

2 Von der Idee zur App

Bevor es mit der App-Entwicklung und dem Einsetzen von PhoneGap richtig losgeht, möchte ich an dieser Stelle einen kurzen Überblick zur Konzeption von Apps geben. Wenn Sie bereits Erfahrung mit der Erstellung und Durchführung von mobilen Anwendungen haben, können Sie dieses Kapitel gerne überspringen.

Allerdings werden Sie hier auch die Anwendungsbereiche von PhoneGap kennenlernen. Dabei sehen Sie die möglichen Unterstützungspunkte, die PhoneGap bietet, aber auch das, was PhoneGap nicht leisten kann bzw. nicht die Aufgabe von PhoneGap ist. Hierbei geht es um das Vorgehen, wie eine App entwickelt wird. Damit müssen Sie einige Aufgaben an andere Bibliotheken delegieren oder diese selbst implementieren. Damit Sie eine bessere Vorstellung haben, welche Schritte von der Idee bis zum App Store nötig sind, ist dieses Kapitel gedacht. Wir gehen bei der folgenden Betrachtung davon aus, dass Sie bereits festgelegt haben, dass es eine Cross-Plattform-App werden soll. Lassen Sie uns also bei der Idee zu einer neuen App anfangen und diese dann alle Entwicklungsschritte bis in den App Store durchlaufen.

2.1 Von der App-Idee zur Skizze

Natürlich beginnt alles mit einer Idee zu einer App. Damit diese auch umsetzbar ist, sollten Sie immer bedenken:

»Eine App sollte meistens nur eine Sache tun, diese aber richtig gut!«

Bevor Sie also eine App in der Art eines Schweizer Taschenmessers ersinnen, überlegen Sie sich lieber genau, wann und wie Ihre Anwendung dem Benutzer einen konkreten Nutzen bietet. Die Beschreibung solcher Szenarien und dessen, was unsere App dabei leistet, nennt man auch Use-Cases.

In unserem Beispiel wollen wir eine Einkaufsliste als App erstellen. Die folgenden zwei Use-Cases soll die geplante App erfüllen:

- Artikel für den nächsten Einkauf erfassen
- Gekaufte Artikel mit Preis im Archiv anzeigen

Natürlich könnte das jede Notizzettel-App auch leisten, aber unsere App unterstützt den Einkaufsprozess besonders gut. Es sind nicht viele Anwendungsfälle, aber wenn diese schnell und effizient ausgeführt werden, ist das ein Nutzen für den Anwender. Wie schon erwähnt, die Use-Cases könnten auch mit einer einfachen To-do-Liste erzielt werden, aber für den schnellen Umgang hat die App spezielle Features. Unsere App soll nämlich folgende Eigenschaften haben:

- verfügbar als Cross-Plattform-Anwendung
- auch ohne Netz nutzbar (offline)
- bietet eine Synchronisation der Daten mit der Cloud, damit auch auf anderen Plattformen nutzbar
- kann GPS-Daten der To-dos erfassen
- kann Artikel-Barcodes scannen

Nun kennen Sie unsere Idee, und wir haben das Featureset dazu abgesteckt.

Natürlich können Sie anstatt Use-Cases auch andere Möglichkeiten wie User Stories oder Ähnliches nutzen. Denken Sie aber immer daran, die Hauptaufgaben zu beschreiben.

Jetzt schauen Sie sich in den virtuellen Marktplätzen der verschiedenen Anbieter um, ob solch eine Anwendung schon existiert. Eine kurze Prüfung ergab, dass eine Einkaufsliste mit diesen Möglichkeiten noch eine gute Chance im Markt hat. Daher gehen wir nun über zur nächsten Phase, dem sogenannten Storyboard.

2.2 Ein Storyboard erstellen

Bevor es mit dem Programmieren losgeht, wird ein kleines Storyboard erstellt. Damit können Sie sehr schnell erste Oberflächen und Klickpfade festlegen. Sie finden solche Storyboards auch unter dem Begriff *Scribbles* oder *Mock-ups*. Dabei geht es aber immer um das Gleiche: das Skizzieren der App-Oberfläche und damit das Übertragen der Idee in eine reale Form.

Im Storyboard werden die durchzuführenden Tätigkeiten erstmals auf eine Benutzeroberfläche umgesetzt. Sie entscheiden dabei, wo Listen oder Knöpfe gesetzt werden, aber auch, welche Ansichten welche Elemente enthalten. Dabei legen Sie nicht unbedingt Wert darauf, dass der Knopf oder die Liste wie ein iOS- oder Android-Knopf aussieht. Es geht erst einmal nur darum, dass er überhaupt existiert und einen Platz hat. Für diesen Arbeitsschritt stehen Softwaretools wie Balsamiq Mockups¹ oder auch Photoshop zur Verfügung.

Selbst ein einfaches DIN-A5-Blatt und ein Bleistift können für diesen Arbeitsschritt ausreichen. Es geht im Wesentlichen darum, dass Sie damit visualisieren, wie viele Ansichten (Views) es gibt und wie diese genutzt werden können.

1. <http://www.balsamiq.com/products/mockups>

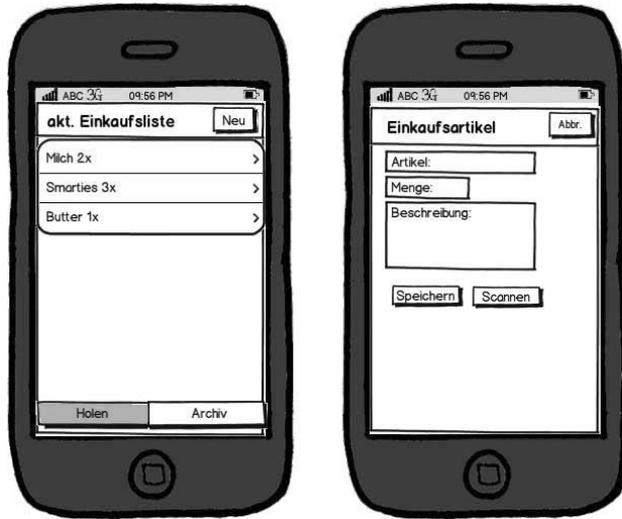


Abb. 2-1 Zwei Views aus einem Storyboard mit Balsamiq Mockups

Für diesen Schritt ist PhoneGap nicht gedacht und bietet auch keinerlei Tool zur Unterstützung.

2.3 Einen ersten Prototyp entwickeln

Gehen wir davon aus, dass sich nach der Idee und dem ersten Mockup der Wille, eine App zu entwickeln, weiter verfestigt hat. Dann erstellen Sie nun am besten einen Prototyp. Das Mock-up hat zwar durchaus Prototypcharakter, aber es ist nun einmal keine Anwendung auf einem Smartphone.

Es ist schon erstaunlich, wie schnell man bei einer echten Umsetzung auf Bedienungsprobleme oder fehlende UI-Komponenten stößt. Da es eine Cross-Plattform-Anwendung werden soll, gibt es hier die Vorgabe, HTML zu nutzen. PhoneGap verpackt ja HTML-Dateien in eine native Anwendung für Ihr Smartphone. Daher nutzen Sie bereits jetzt schon für unseren Prototypen HTML. Damit können Sie das gewünschte App-Gerüst nach dieser Phase weiter verwenden. Eventuelle statische Verlinkungen oder Datenquellen können dann in der Implementierungsphase durch Code dynamisiert werden. Ein Prototyp gibt Ihnen Feedback, ob die gewünschten Elemente und Abläufe vom Mock-up auch in der Praxis funktionieren.

Bei einem Prototyp sind bereits viele Fragen zu klären, die über die App-Struktur und die zu nutzenden Tools Aufschluss geben, angefangen bei der Oberfläche.

2.3.1 Das UI-Framework festlegen

Die Frage, welches Tool bzw. Framework für die Oberflächengestaltung eingesetzt wird, ist nicht leicht zu beantworten. Hier legen Sie fest, ob Sie das User Interface (UI) selbst mit HTML erstellen wollen oder sich von Bibliotheken, wie z.B. jQuery Mobile oder twitter-bootstrap, unterstützen lassen.

Häufig wird angenommen, dass PhoneGap auch für diesen Teil einer Anwendung zuständig ist. Dem ist aber nicht so, da PhoneGap nichts mit der App-Oberfläche zu tun hat. Aus Kapitel 1 wissen Sie ja, dass es um den Hardwarezugriff und das Verpacken in native Anwendungspakete geht. Da PhoneGap den integrierten Browser des Smartphones nutzt, können Sie eigentlich jede HTML-Bibliothek verwenden. Der Vorteil liegt darin, dass mobile UI-Bibliotheken darauf optimiert sind, Elemente für kleine Bildschirme und für die Touch-Bedienung darzustellen. Das heißt, Sie erstellen wie gewohnt Ihre HTML-Elemente und Formulare, und diese werden per CSS und JavaScript von den Frameworks auf die mobilen Geräte angepasst.



Abb. 2-2 Vergleich Standard-HTML zu jQuery Mobile

Um Ihnen die Entscheidung für ein Framework zu erleichtern, möchte ich Ihnen folgende Bibliotheken empfehlen:

■ jQuery Mobile²

Dieses Open-Source-Projekt wurde 2009 ins Leben gerufen und im Oktober 2010 wurde das erste Release 1.0a veröffentlicht. Obwohl es eine Alpha-Version war, wurde es damals schon häufig verwendet. Das liegt unter anderem an der einfachen Verwendung für den Entwickler. Das Ziel von jQuery Mobile ist es, ein Framework zu bieten, das möglichst gut mit allen mobilen Plattformen und Browsern zusammenarbeitet. Wird ein Feature von einem älteren Browser nicht unterstützt, dann ist die Webseite immer noch nutzbar, nur die Optik ändert sich. Im Zusammenhang von jQuery Mobile spricht man dabei von *Mobile Graded Browser Support*³. Dabei wird eine HTML-Seite mittels sogenannter HTML-Data-Attribute erweitert, um sich automatisch an den jeweiligen Browser anzupassen und das bestmögliche Ergebnis zu liefern. Dabei wird versucht, die optimale mobile Touch-Unterstützung zu bieten. Das heißt z.B., dass HTML-Listenelemente wie ein ``-Tag durch jQuery Mobile in eine typische Listendarstellung auf dem Smartphone umgewandelt werden. Die Bibliothek wird gerne in der Web-App-Entwicklung eingesetzt. Die von jQuery Mobile gelieferte Oberfläche ist eine eigene und versucht nicht, Android oder iOS-Aussehen nachzuahmen.

■ Sencha Touch⁴

Neben dem bekannten jQuery Mobile gibt es noch eine andere sehr beliebte Bibliothek für mobile Web-App-Oberflächen: Sencha Touch. Sie wird von der Firma Sencha entwickelt und vertrieben. Die Bibliothek kann sowohl mit einer kommerziellen Lizenz erworben werden wie auch in einer GPL-Variante.

Für den Entwickler ist die Bibliothek grundsätzlich anders zu handhaben als bei jQuery Mobile. Hier programmieren Sie mit einem MVC-Modell unter JavaScript die HTML-Seiten. Ein deklarativer Ansatz wie bei jQuery Mobile wird hier nicht unterstützt. Das macht den Einstieg für Nicht-Entwickler schwieriger, hat allerdings Vorteile in der Ausführungsgeschwindigkeit gegenüber jQuery Mobile, da auf nachgelagerte DOM-Manipulationen verzichtet wird. Außerdem wird Sencha Touch von einer kommerziellen Firma entwickelt, was den Vorteil von Wartungsverträgen und schnellen Bugfixes mit sich bringt.

■ iUI⁵

Es ist eines der ältesten mobilen GUI-Frameworks, wird als Open-Source-Projekt betrieben und unterliegt der MIT-Lizenz. Hier ging es am Anfang um das Nachstellen einer iPhone-Oberfläche. Ziel ist es, ein HTML-Projekt mit möglichst wenig Codeänderungen auf eine iPhone-Oberfläche für mobile Browser

2. <http://jquerymobile.com/>

3. <http://jquerymobile.com/gbs/>

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

5. <http://www.iui-js.org/documentation/latest/getting-started.html>

umzuwandeln. Obwohl das Apple-Betriebssystem Vorbild ist, läuft diese Bibliothek auch auf anderen Smartphone-Systemen.

Nachdem die Frage der Oberfläche geklärt ist, geht es im Folgenden um die Featureliste. Was muss hier beachtet werden?

2.3.2 Die richtigen Tools für die Features auswählen

Für die einzelnen zu implementierenden Features wird hier nach der jeweils anzuwendenden Lösung gesucht. Im Folgenden gehen wir durch die Featureliste und suchen nach Hilfestellungen für die Implementierung.

- Verfügbar als Cross-Plattform-Anwendung
PhoneGap unterstützt Sie beim Verpacken Ihrer Anwendung in das jeweilige Smartphone-Format. Das hierfür benötigte Setup der Entwicklungsumgebungen für die verschiedenen Plattformen finden Sie in diesem Buch in Kapitel 3.
- Auch ohne Netz nutzbar (offline)
Bei diesem Feature kann PhoneGap nur bedingt helfen. Sie können mit PhoneGap feststellen, ob das Smartphone eine Online-Verbindung hat oder nicht, schauen Sie dafür in Kapitel 4. Hier finden Sie den Zugriff auf die jeweiligen Sensoren oder auch Gerätedaten. Für das notwendige lokale Speichern von Daten – hier: Einkaufsartikel – sollten Sie sich allerdings überlegen, ob Sie den vom HTML5-Standard zur Verfügung gestellten `localStorage`⁶ge nutze (hier werden Schlüssel-Wert-Paare gespeichert) oder ob Sie lieber eine SQL-basierte Datenbank verwenden wollen, die sogenannte `webSQL`, die aber nicht auf allen Plattformen verfügbar ist. Natürlich richtet sich diese Entscheidung auch nach Ihrem Datenvolumen. Für die Einkaufsliste wird am Anfang eine kleine Sammlung an JavaScript-Objekten per `localStorage` ausreichen.
- Synchronisation der Daten mit der Cloud
Hier kann PhoneGap leider auch nicht helfen. Die Logik, um Ihre Plattform mit einem Clouddienst wie `API-O-MAT`⁸ in Deutschland oder `PARSE`⁹ in Amerika zu synchronisieren und damit Daten in der Cloud zu halten, müssen Sie selbst per JavaScript implementieren.
- GPS-Daten der Einkaufsartikel erfassen
Bei diesem Feature können Sie PhoneGap nutzen. Beim Einkauf der Artikel soll ja erfasst werden, wo sie gekauft wurden. Daher ist der Zugriff auf den GPS-Sensor notwendig. Wie Sie die GPS-Daten auf Ihrem Gerät abfragen, erfahren Sie in Kapitel 4.

6. http://www.w3schools.com/html/html5_webstorage.asp

7. <http://www.w3.org/TR/webdatabase/>

8. <http://www.apiomat.com/>

9. <https://parse.com/>

■ Artikel-Barcodes scannen

Um Artikel nicht immer per Hand einzugeben, sollen die Strichcodes gescannt werden. Bei diesem Feature zeigt sich die Erweiterbarkeit von PhoneGap. Direkt wird das Scannen von Barcodes nicht unterstützt. Mittels Erweiterungen – sogenannten Plugins – kann diese Funktionalität aber hergestellt werden. Wie Sie Plugins mit PhoneGap nutzen, erfahren Sie in Kapitel 7. Wenn Sie allerdings den Strichcode durch das Plugin erhalten haben, benötigen Sie noch einen Onlineservice, der Ihnen die Produktbeschreibung liefert. Darum müssen Sie sich leider selber kümmern.

Damit sind die Features durchlaufen und die dafür notwendigen Aufgaben besprochen. Lassen Sie uns nun sehen, was in der Implementierungsphase ansteht und wie uns PhoneGap dort unterstützen kann.

2.4 Die App implementieren

Jetzt kommen wir zur eigentlichen Implementierung der Anwendung.

PhoneGap steht Ihnen dabei zur Seite. In Kapitel 3 finden Sie für die jeweiligen Zielplattformen das notwendige Einrichten der Entwicklungsumgebungen. Was aber ist zu tun, wenn Ihnen z.B. für ein Windows-Phone-8-Projekt kein Windows-PC zur Verfügung steht? Dafür gibt es einen Cloudservice, der Ihnen das Kompilieren abnimmt. Schlagen Sie dann für mehr Informationen in Kapitel 8 nach.

2.4.1 Die Entwicklungsumgebung einrichten

Eventuell habe Sie dies bereits beim Prototypen schon für Ihre Lieblingsplattform erledigt, aber Sie sollten es auch für alle anzustrebenden Plattformen durchführen. So können Sie nach jedem Implementierungsschritt die Ergebnisse sofort durch PhoneGap verpacken lassen und auf den verschiedenen Plattformen ausprobieren. Das Einrichten der jeweils nötigen Entwicklungsumgebung können Sie in Kapitel 3 nachschlagen.

2.4.2 Den App-Lifecycle berücksichtigen

Viele Entwickler vergessen das sogenannte Lifecycle-Modell bei mobilen Applikationen. Damit ist der Lebenszyklus einer Anwendung gemeint, der sich stark von dem einer Desktop- oder Webanwendung unterscheiden kann. Eine mobile Anwendung kann z.B. jederzeit durch einen Anruf unterbrochen oder durch den Benutzer geschlossen werden. Damit Sie von diesen Ereignissen nicht überrascht werden, unterstützt Sie PhoneGap auch in dieser Hinsicht. Was zu beachten und zu implementieren ist, lesen Sie in Abschnitt 4.2.

2.4.3 Eine Versionsverwaltung für den Code nutzen

Sicherlich ist die Versionsverwaltung ein Aspekt, den viele Entwickler bei ihrer Arbeit bereits berücksichtigen, aber es ist trotzdem wichtig, immer wieder darauf hinzuweisen: Nutzen Sie Tools wie Git, Subversion, CVS oder Mercurial, um Ihren Code zu versionieren. Nichts ist schlimmer als ein Editorfehler, der Ihnen die Codierungsarbeit einer ganzen Klasse zunichte macht. Auch das Arbeiten mit anderen Entwicklern zur gleichen Zeit kann damit strukturiert werden.

Dieser Aspekt der Softwareentwicklung ist natürlich komplett unabhängig von mobilen Applikationen, er sollte bei jeder Softwareentwicklung berücksichtigt werden. Wenn Sie zu diesem Thema mehr erfahren wollen, schauen Sie doch einmal in das Git-Buch von Bjørn Stachmann und René Preißel: »Git. Dezentrale Versionsverwaltung im Team – Grundlagen und Workflows.«

2.4.4 Die Anwendung testen lassen

Die Überschrift dieses Abschnitts sagt eigentlich schon alles. Aber bitte denken Sie daran, Ihre Anwendung nicht nur selbst zu testen, sondern auch von anderen Personen testen zu lassen. Gerade als Entwickler übersehen Sie leicht Fehler in der Bedienung oder kommen einfach nicht auf verschiedene Nutzungsarten, die vielleicht nicht im Code berücksichtigt wurden. Hierfür kann PhoneGap leider keine Unterstützung leisten. Je nach Smartphone-Plattform müssen Sie die erstellten Binär-Pakete dann an die Tester verteilen. Hierfür gibt es technische Unterstützung von Diensten wie TestFlight¹⁰ oder Zuhium¹¹. Was passiert, wenn dann auf einem Gerät ein Fehler gemeldet wird und Sie ihn als Entwickler finden müssen, sehen Sie im nächsten Abschnitt.

2.4.5 Die Fehler finden

Sicherlich passiert es auch den Besten, dass sich ein Fehler in die Anwendung schleicht und sie ihn suchen müssen. Auch bei dieser Aufgabe hilft Ihnen PhoneGap. Leider ist es häufig so, dass Fehler nur auf einem echten physischen Gerät auftauchen. Um diese Fehler zur Laufzeit einfacher zu finden, hilft Ihnen PhoneGap mit einem Tool. Dafür lesen Sie am besten Kapitel 6 und lernen das Debuggen einer App.

Alle Fehler sind gefunden und die Testuser sind zufrieden – jetzt ist die Anwendung fertig. Im finalen Schritt geht es dann um die Auslieferungen in die App Stores.

10. <https://testflightapp.com/>

11. <https://www.zuhium.com/>

2.5 Die App über die Marktplätze ausliefern

Nun ist Ihre Anwendung fertiggestellt. PhoneGap hilft Ihnen zwar, die Anwendung in das jeweilige native Plattformformat zu verpacken. Die Auslieferung in die virtuellen Marktplätze der Anbieter müssen Sie aber nun selbst vornehmen. Dabei gibt es je nach Anbieter Verschiedenes zu beachten.

Es ist nicht Ziel dieses Buches, die einzelnen Prozesse der Zulassung zu erklären, da sie sich im Detail häufig ändern. Trotzdem möchte ich Ihnen an dieser Stelle wenigstens einige Hilfestellungen für diese Aufgabe geben.

Generell benötigen Sie einen Entwickler-Account bei dem jeweiligen Anbieter. Dieser ist entweder kostenlos oder kostenpflichtig. Bevor Sie dann Ihre Anwendung bei dem Anbieter hochladen und verifizieren lassen, muss diese noch signiert werden. Damit wird Ihre Anwendung quasi mit Ihnen als Hersteller verbunden. Die notwendigen Zertifikate bekommen Sie mit Ihrem Entwickler-Account. Je nach Anbieter muss dann nach dem Signierungsprozess noch ein Onlineformular ausgefüllt werden. Danach heißt es für Sie warten. Je nach Anbieter wird Ihre App intensiv geprüft, was einige Zeit dauern kann. Danach erhalten Sie dann eine E-Mail mit der Botschaft, dass Ihre App nun verfügbar ist.

Nachfolgend möchte ich Ihnen zu den jeweiligen Marktplätzen eine kleine Einführung zum Zulassungsprozess geben.

2.5.1 Apples App Store

Apples App Store ist sicherlich eines der populärsten Ziele für Apps. Damit dies gelingt, benötigen Sie einen Entwickler-Account von Apple, der zurzeit für Einzelpersonen 79 € pro Jahr kostet. Firmen zahlen übrigens deutlich mehr.

Der Zugang ist im Developer-Portal¹² von Apple zu beantragen. Nach wenigen Stunden bekommen Sie dann Zugriff und können Ihre Anwendungen mittels eines Zertifikates nicht nur im Simulator von XCode, sondern auch auf Ihren iOS-Geräten laufen lassen. Damit die von Ihnen erstellte App auch in den App Store kommt, müssen Sie diese an Apple schicken. Dafür melden Sie sich mit Ihrem Entwickler-Account bei iTunes Connect¹³ an. Hier läuft dann der Zulassungsprozess für den App Store. Nutzen Sie die Funktion *Add new App* und folgen Sie den Anweisungen. Apple ist sehr streng, was den Zulassungsprozess angeht und welche Voraussetzungen erfüllt werden müssen. Allerdings finden Sie im Internet immer wieder gute Hilfestellungen für die sich immer wieder ändernden Voraussetzungen.

2.5.2 Google Play Store

Der zweite große Marktplatz neben Apple wird von Google betrieben und heißt Google Play. Ähnlich wie bei Apple müssen Sie hier auch ein Entwicklerkonto er-

12. <https://developer.apple.com/devcenter/ios/index.action>

13. <https://itunesconnect.apple.com>

stellen, wobei eine einmalige Gebühr von aktuell 25 US-\$ fällig wird. Danach erhalten Sie ein Entwicklerzertifikat, mit dem Ihre Anwendung zu signieren ist. Diese Datei können Sie dann zur Veröffentlichung im Play Store hochladen. Dafür hat Google ein Webportal namens Google Play Developer Console¹⁴ vorgesehen. Nach dem Upload Ihrer App folgen Sie den Anweisungen der Webseite. Meistens wird hier schon nach wenigen Minuten Ihre Anwendung zugelassen, denn die Kriterien sind nicht so streng wie bei Apple. Darunter leidet allerdings auch manchmal die Qualität der vorhandenen Apps.

2.5.3 Microsoft Store

Auch bei Microsofts Store wird ein Entwicklerkonto benötigt. Als Ausgangspunkt für Entwickler steht hier das Windows Phone Developer Center¹⁵ zur Verfügung. Ein Account wird hier nur benötigt, um eine Anwendung in den Microsoft Store zu bringen. Dafür werden aktuell 99 US-\$ pro Jahr fällig. Für die Registrierung ist eine Windows LiveId notwendig. Nach der Registrierung bekommen Sie auch hier Zertifikate zum Signieren Ihrer Anwendung. Das weitere Vorgehen wird im Abschnitt *Publish* auf den Webseiten im Phone Developer Center beschrieben.

2.5.4 BlackBerrys App World

BlackBerry bietet mit der App World einen eigenen Marktplatz. Dabei geht RIM als Hersteller einen radikal anderen Weg als seine Mitbewerber. Sie benötigen zwar einen sogenannten Vendor Account, aber dieser ist kostenfrei erhältlich. Dafür müssen Sie sich nur im BlackBerry App World Vendor-Portal¹⁶ anmelden. Außerdem benötigen Sie noch sogenannte Signing Keys, die Sie auf den Entwicklerseiten¹⁷ von BlackBerry kostenfrei anfordern können. Eine damit signierte App kann in das Vendor-Portal hochgeladen und nach dem Ausfüllen eines Formulars in der AppWorld zugelassen werden. Hier wird auch die App geprüft, allerdings nicht so restriktiv wie bei Apple. Zusätzlich können Sie eine Zertifizierung Ihrer Anwendung für das Build-for-BlackBerry-Siegel beantragen.

2.5.5 Samsung Hub (Bada)

Der Samsung Hub¹⁸ ist für die Bada-Geräte der bevorzugte Marktplatz. Hier wird zwischen Anbieter und Entwickler unterschieden. Das heißt, Sie müssen nicht zwingend ein Entwickler sein, um Anwendungen einzustellen und zu verkaufen. Die Anwendung muss durch einen sogenannten Anbieter erfolgen. Damit

14. <https://play.google.com/apps/publish>

15. <http://dev.windowsphone.com/en-us/publish>

16. <https://appworld.blackberry.com/issvportal>

17. <https://www.blackberry.com/SignedKeys/codesigning.html>

18. <http://www.samsungapps.com/>

Sie ein solches Konto erhalten, registrieren Sie sich auf der Partner-Programm-Webseite¹⁹ von Samsung. Danach erhalten Sie Zugang und können alle weiteren Schritte online ausführen, damit Ihre Anwendung bald im Hub zu finden ist.

2.5.6 HP App Catalog (webOS)

Sicherlich ist das webOS von Hewlett Packard nicht die populärste Plattform, aber auch diese hat natürlich einen Store, um Anwendungen für den Palm Pre und das TouchPad bereitzustellen. Damit Sie dort Apps einstellen können, benötigen Sie einen Developer Account, der kostenfrei im webOS Developer Center²⁰ zu beantragen ist. Danach erhalten Sie auch hier Zugang zu einem Online-Portal, mit dem Sie Anwendungen zur Veröffentlichung anmelden können.

2.6 Zusammenfassung

Sie haben nun die nötige Theorie für das Erstellen einer Cross-Plattform-Anwendung erhalten. Von der Idee einer App bis zum Erscheinen im App Store sind Sie theoretisch durch alle Schritte der Anwendungserstellung gegangen. Dabei haben Sie die Punkte identifiziert, die zu beachten sind, und Sie wissen, was mit der Unterstützung von PhoneGap realisiert werden kann. Auch wissen Sie nun, in welchen Kapiteln Ihnen dieses Buch für welche Aufgaben hilfreich zur Seite steht.

Wenn Sie Themen wie jQuery Mobile vertiefen möchten, empfehle ich Ihnen das Buch »Web-Apps mit jQuery Mobile« von Philipp Friberg. Sollten Sie merken, dass Sie das eine oder andere in JavaScript auffrischen möchten oder einen Einstieg benötigen, sollten Sie in das Buch »JavaScript für Enterprise-Entwickler« von Oliver Ochs schauen. Beide Bücher sind im dpunkt.verlag erschienen.

19. <http://www.samsungapps.com/earth/about/becomeSeller.as>

20. https://developer.palm.com/index.php?option=com_user&view=login&signup