

# Business Technology

Architektur, Innovation & Strategie



**Achim Himmelreich:**  
„Der Mensch wird  
noch mobiler.“

## MOBILITÄT

**ÜBERLEBEN IM WIRTSCHAFTSDARWINISMUS**

**MOBILE ENTERPRISE-APPS**

**KULTURWANDEL IM UNTERNEHMEN**

Erfahrungsbericht über die mobile Plattform CarCloud

# How-to: make a mobile App

„Zwei Dinge auf der Welt werden ohne Architekten gebaut: Hundehütten und Software“.

Dieses Zitat von Rainer Janßen (Group Information Executive, Münchener Rück-Gruppe) wird fast jeder, der in der Unternehmens-IT tätig ist, als zutreffend empfinden. Auch wenn jedem, der sich mit Softwareanwendungen befasst hat, schnell klar werden dürfte, dass die Erstellung von Software komplex ist, wird der Planung und Dokumentation der benötigten Funktionen zu wenig Beachtung geschenkt. Ein Grund dafür liegt darin, dass Software in der Planungsphase oft wie ein kleiner und gut zu überschauender Themenkomplex wirkt. Sobald die ersten Teile fertig gestellt sind, tauchen dann aber die Fragen auf und plötzlich ist die kleine, einfache Lösung ein ausgewachsenes Softwareprojekt.



Was für herkömmliche Anwendungen gilt, trifft auch auf mobile Anwendungen zu. Eine tiefergehende Betrachtung ist notwendig, denn neben den allgegenwärtigen mobilen Spielen und kleinen Helfern entstehen nun vermehrt mobile Geschäftsprozesse, für die folglich die Spielregeln von Unternehmenssoftware gelten. Am Beispiel des von uns durchgeführten Projekts CarCloud (Kasten: „Car-Cloud“) wollen wir unsere Erfahrungen mit der Entwicklung einer solchen geschäftsprozessorientierten Lösung für mobile Endgeräte beschreiben und Hinweise geben, wie Fallstricke zu umgehen sind. Am Beginn unseres Erfahrungsbericht steht eine simple Feststellung: Obwohl das Projektteam aus sehr erfahrenen IT-Experten bestand, haben wir die Anforderungen an die neue Plattform am Beginn des Projekts grob unterschätzt. Heute nach einigen Monaten Entwicklung haben wir erkannt, dass es keine Abkürzungen oder Tricks für mobile Entwicklung gibt. Realistische Einschätzungen, gesunder Menschverstand und robuste Entwurfspraktiken sind auch bei mobilen Projekten ein Muss für den nachhaltigen Erfolg.

**FACHKONZEPTION**

Noch immer scheitern mehr IT-Projekte, als dass sie erfolgreich abgeschlossen werden. Ein wesentlicher Grund dafür ist, dass die Anforderungen an die Lösung nicht ausreichend beschrieben wurden. Denn nur wenn klar ist, was am Schluss abgeliefert werden soll, kann ein zufriedenstellendes Projektergebnis erreicht werden.

Diese Erkenntnis ist zugegebenermaßen nicht ganz neu. Wer sich nun für ein strukturiertes Management der Anforderungen entscheidet, steht allerdings schnell vor einer Reihe Entscheidungen. Zu klären sind z. B. die Fragen der Methodik und des Werkzeugeinsatzes.

Es gibt viele Wege, den benötigten Leistungsumfang einer Software zu beschreiben. Die Anforderungen können in Anekdoten-Form aufgenommen werden – so genannte „User Stories“ – oder es werden grafische Modelle zur Abbildung der Fachprozesse erstellt. Wenn die Frage der Methodik geklärt ist, stellt sich die Frage des Werkzeugs. Eine Möglichkeit ist die Anforderungsdokumentation mit Office Tools. Dies hat z. B. den Vorteil, dass die Inhalte von jedem Mitarbeiter ohne Hürden bearbeitet

werden können. Es entstehen keine zusätzlichen Kosten für die Schulung der Anwender oder für Lizenzen. Auf der anderen Seite haben Office Tools aber den Nachteil, dass diese Werkzeuge kaum Struktur vorgeben.

Sollen spezialisierte Werkzeuge zum Einsatz kommen, z. B. UML oder BPMN-Editoren, hilft das bei der Strukturierung der Anforderungen durch die Mittel, die die verschiedenen Lösungen bei der Erfassung der Inhalte erfordern. Auf der anderen Seite führt dies natürlich zu Einschränkungen, da nicht jedes Tool alles so zu modellieren erlaubt, wie sich das ein Projekt vorstellt. Das Methoden- und Werkzeug-Know-how der Nutzer ist ein weiterer entscheidender Faktor. Zudem können die Kosten für professionelle Modellierungswerkzeuge auch in kleinen Projekten schnell in die Höhe schnellen. Tabelle 1 zeigt die Vor- und Nachteile von Anforderungsmanagement-Werkzeugen und Office-Anwendungen.

Auch im betrachteten CarCloud-Projekt standen genau diese Entscheidungen an. Um uns zu entscheiden, haben wir die Situation weiter analysiert und sind zu folgenden Einschätzungen gekommen, die nach unserer Auffassung für mobile Projekte nicht untypisch sind:

- Überschaubarer aber nicht trivialer Funktionsumfang. Oft wird auch eine bestehende Lösung für mobile Geräte angepasst und umgestrickt. Daraus folgt ein mittlerer Komplexitätsgrad der Anforderungen, der entsprechende Analyse erfordert.

**CarCloud**

CarCloud ist eine Lösung, die den Abstimmungsprozess zwischen Autobesitzern und Werkstätten erleichtert. In der Anbahnungsphase ermöglicht CarCloud das Auffinden von Werkstätten, das Bearbeiten von Anfragen und Vereinbaren von Aufträgen. Im Verlauf einer Reparatur schafft CarCloud die Transparenz, welchen Stand die Reparatur an einem Fahrzeug gerade hat und wann das fertige Fahrzeug abgeholt werden kann. Auch Verzögerungen oder Störungen in der Bearbeitung lassen sich mit CarCloud schnell und einfach abwickeln.

CarCloud befindet sich zur Drucklegung dieses Artikels in der Testphase und wird Ende 2012 am Markt erscheinen.

	Anforderungsmanagement-Werkzeuge	Office-Anwendungen
Pro	Strukturierung Bewährte Methodik Auswertung	Einfachheit Offenheit Keine zusätzlichen Kosten
Kontra	Einschränkungen Komplexe Bedienung Hoher Preis	Systematik Auswertungsmöglichkeiten Kollaboratives Arbeiten

Tabelle 1: Vor- und Nachteile





Abb. 1: Grafische Darstellung

- Entwicklung für Android und iOS, sodass mehrere Entwicklungsteams mit Anforderungen versorgt werden müssen.
- Die Erfahrungen der Anwender im Mobile Computing sind noch nicht so ausgeprägt wie bei typischen Web- oder Desktopanwendungen. Eine intensive Abstimmung über die Anforderungen ist daher besonders wichtig.
- Die Erwartung der Anwender an „Look and Feel“ und Usability sind durch die Messlatte, die z. B. Apple mit seinen iOS-Applikationen gesetzt hat, sehr hoch. Deshalb sind ergonomisches Interaktionsdesign und

ansprechende, grafische Gestaltung kritische Erfolgsfaktoren.

- Die Innovationsgeschwindigkeit ist im Bereich mobiler Anwendungen besonders hoch. Eine schnelle Time to Market ist daher bei mobilen Projekten besonders wichtig.

Schnell wurde im Projekt klar, dass ohne ein professionelles Anforderungsmanagement die Erfolgsaussichten gering sind. Klassische schwergewichtige Lösungen zur Anforderungsmodellierung einerseits waren zu starr für diesen Typus von Anwendung und hätten hohe Kosten verursacht, andererseits ist eine reine Beschreibung mit Texten schwerer zu vermitteln. Eine Kombination aus unabhängigen Texten und Grafiken ließe sich realisieren, führt aber schnell zu Versionskonflikten und „Aktualisierungsarten“.

Aufgrund der dargelegten Situation haben wir uns für einen semiformalen Ansatz zur Beschreibung der Anforderungen entschieden. Zum Einsatz kam der SOAPARK Modeller, ein webbasiertes Tool zum serviceorientierten Design. Folgende Vorteile haben wir dadurch erzielt:

## CarCloud-Steckbrief

Die CarCloud-Lösung besteht aus zwei Teilen: einer Webapplikation mit umfangreichen Funktionen für die Werkstatt und einer komfortablen, mobilen Anwendung für den Fahrzeughalter auf iOS und Android. Der Geschäftsprozess zwischen Autohaus und Fahrzeughalter wird über einen Cloud-basierten Applikationsserver koordiniert. Auf Seiten der Autohäuser ist eine Anbindung von bestehenden Werkstattssystemen erforderlich.

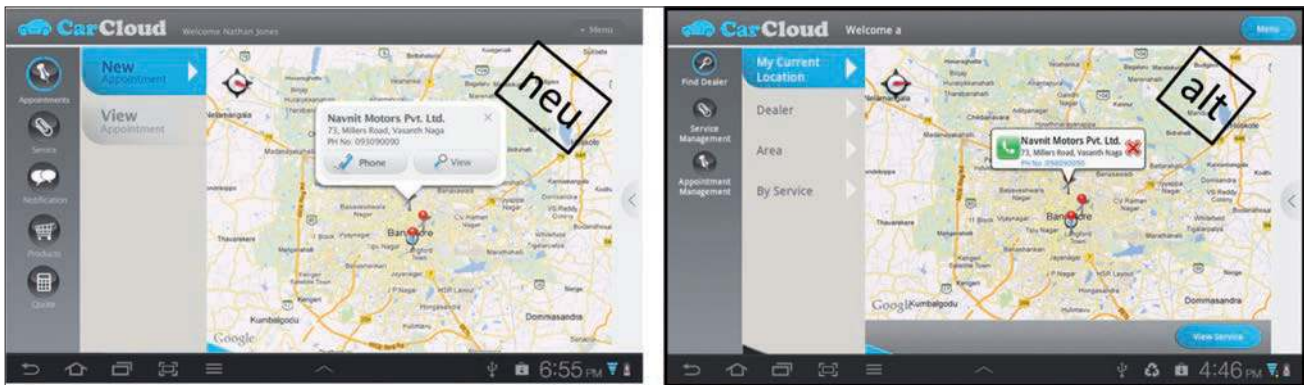


Abb. 2: UI-Design der CarCloud-App

- Das webbasierte Tool ermöglichte die einfache Zusammenarbeit zwischen den verteilten Entwicklungsteams in Deutschland und Indien.
- Die Einfachheit des Ansatzes hat dazu geführt, dass alle Entwicklungsteams die Methodik ohne Widerstände akzeptiert haben.
- Langwierige Schulungen und teure Lizenzen wurden nicht benötigt.
- Die Anforderungen wurden vollständig erfasst.

Unsere Erfahrung zeigt, dass die Kommunikation und die Dokumentation der Anforderungen unumgänglich sind. Hierfür eignen sich grafische Darstellungen wesentlich besser als rein textuelle (Abb. 1). Verwenden Sie allerdings nicht zu viel Zeit auf ausgefeilte Grafiken und Modelle, denn sonst besteht die Gefahr, dass Sie zwar ein methodisch korrektes Abbild der Anforderungen haben, das allerdings keiner versteht, weil es zu komplex ist. Investieren Sie diese Zeit besser in die Abstimmung mit den Kunden und dem Entwicklungsteam. Mit dem gezeigten Kompromiss konnten die Anforderungen sowohl in Richtung der Anwender, als auch zum Entwicklungsteam verständlich kommuniziert werden.

## USER INTERFACE DESIGN

Anwender von Apps sind ein anspruchsvolles Klientel, das ansprechende Gestaltung gewohnt ist. Das Interface-Design für Apps unterscheidet sich aber auch wegen der durch Apple geprägten Formen- und Zeichensprache, der Touch-Steuerung und nicht zuletzt wegen der relativ kleinen Display-Fläche von klassischem Webdesign. Deshalb müssen Sie der Design-Phase besonders große Aufmerksamkeit schenken.

Auch wenn das Design von Apple natürlich gerade aufgrund der hohen Usability so populär ist, prägt es aber auch mit nicht direkt funktionalen Charakterzügen die gesamte mobile Entwicklung. So sind die abgerundeten Rechtecke, mit denen die Apps des iPhone angeordnet

sind, zum Quasi-Standard geworden. Android-Entwickler haben es nicht nur aufgrund der kürzeren Evolution der Plattform schwerer, ein ansprechendes Design zu erstellen. Sowohl für Android als auch für iOS werden den Entwicklern herstellerseitig Interface-Design-Richtlinien an die Hand gegeben, die das Erstellen schöner und nutzerfreundlicher Applikationen erlauben. Für iOS gibt es diese Richtlinien schon lange. Für Android wurde ein solch umfassender Katalog an Design-Elementen erst Anfang 2012 veröffentlicht. Das liegt daran, dass Apple schon lange vor dem ersten mobilen Interface einen starken Fokus auf das Design der Benutzeroberflächen von Mac gelegt hat, was bei Google mit dem Suchmaschinen-Interface eher im Hintergrund stand. Außerdem haben iOS-Entwickler auch weniger Gerätetypen zu bedienen.

Entwickler von Webanwendungen können von wenigen, genormten Bildschirmgrößen ausgehen. Mit den Mobile-Apps kann diese Annahme nicht mehr getroffen werden. Die Anwendung soll möglichst genauso gut auf dem 10"-Tablet-PC im Querformat aussehen wie auf dem 3,2"-Smartphone im Hochformat. Gerade im Android-Umfeld variieren die Displaygrößen und die Auflösung der verschiedenen Geräte stark. Somit ist eine verlässliche Darstellungsqualität nur schwer zu erreichen.

Zu Beginn des Projektes haben wir zunächst – in Erwartung eines ausgereiften App-Designs – Entwürfe von den Entwicklern erstellen lassen. Leider wurde unsere Erwartung nicht erfüllt. Auch wenn die funktionalen Anforderungen abgedeckt waren, entsprach die Anwendung einfach nicht dem gewohnten „Look and Feel“ einer App. Die erste Testgruppe an potenziellen Anwendern bestätigte unseren Eindruck.

Wir entschlossen uns deshalb, für die Oberflächen-gestaltung erfahrene Interaktionsdesigner einzusetzen, die den Entwurf harmonischer und exakt aufeinander abgestimmter Dialoge übernahmen, um eine stimmige Nutzererfahrung zu erreichen. Hierzu zählen zum Beispiel auch wiederkehrende und leicht verständliche

Steuerungselemente sowie eine sinnvolle Positionierung dieser (**Abb. 2**).

Eine solche grundlegende Überarbeitung braucht natürlich Zeit. Da wir diesen Aufwand zunächst unterschätzt haben, sprengte die eingeschobene Design-Phase unseren Projektplan. Die neuen Design-Entwürfe wurden dann zu Mockups zusammen gebaut, das sind Softwareattractanten auf Basis der Bilddateien. So ließen sich die grafischen Änderungen schnell auf ihre Praxistauglichkeit überprüfen und der Bearbeitungsfluss testen. Aus unserem Projekt haben wir folgende Best Practices ableiten können:

- **Professionelle Designer einbeziehen:** Das Design ansprechender Benutzeroberflächen ist eine Kunstform. Das gilt aus bereits genannten Gründen auch für das App-Design.
- **Mit einem Gerät starten:** Beginnen Sie mit einem Gerät, auf das Sie das Design mit Ihrem Interface-Designer abstimmen. Erst bei einem stabilen Design sollten Sie auf andere Geräte/Formate wechseln.
- **Testen Sie das Design:** Mockup-Lösungen erlauben auch in frühen Stadien, das Design auf den Prüfstand zu stellen. Durch ein Verknüpfen der Oberflächenent-

würfe zu einer Softwareattractante lassen sich schnell überladene Darstellungen oder unpraktisch angebrachte Steuerelemente identifizieren.

## REALISIERUNG

Im Bereich der Entwicklung von mobilen Apps haben sich bis heute noch keine klaren Standards etabliert. Anders als bei klassischer Webentwicklung gibt es auch bei den Entwicklungsumgebungen große Unterschiede, die zu unterschiedlichen Ergebnissen führen.

Auf der einen Seite des Spektrums sind native Entwicklungsumgebungen zu betrachten, die für die jeweilige Plattform iOS oder Android vorgesehen sind. Auf der anderen Seite gibt es Cross-Plattform-Umgebungen mit generativen und hybriden Ansätzen sowie HTML5, die die Entwicklung von iOS- und Android-Apps mit einer einzigen Codebasis ermöglichen.

Wir haben uns im CarCloud-Projekt für die native Entwicklung für iOS und Android entschieden und damit in Kauf genommen, dass die Codebasis für iOS und Android voneinander unabhängig gepflegt werden muss. Folgende Vorteile haben uns zu dem Ansatz veranlasst:

# Anzeige

- **Qualität:** Die Unterschiede zwischen iOS und Android können von heutigen Entwicklungsumgebungen nicht befriedigend ausgeglichen werden. Unsere Anwender haben von uns die bessere Performance und das besser an die Plattform angepasste Aussehen einer nativen Implementierung gefordert.
- **Lernkurve:** Nach unserer Erfahrung sind Cross-Plattform-Ansätze mit zahlreichen Tücken gespickt. Daraus folgen erhöhter Einarbeitungsaufwand bzw. zusätzliche Schulungen. Gleichzeitig kann man auf Wissen um die nativen Fähigkeiten der Plattformen in der Praxis nicht verzichten.
- **Fokus:** Wir haben uns dafür entschieden, unsere Testanwender zunächst mit der Android-Version zu konfrontieren und erst danach den Aufwand der iOS-Implementierung anzugehen. Diese Entscheidung hat sich als sehr positiv herausgestellt, da eine parallele Testphase unser Team überfordert hätte.

In einem anderen Projekt mit weniger anspruchsvollen Anwendern hätten wir uns vermutlich anders entschieden. Die Wahl des Entwicklungsansatzes hängt stark von den Anforderungen ab.

## ARCHITEKTUR

Mobile Applikationen stehen, wenn es um verteilte Geschäftsprozesse geht, nicht alleine in der Welt. CarCloud ist hierfür ein typisches Beispiel, da auf Seite der Autohäuser z. B. Terminplanungssysteme, ERP-Lösungen oder Kassensysteme zu integrieren sind. Allein die Verteilung des Geschäftsprozesses auf die mobile Applikation des Fahrzeughalters und die Webapplikation im Autohaus führt gradewege in eine Multi-Kanal-Architektur.

Da die Bereitstellung einer öffentlichen Plattform mit individuellen Integrationspunkten ein wesentliches Entwurfsziel von CarCloud war, erkannten wir sehr schnell, dass jede halbherzige Architektur schnell in eine Sackgasse führen musste. Die typischen Anforderungen an redundanzfreie, ausfallsichere und erweiterbare Unternehmensarchitekturen mussten bei der Umsetzung von CarCloud Anwendung finden.

Das bedeutete, dass der Architektorentwurf mit einer Facharchitektur, also dem Mapping von Use Cases auf fachliche Komponenten, beginnen musste. Diese Aufgabe konnte mit dem SOAPARK Modeller erfüllt werden. Danach ging es darum, die Multi-Kanal- und die Integrations-Fähigkeit in den technischen Entwurf aufzunehmen.

Beispielsweise war die direkte Verdrahtung von Oberflächenelementen mit Datenbankfeldern keine Option für die finale Implementierung. Aus diesem und

anderen Gründen haben wir uns letzten Endes dazu durchgerungen, eine Middletier in der Cloud bereitzustellen, die Geschäftsobjekte hinter Web Services kapselt. Dieser Ansatz ist zwar vordergründig aufwändiger, für eine skalierbare, robuste und ausbaufähige Lösung aber unerlässlich.

Ein weiterer Unterschied zu traditioneller Entwicklung war das Thema Security. Hier galt es, höhere Anforderungen im Vergleich zu Webanwendungen umzusetzen und gleichzeitig achtsamer mit Ressourcen wie Netzwerkbandbreite oder CPU umzugehen.

## ZUSAMMENFASSUNG

Die Eleganz vieler mobiler Applikationen im Umfeld Telefonie oder Spiele verleitet dazu, an die Entwicklung von mobilen Geschäftsprozessen spielerisch heranzugehen. Allerdings zeigt sich in der Praxis, dass es keine magischen Abkürzungen für mobile Softwareprojekte gibt. Seriöses Anforderungsmanagement, robustes Softwaredesign sowie eine wohlüberlegte Architektur sind für den Erfolg von mobilen Projekten ebenso entscheidend wie für andere Projekte in der Unternehmensarchitektur. Zusätzlich zu den traditionellen Entwurfsdisziplinen muss ein mobiles Projekt auf die besonderen Erfordernisse der grafischen Gestaltung der Benutzeroberfläche und Interaktionssteuerung eingehen.



### Dr. Dirk Krafczig

ist Gründer von SOAPARK. Als Sprecher auf Konferenzen und Autor von Artikeln und Büchern gilt Dr. Krafczig als ein Protagonist der serviceorientierten Architektur (SOA) und hat maßgeblich zu der Begriffsbildung in diesem Bereich beigetragen. Insbesondere die SOA-Fallstudien mit der Deutschen Post, Credit Suisse, Halifax Bank of Scotland und Winterthur

Versicherung in seinem Bestseller „Enterprise SOA“ haben viel Aufmerksamkeit auf sich gezogen. Derzeit arbeitet Dr. Krafczig in einem strategischen SOA-Programm bei einem Mobilfunkanbieter an dem Thema Security.



### Sacha Baulan

ist unabhängiger Business-Analyst mit Schwerpunkt auf serviceorientierter Analyse. Er hat in den letzten Jahren im Wesentlichen in den Branchen Versicherung, Versorger und im öffentlichen Sektor gearbeitet.