vgl. [Stamelos 2000], S.4).

Lutz beschreibt den Vorgang der Vorauswahl in drei Phasen innerhalb der, vom Wertanalyse-Arbeitsplan abgeleiteten, Phase Organisation des Evaluierungsprozesses:

1. Alternativensuche
2. Grobauswahl
3. Optimumbestimmung

In der Alternativensuche werden Informationen über potentielle Evaluierungsobjekte eingeholt um K.o.-Kriterien zu definieren. Danach werden in der Grobauswahl die vorher definierten K.o.-Kriterien angewandt, um die Evaluierungsobjekte festzulegen. Die dritte Phase Optimumbestimmung setzt das Vorhandensein eines Kriterienkatalogs voraus und behandelt das Bestimmen des optimalen Evaluierungsobjekts durch Beurteilung der gemessenen Kriterienerträge. Diese Phase erfolgt am Schluss einer Evaluierung und bestimmt das Evaluierungsergebnis (vgl. [Lutz 2009], S.399f.).

Folgende K.o.-Kriterien beeinflussen die Grobauswahl:

- Die Entwicklung von Projekten für das Apple iPhone muss von der IDE (oder dem SDK) unterstützt werden.
- IDEs die auf VPL (Visual Programming Language) basieren und keine Möglichkeit zum Hinzufügen von eigener Funktionalität besitzen,

werden nicht in die Vorauswahl einbezogen, da die Funktionalität der daraus resultierenden Apps stark eingeschränkt ist.

- Bei annähernd gleichwertigen IDEs werden jene mit einer höheren Verbreitung bevorzugt.
- Um die Vielschichtigkeit der Evaluierung zu gewährleisten, wird:
 - mindestens eine IDE bzw. ein SDK evaluiert, die auf einer Open-Source Lizenz (Quellcode der IDE darf frei eingesehen und verändert werden) basiert.
 - mindestens eine IDE bzw. ein SDK evaluiert, die für Cross-Plattform Entwicklung (für mehrere Zielsysteme gleichzeitig) genutzt werden kann.
 - mindestens eine IDE bzw. ein SDK evaluiert, die auf einer Game-Engine (Spezialisierung auf Spiele-Entwicklung) basiert.
 - mindestens eine IDE bzw. ein SDK evaluiert, die für Rapid Prototyping (schnelle Erstellung eines Prototypen) geeignet ist.

Die Evaluation behandelt folgende Evaluierungsobjekte:

- Apple iOS SDK 4.1 mit Xcode 3.2.4
- Adobe Flash CS5 mit Adobe Packager for iPhone v1
- Unity iPhone 1.7
- GameSalad 0.8.9
- PhoneGap 0.9.1 mit Xcode 3.2.4

2.2.2. Formulieren des Evaluierungsziels

Laut Lutz folgt das Evaluierungsziel aus dem geplanten Verwendungszweck des Resultats der Evaluation (vgl. [Lutz 2009], S.394). Außerdem definiert das Resultat den Typ der Evaluierung. Es wird dabei zwischen verschiedenen rangbasierenden und beschreibenden Evaluierungen unterschieden. Da das geplante Ergebnis einen formalen, nicht bewerteten Überblick über die verschiedenen IDEs gibt, handelt es sich um eine beschreibende Evaluierung (vgl. [Stamelos 2000], S.4).

Infolgedessen lässt sich das Ziel bzw. der Zweck dieser Evaluierung definieren: Das Evaluierungsziel ist die Ermittlung der Vor- und Nachteile der

einzelnen IDEs untereinander und im Vergleich zum Apple iOS SDK, mit dem Hauptfokus einen Überblick bereitzustellen.

2.2.3. Ableiten der Evaluierungskriterien

"Evaluierungskriterien sind Eigenschaften eines Evaluierungsobjekts, die unter Berücksichtigung des Evaluierungsziels aus diesem abgeleitet werden." ([Lutz 2009], S.394) Da eine strukturierte Übersicht das Hauptziel dieser Evaluierung ist, werden die Kriterien in vier Hauptkriterien eingeteilt:

1. Allgemein
2. Code-Editor
3. User Interface Designer
4. IDE Funktionalität

Diese Kriterien formen die Struktur für weitere Sub-Kriterien und werden als "compound attributes" bezeichnet (vgl. [Stamelos 2000], S.4).

2.2.4. Kriterienkatalog

Der folgende Katalog umfasst die vier definierten Hauptkriterien mit ihren Sub-Kriterien, nach denen die Entwicklungsumgebungen evaluiert werden.

2.2.4.1. Allgemein

Die folgenden Kriterien beziehen sich auf allgemeine Themen und sind in Tabelle 1 abgebildet.

App Store Publishing	Darunter wird Möglichkeit verstanden, ein Projekt im Apple App Store zu veröffentlichen. Durch kürzlich veränderte Lizenzbestimmungen des Apple App Store, gibt es keine Einschränkungen bezüglich der verwendeten Programmiersprachen oder Werkzeuge, solange kein Programmcode zur Laufzeit nachgeladen und interpretiert wird (vgl. [http://www.apple.com][1]).
Community / Kundensupport	Kundensupport bezeichnet die Möglichkeit freie oder kostenpflichtige Kundenbetreuung durch den Hersteller zu beanspruchen. Community wird gewichtet durch die

	Anzahl der Nutzer der durch den Hersteller zur Verfügung gestellten Austauschplattform.
Dokumentation	Eine Programmiersprache ist nur so gut wie ihre Dokumentation. Bereitgestellte Tutorials helfen beim Erlernen einer Programmiersprache. Dadurch behandelt dieses Kriterium auch gleichermaßen die Einsteigerfreundlichkeit.
Licensing / Kosten	Dieses Kriterium behandelt die Kosten, die beim Kauf der IDE anfallen bzw. als Lizenzgebühren bei der Verwendung oder bei der Veröffentlichung einer App geleistet werden müssen.

Tabelle 1: Allgemeine Kriterien

2.2.4.2. Code-Editor

Das Hauptkriterium Code-Editor befasst sich mit der Funktionalität des in der IDE eingesetzten Codeeditors. Die untersuchten Kriterien sind in Tabelle 2 abgebildet.

Quellcodeformatierung	<i>"Der Quellcode wird automatisch nach bestimmten Regeln, wie z.B. konsistentes Einrücken bei Blöcken oder Schleifen, formatiert."</i> ([Wihsböck 2007], S.6)
Code Hilfe	Der Editor unterstützt den User mit Informationen abhängig von der aktuellen Position des Cursors auf dem Code. Dies können z.B. Informationen zu den zulässigen Parametern einer Methode sein.
Code Completion	Stellt eine Liste von, an der aktuellen Cursorposition, möglichen Ausdrücken zur Verfügung.

RegEx-Suche	Durchsuchen und ändern von im Projekt befindlichen Codepassagen mittels regulären Ausdrücken.
Code Folding	Der Editor ist fähig zusammenhängende Blockpassagen zu erkennen und diese, falls gewünscht, einzuklappen.
ToDo-List & Bookmarks	Es ist möglich Bookmarks von Codeteilen zu verknüpfen und ToDo-Lists zu erstellen.

Tabelle 2: Editor Kriterien (vgl. [Wihsböck 2007], S.6f.)

2.2.4.3. User Interface Designer

Die Gestaltung einer grafischen Benutzeroberfläche GUI erfolgt im User Interface Designer, mit dessen Hilfe einzelne Komponenten intuitiv angeordnet und verändert werden können. Die Evaluierungskriterien sind in Tabelle 3 abgebildet.

Vordefinierte Elemente	Wichtige Elemente sind vordefiniert und müssen nicht durch Code erstellt werden.
Design erstellen / gestalten	Elemente können intuitiv angeordnet, gruppiert und manipuliert werden.
Automatische Codeerzeugung	Durch Erstellen und Ändern der Elemente im User Interface Designer werden automatisch die Codezeilen des Elements generiert und geschrieben.
Attribute via GUI änderbar	Im User Interface Designer können erweiterte Attribute eines Elements durch ein GUI angezeigt und manipuliert werden.
Attribute via Code änderbar	Änderungen am Code eines Elements sind möglich und werden im User

	Interface angezeigt.	Designer	automatisch
--	-------------------------	----------	-------------

Tabelle 3: User Interface Designer Kriterien

2.2.4.4. IDE Funktionalität

Das letzte Hauptkriterium befasst sich mit der Analyse des möglichen Funktionsumfangs bei der Entwicklung für das Apple iPhone und erweiterten IDE Funktionen wie z.B. Debugging und Rapid Prototyping.

Refactoring	Bezeichnet die Fähigkeit einer IDE signifikante, logisch miteinander zusammenhängende Codeteile dokumentenübergreifend zu manipulieren und dabei automatisch die Konsistenz des Projekts aufrecht zu erhalten (vgl. [Wihsböck 2007], S.9).
UML-Diagramm	Die IDE kann automatisch Klassendiagramme eines Projekts erstellen.
Emulation	Die Ausführung des Projekts ist innerhalb der IDE durch Emulation möglich.
Debugging	Die IDE kann Code ausführen und Programmabläufe eines Projekts während der Ausführung manipulieren.
Profiling	Die IDE kann Code direkt am Endgerät ausführen und Programmabläufe durch Manipulation innerhalb der IDE verändern.
Unterstützt alle iPhone Features	Die IDE unterstützt alle Funktionen des Apple iPhone wie z.B. Neigungssensor, digitaler Kompass und GPS-

Rapid Prototyping Eignung	<p>Lokalisierung. Außerdem können Services wie Apple Push Notification Service in die Projekte implementiert werden.</p> <p>Die IDE eignet sich zur Entwicklung von Prototypen mit denen in kurzer Zeit eine frühe eingeschränkte Funktionalität gezeigt werden kann.</p>
---------------------------	---

Tabelle 4: IDE Funktionalität Kriterien

2.2.5. Gewichten der Evaluierungskriterien

Um den Wert eines Evaluierungsobjekts zu ermitteln, bedarf es einer Gewichtung der Evaluierungskriterien. *"Es wird eine Präferenzordnung der Evaluierungskriterien hergestellt und verwendet."* ([Lutz 2009], S.395f.)

Das Evaluierungsziel dieser Evaluierung ist einen Überblick aufzuzeigen. Da ein beschreibender Evaluierungstyp verwendet wird, wird auf eine Gewichtung der Kriterien verzichtet.

2.2.6. Abbilden der Evaluierungskriterien in Messgrößen

*"Jedem Kriterium, das sich nicht in weitere Sub-Kriterien teilen lässt, muss eine Messgröße assoziiert werden. [...] Die Skala muss dabei mindestens ordinal skaliert sein [...] um ausdrücken zu können, welcher Wert von zwei Möglichkeiten bevorzugt wird."*¹ ([Stamelos 2000], S.5)

Im Folgenden wird für alle Evaluierungskriterien, deren zu messende Kriterienerträge nicht metrisch skaliert sind, eine Skala verwendet. Die Skala benutzt die +/-o-Notation, für die gilt:

- Das Symbol + bezeichnet einen Kriterienertrag eines Evaluierungsobjekts dessen angesprochenes Evaluierungskriterium komplett erfüllt wird.

¹ Übersetzung durch den Verfasser.

- Das Symbol o bezeichnet einen Kriterienertrag eines Evaluierungsobjekts dessen angesprochenes Evaluierungskriterium teils erfüllt wird.
- Das Symbol - bezeichnet einen Kriterienertrag eines Evaluierungsobjekts dessen angesprochenes Evaluierungskriterium nicht erfüllt wird.

Der Kriterienertrag des Evaluierungskriteriums Licensing / Kosten bildet die aufzubringenden Kosten in Geldeinheiten ab und ist metrisch skaliert. Je höher der gemessene Wert desto schlechter.

2.2.7. Auswählen von Messmethoden

Kriterienerträge können direkt wie z.B. durch Beobachten oder indirekt wie z.B. durch Befragen erfasst werden. Beobachten bedeutet dabei methodisches, selektives und systematisches Vorgehen zu einem bestimmten Evaluierungszweck (vgl. [Lutz 2009], S.397f.).

Diese Evaluierung setzt direktes Beobachten ein, um die Evaluierungsobjekte nach ihren Kriterien zu messen. Durch den Einsatz des definierten Kriterienkatalogs wird ein kontrolliertes Vorgehen erreicht.

2.2.8. Messen der Kriterienerträge und Organisation des Evaluierungsprozesses

Die Phase der Organisation des Evaluierungsprozesses bezeichnet den kompletten Prozess der Evaluierung und wurde bereits in Kapitel 2.2.1, aufgrund der praktischen Anwendung der Phasen Alternativensuche und Grobauswahl, behandelt.

In den folgenden Kapiteln werden die definierten Messmethoden am Evaluierungsobjekt angewandt. Diese letzte Phase wird Messen der Kriterienerträge genannt (vgl. [Lutz 2009], S.399).

Evaluationsdatum ist der 16.09.2010. Alle IDEs werden in der zu diesem Zeitpunkt verfügbaren, stabilen Version evaluiert. Als Testplattform kommen ein Apple MacBook Pro mit Apple Mac OS X 10.6.3 bzw. ein PC mit Microsoft Windows 7 zum Einsatz.

3. Evaluierung

Dieses Kapitel behandelt die praktische Anwendung des, im Kapitel 2 vorgestellten, Evaluierungsmodells an den Evaluierungsobjekten.

3.1. Apple iOS SDK

Das von Apple bereitgestellte iOS SDK ist das offizielle Werkzeug zur Entwicklung von Apps für das Apple iPhone und für andere, auf dem Betriebssystem iOS basierende, Apple-Produkte. Die Verwendung der Software ist gratis, vorausgesetzt wird jedoch die kostenlose Registrierung eines Apple Developer Benutzerkontos und ein Intel x86 fähiger MAC mit dem Betriebssystem Apple Mac OS X.

Das SDK besteht aus mehreren Komponenten, die zusammen die Basis der IDE formen (vgl. [Sadun 2008], S.1f.):

- Xcode ist eine umfangreiche Projektentwicklungs- und Managementumgebung und das wichtigste Werkzeug des SDK.
- Instruments wird benutzt um den Speicherverbrauch und die Leistung eines iPhone Apps bei Ausführung zu überwachen. Dieser Profiler ist ein wichtiges Werkzeug zur Verbesserung der Effizienz und Beseitigung von Speicherlöchern.
- Dashcode ist ein alternativer Ansatz zur Entwicklung von Apps. Es erlaubt Web-Apps zu erstellen die sowohl nativ als auch innerhalb eines Browsers lauffähig sind.
- Simulator erlaubt die direkte Ausführung von, mit dem Apple iOS SDK entwickelten, Apps und simuliert die Benutzeroberfläche und Funktionen des Apple iPhone.
- Interface Builder ist ein User Interface Designer zur Erstellung und Gestaltung der graphischen Elemente einer App.

Tabelle 5 bildet die Basisinformationen des iOS SDK ab.

Hersteller	Apple
------------	-------

Version	iOS SDK 4.1 mit Xcode 3.2.4
Mindestvoraussetzungen	Intel x86 fähiger MAC mit Apple Mac OS X 10.6 oder höher
Zielplattformen	Apple iOS kompatible Produkte, Apple Mac OS X kompatible Produkte (Cocoa)
Entwicklungssprache	Objective-C und Cocoa Framework
Homepage	http://developer.apple.com/

Tabelle 5: iOS SDK Basisinformationen

Aufgrund des modularen Aufbaus von Xcode werden auch andere Programmiersprachen wie z. B. C, C++ und Java unterstützt. Diese sind für die Entwicklung für das Apple iPhone, innerhalb dieser IDE, jedoch nicht relevant. Abbildung 1 zeigt die Komponenten Xcode, Interface Builder und Simulator.

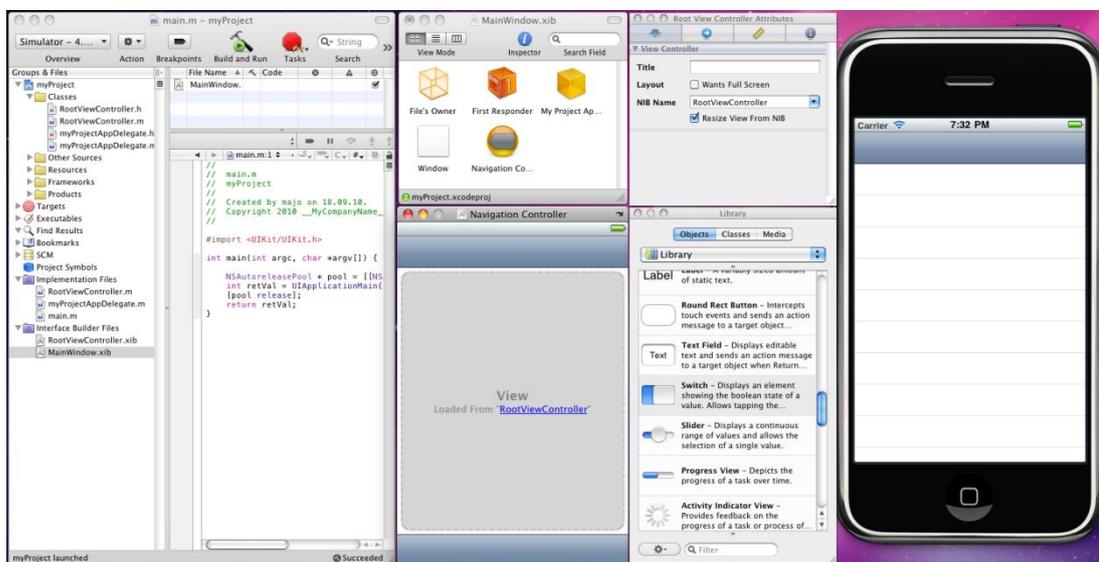


Abbildung 1: iOS SDK Komponenten

3.1.1. Beurteilung nach Kriterienkatalog

Nachfolgend werden die Ergebnisse der Evaluierung dieser IDE textuell und in +/-o-Notation dargestellt. Sofern nicht anders angegeben, wurden die Kriterien erträge direkt (durch Beobachten) festgestellt.

3.1.1.1. Allgemein

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
App Store Publishing	Apple iOS SDK und Xcode ist die offizielle IDE zur Entwicklung von Apple iOS kompatiblen Produkten. Infolgedessen wird die Veröffentlichung von Projekten, nach Beachtung der Lizenzkosten, unterstützt.	+
Community / Kundensupport	Apple stellt nach der kostenlosen Registrierung eines Apple Developer Benutzerkontos ein umfangreiches Forum und andere Ressourcen zur Verfügung. Mitglieder des kostenpflichtigen iOS Developer Programm haben die Möglichkeit zwei technische Supportanfragen per Jahr zu stellen (siehe Licensing / Kosten) bzw. weitere käuflich zu erwerben (vgl. [http://developer.apple.com] [1]).	+
Dokumentation	Apple stellt eine umfangreiche Reference Library mit Klassenbeschreibungen und (Video-)Tutorials zu Verfügung (vgl. [http://developer.apple.com] [2]).	+
Licensing / Kosten	Die Verwendung der IDE ist nach Registrierung eines Apple Developer Benutzerkontos kostenlos. Um Projekte auf einem Apple iOS kompatiblen Gerät zu testen oder im	99\$ bzw. 299\$ pro Jahr

	<p>Apple App Store zu veröffentlichen, müssen Lizenzkosten für die Teilnahme an einem Apple Developer Programm entrichtet werden. Folgende Mitgliedschaften werden angeboten (vgl. [http://developer.apple.com][3]):</p> <ul style="list-style-type: none"> • iOS Developer Programm für 99\$ pro Jahr (zur Distribution der Apps für Apple App Store) • iOS Enterprise Programm für 299\$ pro Jahr (zur Distribution der Apps innerhalb eines Unternehmens) • iOS University Programm ist kostenlos und nur für Universitäten zugänglich.
--	--

Tabelle 6: iOS SDK Allgemein

3.1.1.2. Code-Editor

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Quellcodeformatierung	Xcode formatiert Codeblöcke konsistent nach bestimmten Regeln.	+
Code Hilfe	Es werden keine zusätzlichen Informationen, abhängig von der aktuellen Position des Cursors auf dem Code, angezeigt.	-
Code Completion	Completion Lists stellen eine Liste von möglichen Ausdrücken zur Verfügung.	+
Regex-Suche	Die Suche und Änderung von	+

	Codepassagen mittels regulären Ausdrücken wird projektübergreifend unterstützt.	
Code Folding	Xcode markiert zusammenhängende Codeblöcke und ist fähig diese einzuklappen.	+
ToDo-List & Bookmarks	Bookmarks werden unterstützt und können verwaltet werden. Es ist nicht möglich ToDo-Lists zu erstellen oder diese mit Bookmarks zu verbinden.	o

Tabelle 7: iOS SDK Code-Editor

3.1.1.3. User Interface Designer

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Vordefinierte Elemente	Interface Builder bietet zahlreiche vordefinierte Elemente zur grafischen Gestaltung und Erweiterung der Funktionalität an.	+
Design erstellen / gestalten	Die Anordnung der Elemente erfolgt durch Drag & Drop. Es ist möglich die Elemente zu gruppieren.	+
Automatische Codeerzeugung	Interface Builder erzeugt .xib (XML Interface Builder) Dateien, die das User Interface eines iPhone Apps beschreiben.	+
Attribute via GUI änderbar	Erweiterte Attribute eines Elements können in einer GUI angepasst werden.	+
Attribute via Code änderbar	Die Elemente sollten nur innerhalb des Interface Builder manipuliert werden,	o

	Änderungen am Code einer .xib Datei sind nicht vorgesehen, aufgrund der auf XML basierenden Struktur jedoch möglich.
--	--

Tabelle 8: iOS SDK User Interface Designer

3.1.1.4. IDE Funktionalität

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Refactoring	Es ist möglich Refactoring einzusetzen.	+
UML-Diagramm	Es können automatisch Diagramme erstellt werden.	+
Emulation	Simulator erlaubt die Ausführung eines Projekts in einer Apple iOS ähnlichen Umgebung.	+
Debugging	Debugging wird unterstützt und es ist möglich Programmabläufe zu manipulieren.	+
Profiling	Mit Instruments steht ein mächtiges Profiling-Werkzeug zur Verfügung.	+
Unterstützt alle iPhone Features	Das Apple iOS SDK ist das offizielle Entwicklungskit für Apple iOS kompatible Produkte und unterstützt alle Gerätefunktionen (vgl. http://developer.apple.com)[4]).	+
Rapid Prototyping Eignung	Interface Builder kann zur schnellen Erstellung von Prototypen verwendet werden. Vordefinierte Elemente verfügen bereits über Funktionalität.	+

Tabelle 9: iOS SDK IDE Funktionalität

3.1.2. Fazit

Die offizielle IDE von Apple unterstützt nahezu alle evaluierten Kriterien. Allerdings muss zur Entwicklung eine Apple Mac OS X fähige Plattform vorhanden sein und die, bis auf Apple Produkte, wenig verbreitete Programmiersprache Objective-C eingesetzt werden.

3.2. Adobe Flash CS5 mit Adobe Packager for iPhone

Adobe Flash wird zur Erstellung von Anwendungen mit multimedialen und interaktiven Inhalten für das Web, Desktop-Betriebssysteme und mobile Plattformen verwendet. In der neuesten Version der Entwicklungsumgebung Adobe Flash Professional CS5 steht außerdem die Möglichkeit zur Erstellung von Apps, für Apple iOS kompatible Produkte, zur Verfügung.

Abbildung 2 zeigt die Entwicklungsumgebung Adobe Flash CS5 mit der Szenenansicht und dem integrierten Code-Editor.

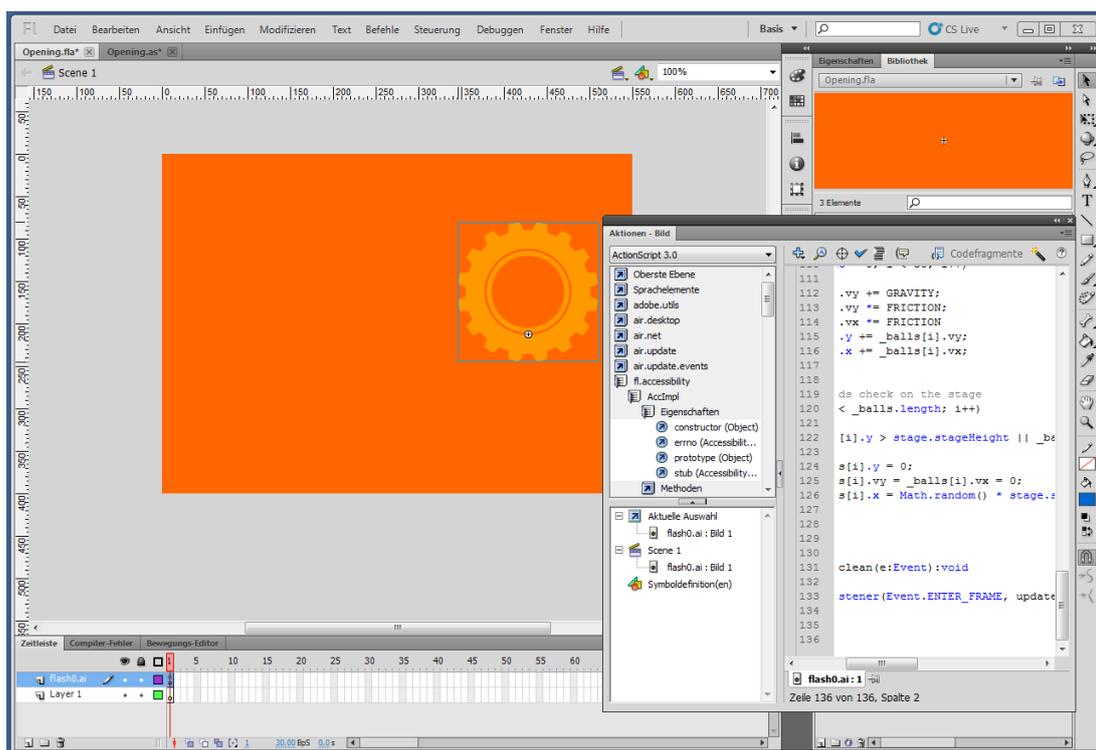


Abbildung 2: Flash CS5 Entwicklungsumgebung

In der Regel werden mit Adobe Flash entwickelte Applikationen zur Laufzeit interpretiert und sind plattformunabhängig. Die Ausführung erfolgt mittels

einer Laufzeitumgebung bzw. eines Browser-Plug-Ins. Da Apple die Interpretation von Programmcode zur Laufzeit auf Apple iOS Produkten untersagt und den offenen Standard HTML5 für multimediale Webinhalte präferiert, können diese Inhalte jedoch nicht ausgeführt werden (vgl. [\[http://www.apple.com\]](http://www.apple.com)[2]).

Adobe Packager for iPhone ist ein Werkzeug zur Kompilierung von Adobe Flash ActionScript 3.0 Projekten in Apple iOS kompatible Apps die nativ ausgeführt werden können und keine zusätzliche Laufzeitumgebung benötigen. Zur Entwicklung wird die von Adobe entwickelte Programmiersprache ActionScript 3.0 verwendet. Neben Adobe Flash CS5 unterstützt Adobe Packager for iPhone auch die Kompilierung von Projekten, die mit dem Adobe AIR 2.0.1 SDK entwickelt wurden (vgl. [\[http://labs.adobe.com\]](http://labs.adobe.com)[1]).

Tabelle 10 bildet die Basisinformationen ab.

Hersteller	Adobe
Version	Adobe Flash CS5 11.0.0.485 mit Adobe Packager for iPhone v1
Mindestvoraussetzungen	Intel x86 fähiger MAC mit Apple Mac OS X 10.5.7 oder höher bzw. PC mit Microsoft Windows XP SP2, Windows Vista oder Windows 7
Zielplattformen	Plattformunabhängig (via Plug-In oder RTE auf Microsoft Windows, Apple Mac OS X, Linux), Apple iOS kompatible Produkte
Entwicklungssprache	ActionScript 3.0
Homepage	http://www.adobe.com/products/flash/

Tabelle 10: Flash CS5 Basisinformationen

3.2.1. Beurteilung nach Kriterienkatalog

Nachfolgend werden die Ergebnisse der Evaluierung dieser IDE textuell und in +/-/o-Notation dargestellt. Sofern nicht anders angegeben, wurden die Kriterienerträge direkt (durch Beobachten) festgestellt.

3.2.1.1. Allgemein

Tabelle 11 bildet die gemessenen Kriterienerträge des Hauptkriterium Allgemein ab.

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
App Store Publishing	Nach geänderten Lizenzbedingungen seitens Apple ist die Veröffentlichung von, mit Adobe Packager for iPhone entwickelten Apps, im Apple App Store wieder erlaubt. Adobe nimmt die Weiterentwicklung der Software wieder auf (vgl. http://labs.adobe.com)[2]).	+
Community / Kundensupport	Adobe stellt ein Forum für Fragen die sich auf den Adobe Packager for iPhone beziehen zur Verfügung. Aufgrund der Verbreitung von Adobe Flash ist die Community sehr groß. Es ist möglich kostenpflichtige technische Supportanfragen zu stellen (vgl. http://labs.adobe.com)[3]).	+
Dokumentation	Adobe stellt ein FAQ für häufig gestellte Fragen zum Adobe Packager for iPhone zur Verfügung. Daneben existiert ein Developers Guide. Insgesamt sind beide Dokumente jedoch nicht sehr umfangreich (vgl.	o

Licensing / Kosten	<p>[http://labs.adobe.com][1]).</p> <p>Adobe Flash CS5 ist kostenpflichtig, eine voll funktionsfähige Testversion für 30 Tage ist verfügbar. Die Software kann auch als Bestandteil von verschiedenen Paketen erworben werden, die andere Adobe Produkte enthält. Außerdem besteht die Möglichkeit des Upgrades von der früheren Version Adobe Flash CS4 (vgl. [http://www.adobe.com][1]).</p> <p>Für die Veröffentlichung von Apps im Apple App Store wird zusätzlich eine kostenpflichtige Mitgliedschaft im Apple Developer Programm vorausgesetzt.</p>	<p>179\$ für Student and Teacher Edition</p> <p>199\$ für Upgrade</p> <p>699\$ für normale Version</p>
--------------------	---	--

Tabelle 11: Flash CS5 Allgemein

3.2.1.2. Code-Editor

Nachfolgend wird der integrierte Code-Editor in Adobe Flash Professional CS5 evaluiert. Die Ergebnisse sind in Tabelle 12 abgebildet.

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Quellcodeformatierung	Der integrierte Code-Editor formatiert Codeblöcke automatisch bzw. nach Anweisung des Benutzers.	+
Code Hilfe	Es werden keine zusätzlichen Informationen angezeigt.	-
Code Completion	Code Completion wird unterstützt, zeigt jedoch nur Ausdrücke der Adobe	o

	Action Script 3.0 API.	
RegEx-Suche	Die Suche mit regulären Ausdrücken wird nicht unterstützt.	-
Code Folding	Es ist möglich Codeblöcke durch Markieren der Zeilen auszublenden. Die Funktionalität, Codeblöcke zwischen Klammern automatisch auszublenden ist zwar vorhanden, funktioniert jedoch nur bei kurzen Blockpassagen.	o
ToDo-List & Bookmarks	Diese Funktionalität wird nicht unterstützt.	-

Tabelle 12: Flash CS5 Code-Editor

3.2.1.3. User Interface Designer

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Vordefinierte Elemente	Es sind keine vordefinierten Elemente zur Verwendung für Apple iOS kompatible Produkte vorhanden.	-
Design erstellen / gestalten	Da Adobe Flash Professional CS5 auf die Gestaltung von multimedialen Inhalten ausgelegt ist, wird dieses Kriterium komplett erfüllt.	+
Automatische Codeerzeugung	Elemente erzeugen keinen Programmcode der verändert werden könnte.	-
Attribute via GUI änderbar	Attribute die sich auf die Darstellung eines Elements beziehen, können mittels GUI verändert werden.	+

Attribute via Code änderbar	Änderungen von Attributen eines Elements im Code wirken sich nicht auf die Darstellung im User Interface Designer aus.	-
--------------------------------	--	---

Tabelle 13: Flash CS5 User Interface Designer

3.2.1.4. IDE Funktionalität

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Refactoring	Refactoring wird nicht unterstützt.	-
UML-Diagramm	Dieses Kriterium wird nicht unterstützt.	-
Emulation	Adobe AIR Debug Launcher kann zum Testen eines Projekts verwendet werden, gibt jedoch keinen Aufschluss auf die Leistungsfähigkeit auf einem nativen Gerät. Die Komponente Simulator die in der IDE Apple Xcode mit iOS SDK enthalten ist, kann nicht für Apps die mit Flash erstellt wurden, verwendet werden (vgl. http://labs.adobe.com)[1]).	o
Debugging	Adobe Flash CS5 unterstützt Debugging. Es ist möglich Programmabläufe zur Laufzeit zu manipulieren.	+
Profiling	Dieses Kriterium wird nicht unterstützt.	-
Unterstützt alle iPhone Features	Es werden folgende Features unterstützt (vgl. http://labs.adobe.com)[1]): Multitouch Eingabe, Neigungssensor,	o

Rapid Prototyping Eignung	Beschleunigungssensor, GPS Lokalisierung, programmübergreifendes Ausschneiden/Kopieren/Löschen. Apple Push Notifications ist nicht möglich. Aufgrund der Abwesenheit von - vordefinierten Elementen, eignet sich Adobe Flash CS5 nicht zur Erstellung von Prototypen für Apple iOS kompatible Produkte.
------------------------------	--

Tabelle 14: Flash CS5 IDE Funktionalität

3.2.2. Fazit

Adobe Packager for iPhone bietet die Möglichkeit bestehende Flash-Projekte auf Apple iOS kompatible Produkte anzupassen. Allerdings ist der Aufwand, vor allem durch das Fehlen von vordefinierten Elementen, nicht zu unterschätzen. Wegen der fehlenden Konvertierbarkeit zu einem Xcode Projekt und nicht vorhandener Profiling-Funktionen, gestaltet sich die Optimierung und das Testen der Flash-Apps auf einem nativen Gerät schwierig.

3.3. Unity iPhone

Unity ist ein Spiele-Entwicklungstool, das neben einer 3D-Engine eine integrierte Entwicklungsumgebung besitzt. Das Werkzeug unterstützt Cross-Plattform-Entwicklung für mobile Systeme, Desktop-PCs, Konsolen und für das Web. Das heißt, ein fertiggestelltes Projekt kann mit wenigen Anpassungen auf unterschiedlichsten Plattformen ausführbar gemacht werden, wie z.B. auf der Nintendo Wii, dem Betriebssystem Apple Mac OS X und dem Apple iPhone (vgl. [<http://unity3d.com>][1]).

Um Funktionalität herzustellen, können drei verschiedene Programmiersprachen verwendet werden: JavaScript, C# und der an Python angelehnte Dialekt Boo (vgl. [<http://unity3d.com>][2]).

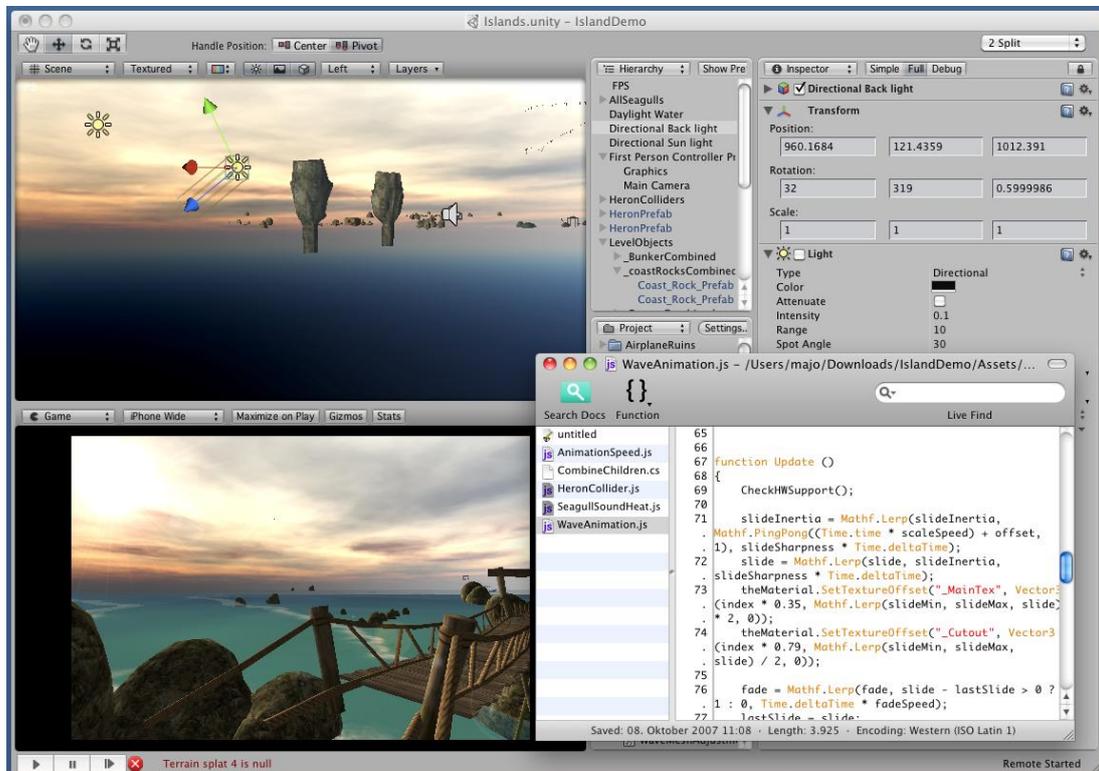


Abbildung 3: Unity iPhone Entwicklungsumgebung

Abbildung 3 zeigt die Entwicklungsumgebung mit eingblendeten Editor und den zwei verschiedenen Raumansichten. Die obere Ansicht des Raumes dient zur Manipulation aller grafischen Elemente und wird Szenenansicht genannt. Die untere Ansicht zeigt die Vorschau des Raumes auf der Zielplattform und wird Spielansicht genannt. Das Projekt kann direkt in der Spielansicht getestet werden, während Attribute geändert werden oder Funktionalität durch Codezeilen hinzugefügt wird. Dies wird WYSIWYG-Prototyping bezeichnet (vgl. [<http://unity3d.com>][3]).

Beim Veröffentlichen von Projekten wird das komplette Unity Projekt automatisch in ein Apple Xcode Projekt konvertiert. Es können sämtliche Features der Apple Xcode IDE und dadurch auch alle Features eines Apple iOS kompatiblen Produkts benutzt werden. Außerdem kann jederzeit

während des Entwicklungsprozesses ein natives Gerät für eine Live-Vorschau zugeschaltet werden, während Objekte und Programmabläufe in der IDE modifiziert werden (vgl. [<http://unity3d.com>][3]).

Hersteller	Unity Technologies
Version	Unity iPhone 1.7
Mindestvoraussetzungen	Intel x86 fähiger MAC mit Apple Mac OS X 10.5.7 oder höher
Zielplattformen	Microsoft Windows XP, Windows Vista, Windows 7, Apple iOS kompatible Produkte, Apple Mac OS X, Nintendo Wii, Webapplikationen
Entwicklungssprache	JavaScript, C#. Boo wird auf Unity iPhone nicht unterstützt (vgl. [http://unity3d.com][4]).
Homepage	http://unity3d.com

Tabelle 15: Unity iPhone Basisinformationen

3.3.1. Beurteilung nach Kriterienkatalog

Nachfolgend werden die Ergebnisse der Evaluierung dieser IDE textuell und in +/-o-Notation dargestellt. Sofern nicht anders angegeben, wurden die Kriterienerträge direkt (durch Beobachten) festgestellt.

3.3.1.1. Allgemein

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
App Store Publishing	Die Veröffentlichung von Projekten auf Apple App Store wird unterstützt, Unity iPhone Projekte werden in Xcode Projekte umgewandelt.	+
Community / Kundensupport	Unity stellt neben einem Forum, einer Chatfunktion und einer Script Wiki	+

	<p>außerdem noch eine benutzergesteuerte Wunschliste und FAQ mit Voting-System zur Verfügung (vgl. http://unity3d.com)[5]).</p>	
Dokumentation	<p>Neben (Video-) Tutorials stehen Beispielprojekte, ein Benutzer- und Referenzhandbuch und eine umfangreiche Reference Library zur Verfügung (vgl. http://unity3d.com)[6]).</p>	+
Licensing / Kosten	<p>Es stehen verschiedene Produktpakete von Unity zur Verfügung. Die Benutzung von Unity iPhone ist kostenlos. Um Projekte für das Apple iPhone zu veröffentlichen, stehen zwei Lizenzen zur Auswahl (vgl. http://unity3d.com)[4]):</p> <ul style="list-style-type: none"> • Unity iPhone Basic ist für Privatanwender oder kleinen Unternehmen mit einem Jahresumsatz von maximal 100.000\$ verfügbar. • Unity iPhone Pro ist bei Unternehmen mit einem größeren Jahresumsatz als 100.000\$ verpflichtend. Zusätzlich optimiert diese Version Speicherverbrauch und Leistung des Projekts für das Endgerät. 	<p>300\$ für iPhone Basic 1500\$ bis 2400\$ für Unity iPhone Pro</p>

Tabelle 16: Unity iPhone Allgemein

3.3.1.2. Code-Editor

Neben dem, in Unity iPhone integrierten, Editor Unitron können auch andere externe Editoren in der IDE definiert werden. Tabelle 17 bildet die Kriterien erträge des internen Editors ab.

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Quellcodeformatierung	Unitron unterstützt die konsistente Formatierung von Codepassagen nach bestimmten Regeln.	+
Code Hilfe	Es werden keine zusätzlichen Informationen, abhängig von der Position des Cursors, angezeigt.	-
Code Completion	Completion-Lists zeigen mögliche passende Ausdrücke an der aktuellen Cursorposition an.	+
RegEx-Suche	Reguläre Ausdrücke werden unterstützt.	+
Code Folding	Das Einklappen von zusammenhängenden Codepassagen wird nicht unterstützt.	-
ToDo-List & Bookmarks	Bookmarks und ToDo-Lists werden nicht unterstützt.	-

Tabelle 17: Unity iPhone Code-Editor

3.3.1.3. User Interface Designer

Das Unity Entwicklungstool integriert eine eigene 3D-Engine. Das Hauptkriterium User Interface Designer beschäftigt sich bei Unity mit der Erstellung von zwei- und dreidimensionalen Räumen und ist in Tabelle 18 abgebildet.

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Vordefinierte Elemente	Es sind zahlreiche vordefinierte grafische Körper, Partikel-, Physik- und Audioeffekte vorhanden.	+
Design erstellen / gestalten	Alle Elemente können durch Maus- oder Tastatureingaben angeordnet, gruppiert und ausgerichtet werden.	+
Automatische Codeerzeugung	Unity erzeugt keinen Programmcode bei der Erstellung von Elementen der durch den User manipuliert werden könnte. Aufgrund der Ausrichtung im Raum und der komplexen Körper, wäre die Definition und Änderung von Elementen im Programmcode sehr komplex.	-
Attribute via GUI änderbar	Alle Attribute eines Objekts können via GUI verändert werden, bzw. weitere Funktionalitäten, Animationen und Effekte können intuitiv hinzugefügt werden.	+
Attribute via Code änderbar	Änderungen am Programmcode eines grafischen Objekts durch den User sind nicht vorgesehen, da diese durch eine GUI erfolgen. Änderungen von Funktionalitäten innerhalb des Programmcodes werden jedoch sofort innerhalb der Spielvorschau angezeigt.	o

Tabelle 18: Unity iPhone User Interface Designer

3.3.1.4. IDE Funktionalität

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Refactoring	Refactoring wird nicht unterstützt.	-
UML-Diagramm	Unity Projekte können nicht als UML-Diagramm abgebildet werden.	-
Emulation	Es kann sowohl die Hardware des Apple iPhone als auch die Netzwerkverbindung mit anderen iPhones (z.B. bei Mehrspielerfunktionalität) emuliert werden.	+
Debugging	Unity teilt sein Programmfenster in zwei Ansichten. Manipulationen innerhalb der Szenenansicht beeinflussen die Spielansicht.	+
Profiling	Durch iPhone Remote können Unity-Projekte in Echtzeit bearbeitet und gleichzeitig auf einem nativen Gerät ausgeführt werden (vgl. http://unity3d.com [3]).	+
Unterstützt alle iPhone Features	Unity iPhone unterstützt durch die Konvertierbarkeit in ein Xcode Projekt alle Features des Apple iPhone.	+
Rapid Prototyping Eignung	Durch die zwei Ansichten der IDE werden Änderungen jeglicher Art sofort in der Spielansicht sichtbar. Zusätzlich beschleunigen zahlreiche vordefinierte Elemente die Entwicklung eines Prototypen.	+

Tabelle 19: Unity iPhone IDE Funktionalität

3.3.2. Fazit

Unity iPhone besitzt alle nötigen Werkzeuge zur Entwicklung von dreidimensionalen Spielen für Apple iOS kompatible Produkte. Durch die Profiling- und Prototyping-Fähigkeiten kann die Testphase eines Projekts neben der Entwicklung erfolgen. Die verschiedenen einsetzbaren Programmiersprachen und die Möglichkeit der Cross-Plattform-Entwicklung erhöhen die Flexibilität.

3.4. GameSalad

GameSalad ist ein alternativer Ansatz zur Erstellung von Spielen für Apple iOS kompatible Produkte. Die Entwicklungsumgebung verzichtet auf eine textuelle Programmiersprache, Funktionalität wird durch Definition von Verhaltensregeln der grafischen Objekte im GUI der IDE geschaffen (vgl. [\[http://gamesalad.com\]](http://gamesalad.com)[1]). GameSalad ist eine VPL. Verhaltensregeln können durch spezifizierte Ereignisse ausgelöst und mit Abfragen verzweigt werden.

Neben Apple iOS kompatiblen Produkten unterstützt GameSalad auch die Veröffentlichung von Projekten im Web. Durch die Komponente Preview Player können Projekte außerdem jederzeit getestet werden. (vgl. [\[http://gamesalad.com\]](http://gamesalad.com)[2]).

Abbildung 4 zeigt die Entwicklungsoberfläche von GameSalad. Verhaltensweisen können aus der Liste auf der linken Seite ausgewählt werden und per Drag & Drop auf der rechten Seite zu Verhaltensregeln verzweigt werden.

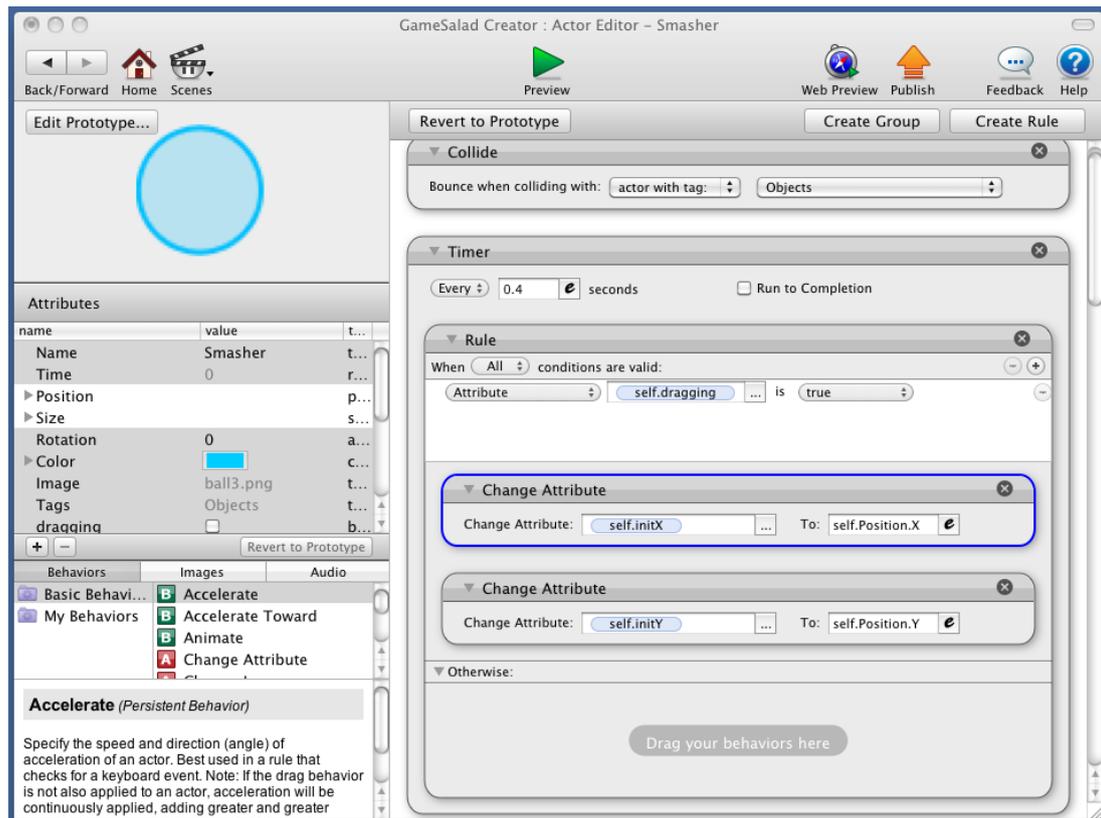


Abbildung 4: GameSalad Entwicklungsumgebung

Tabelle 20 bildet die Basisinformationen von GameSalad ab.

Hersteller	Gendai Games
Version	GameSalad 0.8.9
Mindestvoraussetzungen	Intel x86 fähiger MAC mit Apple Mac OS X 10.5.7 oder höher
Zielplattformen	Apple iOS kompatible Produkte, Webapplikationen
Entwicklungssprache	VPL
Homepage	http://gamesalad.com/

Tabelle 20: GameSalad Basisinformationen

3.4.1. Beurteilung nach Kriterienkatalog

Nachfolgend werden die Ergebnisse der Evaluierung dieser IDE textuell und in +/-o-Notation dargestellt. Kriterien, die aufgrund der Beschaffenheit dieser IDE nicht gemessen werden können, sind mit n.v. (nicht vorhanden) vermerkt. Sofern nicht anders angegeben, wurden die Kriterienerträge direkt (durch Beobachten) festgestellt.

3.4.1.1. Allgemein

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
App Store Publishing	Die Veröffentlichung von GameSalad-Projekten im Apple App Store wird unterstützt (vgl. http://gamesalad.com)[1]). Die Veröffentlichungsfunktion konvertiert ein GameSalad-Projekt in ein Xcode Projekt.	+
Community / Kundensupport	Gendai Games stellt ein Forum für registrierte Mitglieder zur Verfügung. Die Registrierung ist kostenlos. Herstellersupport steht nur Mitgliedern des GameSalad Pro Programmes zur Verfügung (siehe Licensing / Kosten) (vgl. http://gamesalad.com)[3]).	o
Dokumentation	Es stehen (Video-) Tutorials und eine Reference Library zur Verfügung. Zahlreiche abgeschlossene Projekte können frei innerhalb der Anwendung heruntergeladen und verändert werden.	+
Licensing / Kosten	Die Benutzung der Software ist kostenlos. Um Projekte im Apple App	99\$ pro Jahr für

	<p>Store zu veröffentlichen und die Profiling-App GameSalad Viewer nutzen zu können, muss eine kostenpflichtige Mitgliedschaft eingerichtet werden (vgl. [http://gamesalad.com][3]):</p> <ul style="list-style-type: none"> • GameSalad Express ist die Standard-Mitgliedschaft und ermöglicht das Testen und Veröffentlichen von GameSalad-Projekten für Apple iOS kompatible Produkte. • GameSalad Pro bietet zusätzlich zu den Vorzügen der Express Mitgliedschaft, die Nutzung des Herstellersupports und weitere Möglichkeiten zur Individualisierung eines Projekts. 	Express, 1999\$ pro Jahr für Pro
--	---	----------------------------------

Tabelle 21: GameSalad Allgemein

3.4.1.2. Code-Editor

GameSalad ist eine VPL und besitzt daher keinen Code-Editor. Infolgedessen wird auf die Evaluierung des Hauptkriteriums Code-Editor verzichtet.

3.4.1.3. User Interface Designer

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Vordefinierte Elemente	Projektvorlagen und wichtige Verhaltensweisen eines Elements sind vordefiniert.	+

Design erstellen / gestalten	Elemente können durch Drag & Drop auf der Szene platziert und verändert werden.	+
Automatische Codeerzeugung	Dieses Kriterium kann nicht evaluiert werden. GameSalad ist eine VPL und erzeugt keinen manipulierbaren Code.	n.v.
Attribute via GUI änderbar	Alle Attribute eines Elements können durch die GUI verändert werden.	+
Attribute via Code änderbar	Dieses Kriterium kann nicht evaluiert werden. GameSalad ist eine VPL und erzeugt keinen manipulierbaren Code.	n.v.

Tabelle 22: GameSalad User Interface Designer

3.4.1.4. IDE Funktionalität

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Refactoring	Dieses Kriterium kann nicht evaluiert werden. GameSalad ist eine VPL und erzeugt keinen manipulierbaren Code.	n.v.
UML-Diagramm	Dieses Kriterium kann nicht evaluiert werden. GameSalad ist eine VPL und erzeugt keine Klassen.	n.v.
Emulation	Die Komponente GameSalad Preview Player ermöglicht die Ausführung des Projekts in einer iPhone ähnlichen Umgebung.	+
Debugging	Die IDE beherrscht keine Debugging-Fähigkeiten.	-
Profiling	GameSalad Viewer ist eine Profiling-App für Apple iOS kompatible Produkte	o

	und überwacht die Leistung und den Speicherverbrauch eines auf einem nativen Gerät ausgeführten GameSalad-Projekts. Es sind nur einfache Manipulationen wie z.B. Pausieren oder Neustarten möglich (vgl. [http://gamesalad.com][4]).	
Unterstützt alle iPhone Features	GameSalad unterstützt in der aktuellen Version keine GPS-Lokalisierung oder den digitalen Kompass.	o
Rapid Prototyping Eignung	GameSalad ist eine VPL und besitzt vordefinierte Verhaltensweisen die leicht in komplexe Regeln verschachtelt werden können. Daher eignet sich diese IDE besonders für Rapid Prototyping.	+

Tabelle 23: GameSalad IDE Funktionalität

3.4.2. Fazit

Das Ziel von GameSalad ist die Schaffung einer IDE, die auch ohne Programmierkenntnisse produktiv eingesetzt werden kann. Die grafische Formulierung der Verhaltensregeln eines Elements und die intuitive Bedienung, ermöglichen es in kurzer Zeit Prototypen zu Spiele-Ideen zu erstellen.

3.5. PhoneGap

PhoneGap ist ein offenes Cross-Plattform-Entwicklungsframework und unterstützt die Entwicklung für verschiedene Smartphone-Betriebssysteme und Apple iOS kompatiblen Produkte. Es kombiniert die Zugänglichkeit von HTML und JavaScript in der Programmierung mit der Möglichkeit

gerätespezifische Features des Apple iPhones einzusetzen (vgl. <http://www.phonegap.com>[1]).

Die Entwicklung für das Apple iPhone mit PhoneGap erfolgt direkt in Xcode (vgl. <http://phonegap.pbworks.com>[1]). Daher können alle Features dieser IDE genutzt werden.

Tabelle 24 bildet die Basisinformationen von PhoneGap ab.

Hersteller	Nitobi
Version	PhoneGap 0.9.1
Mindestvoraussetzungen	Intel x86 fähiger MAC mit Apple Mac OS X 10.5 oder höher
Zielplattformen	Apple iOS kompatible Produkte, Android, Blackberry, Palm, Microsoft Mobile, Nokia Symbian
Entwicklungssprache	JavaScript, HTML
Homepage	http://www.phonegap.com/

Tabelle 24: PhoneGap Basisinformationen

3.5.1. Beurteilung nach Kriterienkatalog

Nachfolgend werden die Ergebnisse der Evaluierung dieser IDE textuell und in +/-o-Notation dargestellt. Sofern nicht anders angegeben, wurden die Kriterienerträge direkt (durch Beobachten) festgestellt.

3.5.1.1. Allgemein

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
App Store Publishing	Die Veröffentlichung von PhoneGap Projekten wird unterstützt (vgl. http://phonegap.pbworks.com [2]).	+
Community /	Nitobi stellt kein eigenes Forum für PhoneGap zur Verfügung. Auf der	o

Kundensupport	Google Gruppe PhoneGap können Fragen gestellt und diskutiert werden (vgl. [http://www.phonegap.com][2]).	
Dokumentation	Es ist eine umfangreiche Wiki mit verschiedenen Tutorials vorhanden. (vgl. [http://www.phonegap.com][2]).	o
Licensing / Kosten	PhoneGap steht unter der MIT Lizenz und darf dadurch frei benutzt und verändert werden. Die kommerzielle Nutzung ist erlaubt (vgl. [http://www.phonegap.com][3]).	0\$

Tabelle 25: PhoneGap Allgemein

3.5.1.2. Code-Editor

Die Xcode IDE wird zur Entwicklung von PhoneGap-Apps eingesetzt. Der Code-Editor der Xcode IDE wurde bereits in Kapitel 3.1.1.2 evaluiert.

3.5.1.3. User Interface Designer

Interface Builder, eine Komponente von Xcode, wird zum Gestalten der Benutzeroberfläche verwendet. Diese Komponente wurde bereits in Kapitel 3.1.1.3 evaluiert.

3.5.1.4. IDE Funktionalität

<i>Kategorie</i>	<i>Beschreibung</i>	<i>Wertung</i>
Refactoring	Refactoring wird durch Xcode unterstützt.	+
UML-Diagramm	Klassendiagramme werden durch Xcode unterstützt, finden bei den auf JavaScript basierenden PhoneGap-Projekten jedoch keine Anwendung.	n.v.
Emulation	Die Komponente Simulator kann bei	+

	PhoneGap-Projekten verwendet werden.	
Debugging	Xcode unterstützt Debugging.	+
Profiling	Mit Instruments steht ein mächtiges Profiling-Werkzeug zur Verfügung.	+
Unterstützt alle iPhone Features	Bis auf programmübergreifendes Kopieren und Einfügen werden alle Features unterstützt (vgl. http://phonegap.pbworks.com)[3]).	o
Rapid Prototyping Eignung	HTML und JavaScript erlauben die Anfertigung von funktionsfähigen Prototypen in kurzer Zeit. Interface Builder stellt zahlreiche vordefinierte Elemente zur Verfügung.	+

Tabelle 26: PhoneGap IDE Funktionalität

3.5.2. Fazit

PhoneGap eignet sich durch den Einsatz von HTML und JavaScript und der umfangreichen Funktionalität der Xcode IDE hervorragend für Rapid Prototyping. Durch die Cross-Plattform-Fähigkeit können Projekte auf zahlreichen mobilen Plattformen veröffentlicht werden. Hervorzuheben ist außerdem ist die kostenlose Nutzung unter der MIT Lizenz und die nahezu vollständige Unterstützung der Features des Apple iPhone.

4. Zusammenfassung

Dieses Kapitel beinhaltet die übersichtliche Darstellung der Ergebnisse mit abschließenden Schlussfolgerungen und einem Ausblick auf die Zukunft.

4.1. Übersichtstabelle

Nachfolgend werden die Ergebnisse der Evaluierung textuell und in +/-/o-Notation dargestellt. Kriterien, die aufgrund der Beschaffenheit einer IDE nicht gemessen werden können, sind mit n.v. (nicht vorhanden) vermerkt. Die Ergebnisse sind in Tabelle 27 dargestellt.

	<i>Apple iOS SDK & Xcode</i>	<i>Adobe Flash CS5 & Packager for iPhone</i>	<i>Unity for iPhone</i>	<i>GameSalad</i>	<i>PhoneGap & Xcode</i>
Entwicklungssprache	Objective-C und Cocoa Framework	Action Script 3.0	Java Script und C#	VPL	JavaScript, HTML
<i>Allgemein</i>					
App Store Publishing ²	+	+	+	+	+
Community / Kundensupport	+	+	+	o	o
Dokumentation	+	o	+	+	o
Licensing / Kosten	99\$ bzw. 299\$ pro Jahr	179\$ bis 699\$	300\$ bis 2400\$	99\$ bzw. 1999\$ pro Jahr	Frei nutzbare Lizenz

² Lizenzbestimmungen des Apple App Store zum Zeitpunkt der Evaluierung. Die Mitgliedschaft im kostenpflichtigen Apple Developer Programm ist voraussetzend.

	<i>Apple iOS SDK & Xcode</i>	<i>Adobe Flash CS5 & Packager for iPhone</i>	<i>Unity for iPhone</i>	<i>GameSalad</i>	<i>PhoneGap & Xcode</i>
<i>Code-Editor^{3 4}</i>					
Quellcode- formatierung	+	+	+	n.v.	+
Code Hilfe	-	-	-	n.v.	-
Code Completion	+	0	+	n.v.	+
RegEx-Suche	+	-	+	n.v.	+
Code Folding	+	0	-	n.v.	+
ToDo-List & Bookmarks	0	-	-	n.v.	0
<i>User Interface Designer⁵</i>					
Vordefinierte Elemente	+	-	+	+	+
Design gestalten / erstellen	+	+	+	+	+
Automatische Codeerzeugung	+	-	-	n.v.	+
Attribute via GUI änderbar	+	+	+	+	+
Attribute via Code änderbar	0	-	0	n.v.	0

3 PhoneGap nutzt den Code-Editor der Xcode IDE. Ergebnisse werden daher grau dargestellt.

4 GameSalad basiert auf VPL und benötigt keinen Code-Editor.

5 PhoneGap nutzt die Xcode Komponente Interface Builder. Ergebnisse werden daher grau dargestellt.

	<i>Apple iOS SDK & Xcode</i>	<i>Adobe Flash CS5 & Packager for iPhone</i>	<i>Unity for iPhone</i>	<i>GameSalad</i>	<i>PhoneGap & Xcode</i>
<i>IDE Funktionalität</i>					
Refactoring	+	-	-	n.v.	+
UML-Diagramm	+	-	-	n.v.	n.v.
Emulation	+	o	+	+	+
Debugging	+	+	+	-	+
Profiling	+	-	+	o	+
Unterstützt alle iPhone Features	+	o	+	o	o
Rapid Prototyping Eignung	+	-	+	+	+

Tabelle 27: Übersichtstabelle

4.2. Ergebnisse und Ausblick

Neben Apple iOS SDK und Apple Xcode entwickelten sich in den letzten Monaten ernstzunehmende Alternativen. Bei allen evaluierten Entwicklungsumgebungen ist es (derzeit) möglich, die Apps im Apple App Store zu veröffentlichen. Die Entscheidung welche IDE verwendet werden sollte, wird durch das Projekt getroffen.

PhoneGap eignet sich beispielsweise hervorragend für die Erstellung von interaktiven Unternehmensapplikationen. Die Erstellung der App wird durch HTML und Java-Script ungemein beschleunigt und nebenbei können alle mobilen Plattformen durch die Cross-Plattform-Entwicklung abgedeckt werden.

Unity geht sogar noch einen Schritt weiter: Die Veröffentlichung von demselben Spiel auf mobilen Plattformen, Konsolen und Desktop-PCs.

Testphase und Entwicklungsphase verschmelzen bei Unity, reduzieren den Arbeitsaufwand und erhöhen die Flexibilität.

GameSalad bietet die schnellste Möglichkeit einen Prototyp einer (2D-) Spielidee zu erstellen. Die auf VPL basierende IDE kann auch ohne Programmiererfahrung benutzt werden und vermittelt dennoch grundlegende Kenntnisse.

Adobe Flash CS5 wird in nächster Zeit durch die immense Anzahl von bestehenden Flash-Web-Spielen eine größere Bedeutung gewinnen. Derzeit ist die Anpassung an die Funktionen des Apple iPhone mit einem größeren Aufwand verbunden und die derzeitige Leistung, der mit dem Adobe Packager for iPhone kompilierten Apps, nicht mit den Desktop-Anwendungen vergleichbar. Diese Barriere kann aber mit weiteren Optimierungen sicherlich überwunden werden.

Apple hat seine Entwicklungstools nahezu perfektioniert, sollte sich aber vor anderen mobilen Plattformen in Acht nehmen. Die Konkurrenz schläft nicht und mit der zunehmenden Anzahl an Android-Geräten, sollte sich Apple um jede zusätzliche Entwicklungsumgebung bemühen. Die geänderten Lizenzbestimmungen des Apple App Store sind ein Schritt in die richtige Richtung.

Literaturverzeichnis

[EN 1325-1 1996]

Value Management, Wertanalyse, Funktionenanalyse Wörterbuch, Teil 1: Wertanalyse und Funktionenanalyse 1996

[Franke 2010]

Franke, Jens: Interaktiver Goldtausch, in: Weave, 4. Jg., 2/2010, S. 38-45. 2010

[Lutz 2009]

Lutz, Heinrich; Stelzer, Dirk: Informationsmanagement. Grundlagen, Aufgaben, Methoden, 9. Aufl., Oldenbourg Verlag München 2009

[Sadun 2008]

Sadun, Erica: The iPhone Developer's Cookbook. Building Applications with the iPhone SDK, 1. Aufl., Addison-Wesley Professional 2008

[Stamelos 2000]

Stamelos, Ioannis; Vlahavas, Ioannis; Refanidis, Ioannis; Tsoukiàs, Alexis: Knowledge Based Evaluation of Software Systems. A Case Study, Aristotle University of Thessaloniki 2000

[Wihsböck 2007]

Wihsböck, Michael: Moderne Softwareentwicklungsumgebungen. Evaluierung Java-basierter Ansätze. Diplomarbeit Technische Universität Wien 2007

Internetquellen:

[\[http://www.apple.com\]](http://www.apple.com)

[1] Statement by Apple on App Store Review Guidelines;
<http://www.apple.com/pr/library/2010/09/09statement.html>; 26.9.2010

[2] Thoughts on Flash; <http://www.apple.com/hotnews/thoughts-on-flash/>;
26.9.2010

[<http://developer.apple.com>]

[1] Apple Developer Technical Support;

<http://developer.apple.com/support/resources/technical-support.html>;

26.9.2010

[2] iOS Reference Library; <http://developer.apple.com/library/ios/navigation/>;

26.9.2010

[3] iOS Developer University Program;

<http://developer.apple.com/programs/ios/university/>; 26.9.2010

[4] iOS Features; <http://developer.apple.com/technologies/ios/features.html>;

26.9.2010

[<http://www.adobe.com>]

[1] Adobe Education pricing;

http://www.adobe.com/education/purchasing/education_pricing.html;

26.9.2010

[<http://labs.adobe.com>]

[1] Packager for iPhone Developer FAQ;

http://labs.adobe.com/wiki/index.php/Packager_for_iPhone:Developer_FAQ;

26.9.2010

[2] Packager for iPhone Special Update;

<http://labs.adobe.com/technologies/packagerforiphone/>; 26.9.2010

[3] Adobe-Support für Einzelanwender;

<http://www.adobe.com/de/support/programs/individuals/>; 26.9.2010

[<http://unity3d.com>]

[1] Unity One-Click Deployment;

<http://unity3d.com/unity/features/deployment>; 26.9.2010

[2] Unity Scripting; <http://unity3d.com/unity/features/scripting>; 26.9.2010

[3] Unity iPhone Publishing; <http://unity3d.com/unity/features/iphone-publishing>; 26.9.2010

[4] Unity License Comparision; <http://unity3d.com/unity/licenses>; 26.9.2010

[5] Unity Community; <http://unity3d.com/support/community>; 26.9.2010

[6] Unity Documentation; <http://unity3d.com/support/documentation/>;
26.9.2010

<http://gamesalad.com>

[1] GameSalad About; <http://gamesalad.com/products/creator>; 26.9.2010

[2] GameSalad Intro; http://gamesalad.com/wiki/gamesalad_intro; 26.9.2010

[3] GameSalad Membership Comparision;
<http://gamesalad.com/membership/pricing>; 26.9.2010

[4] GameSalad Viewer;
http://gamesalad.com/wiki/developing_for_iphone:building_gamesalad_viewer;
26.9.2010

<http://www.phonegap.com>

[1] What is PhoneGap; <http://www.phonegap.com>; 26.9.2010

[2] PhoneGap Community; <http://www.phonegap.com/community/>; 26.9.2010

[3] PhoneGap License; <http://www.phonegap.com/license/>; 26.9.2010

<http://phonegap.pbworks.com>

[1] Getting Started with PhoneGap; [http://phonegap.pbworks.com/Getting-Started-with-PhoneGap-\(iPhone\)](http://phonegap.pbworks.com/Getting-Started-with-PhoneGap-(iPhone)); 26.9.2010

[2] PhoneGap FAQ; <http://phonegap.pbworks.com/FAQ>; 26.9.2010

[3] PhoneGap Roadmap; <http://phonegap.pbworks.com/Roadmap>; 26.9.2010