

Generierung mobiler Dokumentation nach dem Single-Source-Prinzip

1 Einführung

Vor Feierabend noch mit der Navi-App kurz nachschauen, ob es auf der Heimfahrt Stau gibt, eine Nachricht mit der Messenger-App nach Hause schicken, dass man sich zehn Minuten verspätet und nebenbei noch kurz mit der Wetter-App prüfen, ob das Wetter für die Grillparty durchhält. Mobile Endgeräte mit ihren spezialisierten Anwendungen, den sogenannten Apps, sind für viele Menschen zu einem selbstverständlichen Teil des Alltags geworden.

Die Autofahrt beginnt. Auf dem Weg nach Hause blinkt ein unbekanntes Symbol bedrohlich im Cockpit. Wäre es jetzt nicht naheliegend, sein Smartphone aus der Tasche zu ziehen, eine App zu öffnen und nach der Lösung zu suchen? Erwarten die Nutzer nicht schon so etwas wie mobile Dokumentation?

Die Benutzung mobiler Endgeräte steigt rasant an und revolutioniert das Nutzerverhalten (Grünwied 2014). Die Technische Dokumentation muss sich dieser Veränderung nicht nur stellen, sie hat mit ihren Methoden und Systemen sogar die besten Voraussetzungen dazu. Content-Management-Systeme (CMS), die basierend auf dem Single-Source-Prinzip in verschiedene Medien und Formate publizieren, sind in vielen Unternehmen bereits Standard und das Potenzial der medienneutralen Datenhaltung ist noch lange nicht ausgeschöpft. Im Gegenteil: Es bietet sich die Chance, der Technischen Dokumentation im Gewand einer attraktiven App zu neuem Glanz zu verhelfen.

Doch welche Möglichkeiten bieten sich für das Generieren mobiler Dokumentation nach dem Single-Source-Prinzip? Wie können Inhalte aus einem CMS wiederverwendet und in eine App-Variante der Dokumentation eingebunden werden? Und kann man so etwas einfach als weitere Publikationsstrecke betrachten?

2 Ziele des Beitrags

Im folgenden Text sollen die Möglichkeiten untersucht werden, mobile Dokumentation aus bestehendem Content nach dem Single-

Source-Prinzip zu generieren. Der Artikel konzentriert sich dabei auf das Publizieren von mobilen Apps, die auf Smartphones und Tablets lauffähig sind. Neben den zugrundeliegenden Methoden und Ideen werden auch beispielhafte Prozesse dargestellt, um eine Publikationsstrecke für mobile Apps zu implementieren. Technische Grundlage für alle Varianten sind hierbei moderne und frei verfügbare Webtechnologien aus dem HTML5-Umfeld sowie publikationsfähige Content-Management-Systeme.

3 Ausgangslage

3.1 Single Sourcing

Die kontrollierte Wiederverwendung von Modulen, das sogenannte Single-Source-Prinzip, ist eine der wichtigsten Grundlagen effektiven Content Managements (Drewer/Ziegler 2011, 296). Als Module werden hierbei Informationseinheiten bezeichnet, die von der Satz- bis auf die Kapitelebene gebildet werden können und auf eine Wiederverwendung ausgelegt sind.

Neben einer Steigerung von Qualität und Konsistenz des Inhalts können so Übersetzungskosten gesenkt und die Time-to-Market verkürzt werden. Den Kern der Methode bildet das Prinzip des „Single Point of Truth“,¹ also die Vermeidung unkontrollierter Redundanzen von Modulen durch nur eine Ablagestelle in der Datenhaltung bei gleichzeitigem Zulassen von kontrollierter Wiederverwendung in der Publikation (was dann als Single Source Publishing bezeichnet wird).

Die Vorteile des Single Sourcing überwiegen klar den erhöhten Mehraufwand, weshalb das Prinzip in vielen Redaktionen bereits fest zur Arbeitsweise gehört. Auch für die Erstellung mobiler Dokumentation erweist sich modularer Content als ideale Ausgangsbasis, denn die Darstellung auf Endgeräten mit kleinem Bildschirm erfordert eine Einteilung in kompakte ‚Informationshappen‘ und eine Abkehr von einer printorientierten Seitendarstellung. Ob die Größe der bereits erstellten Module auch geeignet für die Darstellung auf mobilen Endgeräten ist, muss zunächst festgestellt werden. Ein guter Ansatzpunkt sind hierbei die aus dem Bereich der Online-Hilfe bekannten Topics, also inhaltlich abgeschlossene Einheiten, die in unterschiedlichen Kontexten verwendbar sein sollen.

Ideale Ausgangsbasis

Beste Voraussetzungen sind in Redaktionen, die das Single-Source-Prinzip anwenden, in jedem Fall gegeben. Doch welche Mittel werden benötigt, um mobile Dokumentation aus bestehenden Inhalten zu generieren und die Publikation in den Redaktionsprozess mit einzubinden?

¹ In englischsprachiger Literatur auch als „Single Source of Truth“ bezeichnet.

3.2 Content-Management-Systeme

Klassisches Single Source Publishing wird im Bereich der Technischen Dokumentation gewöhnlich mit einem Content-Management-System umgesetzt, das den Bereich der Datenverwaltung zu großen Teilen abdeckt. Das darauf aufbauende medien-spezifische Publizieren des medienneutralen Inhalts bezeichnet man als Cross Media Publishing. Das Single-Source-Prinzip kann auch ohne CMS angewendet werden, solange die methodischen Grundlagen eingehalten werden und der Quellinhalt modular aufgebaut ist und strukturiert abgelegt wird.

Vielseitige Werkzeuge

Neben der reinen Datenhaltung stellen moderne CMS auch Werkzeuge zur Erstellung und Versionsverwaltung von Content sowie ggf. ein integriertes Workflow- und Übersetzungsmanagement bereit. Content-Management-Systeme erleichtern den Umgang mit Modulen erheblich und sind deshalb erste Wahl, wenn es um die effektive Umsetzung von Single Sourcing geht.

Warum also diese Vorteile nicht nutzen, um auch die Inhalte der mobilen Dokumentation zu verwalten und Kosten zu sparen? Und kann aus CMS-Content einfach eine App generiert werden?

3.3 Publishing

Die Publikation aus einem Content-Management-System erfolgt in den meisten Fällen durch eine Transformation der im XML-Format abgelegten Inhalte in ein bestimmtes Zielformat, wie z. B. PDF für Printdokumentationen oder HTML für webbasierte Online-Hilfen.

Aufbereitung für mobile Dokumentation

Um Inhalte aus einem CMS für die Verwendung in einer webbasierten App aufzubereiten, reicht es in den meisten Fällen, die Inhaltsmodule als schlanke XML- oder HTML-Dateien zu publizieren. Diese werden dann bei Bedarf in ein sogenanntes Template geladen, das den funktionalen und gestalterischen Rahmen der App definiert. Bei der Publikation sollte dabei nur die Struktur der Inhaltsmodule angepasst werden. Alle gestalterischen Anpassungen werden ausschließlich über das Template oder externe CSS-Stylesheets vorgenommen.

3.4 Mobile Dokumentation

Das Thema ‚Mobile Dokumentation‘ ist in vielen Redaktionen im Gespräch, doch die zahlreichen Umsetzungsmöglichkeiten lassen Verantwortliche noch zurückschrecken (Leicht 2013). Besonders interessant erscheint die Möglichkeit, aus einem CMS heraus und mit bestehendem Content mobile Apps zu generieren. Die Vorteile eines solchen Vorgehens liegen auf der Hand:

Vorteile automatisierter Generierung

- Kosteneinsparung durch Verwendung bestehender Inhalte und Systeme
- Einhaltung des Single-Source-Prinzips

- Schnelle Aktualisierbarkeit der Inhalte
- Automatisierbarkeit der Publikation

Doch ist das überhaupt möglich? Unterscheidet sich das Medium App nicht zu sehr von den klassischen Publikationen der Technischen Dokumentation?

4 König Content

4.1 Dokumentation als App?

Mobile Dokumentation tritt in verschiedensten Formen und Ausprägungen als mobile App auf: In manchen Fällen dient die eigentliche Applikation nur als Anzeigeprogramm (Viewer) für die Inhalte aus dem CMS, in anderen Fällen wird sie um aufwendige Zusatzfunktionen erweitert.

Die zahlreichen Multimedia- und Interaktionsmöglichkeiten, die moderne Smartphones und Tablets bieten, lassen das alleinige Anzeigen des meist printorientierten CMS-Contents als verschenktes Potenzial erscheinen. Oft besteht der Wunsch, dass Videos, interaktive Grafiken oder technische Gimmicks wie QR-Code-Scanner oder Augmented-Reality-Anwendungen den vollen Leistungsumfang der mobilen Endgeräte ausschöpfen, um Kunden einen echten Mehrwert gegenüber einer Printvariante bieten zu können.

Dieser Umstand stellt sich als größtes Hindernis für die automatisierte Generierung mobiler Dokumentation heraus. Multimedia-Elemente, Skript-Komponenten oder interaktive Menüs lassen sich nur begrenzt in Content-Management-Systemen verwalten; die jeweiligen Erstellungsumgebungen besitzen oft keine Schnittstelle zum CMS. Wie können diese Inhalte unter Einhaltung des Single-Source-Prinzips in eine App integriert werden?

Hindernisse für
Automatisierung

Neben der Zielgruppe des beschriebenen Produkts spielt aber auch der Anspruch des Herstellers eine große Rolle, denn aufwendig entwickelte Apps haben in der Regel auch ihren Preis. Welche Möglichkeiten gibt es, den CMS-Inhalt so anzureichern, dass er app-tauglich wird? Wie können die Zusatzkosten gering gehalten werden?

4.2 Mehr herausholen

Attraktive mobile Dokumentationen leben vor allem von der Darbietung ihrer Inhalte. Werden die Möglichkeiten des neuen Mediums App ausgenutzt, werden die Nutzer öfter dazu verleitet, die Anwendung zu öffnen und einen Blick in die mobile Dokumentation zu werfen. Eine größere Nutzerzufriedenheit kann eine gute Doku-App auch zum Marketinginstrument werden lassen.

Doch oft scheitern App-Projekte im Umfeld mobiler Dokumentation an einer fehlenden Automatisierbarkeit der Publikation.

Welche Wege gibt es, eine App attraktiver zu gestalten bei gleichzeitigem Erhalt der Automatisierbarkeit?

4.2.1 Content Retrieval

Eine gängige Praxis, die schon im Bereich der Online-Hilfen angewendet wird, besteht darin, die eigentlichen Inhalte durch verschiedene Retrieval-Funktionen, wie Volltextsuchen oder hierarchische Menüs zu erweitern.

Integriert in ein App-Template lassen sich diese Funktionen wie ein Rahmen anwenden und haben auf die publizierten Inhalte nur wenig bis keinen Einfluss. Doch ist nicht gerade die attraktive Darstellung des Inhalts die Königsdisziplin einer App?

4.2.2 Content Enhancement

Strukturierter Content aus einem CMS kann durch geschickte Transformationen automatisiert mit interaktiven Funktionen angereichert werden. Das funktioniert umso besser, je semantischer das zugrunde liegende Informationsmodell ist. Basierend auf Auszeichnungselementen kann der Inhalt so systematisch durch Funktionen erweitert werden:

- Ausklapptexte enthalten Zusatzinformationen oder werden zu sogenannten Akkordeons kombiniert, um Texte zu komprimieren.
- Inhaltsfilter erleichtern dem Nutzer das Finden der gewünschten Information auf Basis von semantischen Merkmalen.
- Tabellen werden sortierbar oder lassen sich nach Werten durchsuchen.
- Grafiken können vergrößert werden; CAD-Daten lassen sich im Raum drehen und manipulieren.
- Legendeneinträge werden interaktiv, indem ein Fingertipp den entsprechenden Teil der Grafik hervorhebt.
- Tooltips können nach Fingertipp auf eine Bezeichnung z. B. Zusatzinformationen aus einem Ersatzteilkatalog anzeigen.

Bei der Publikation kann eine Transformation die Grundlage der Funktionen in HTML umsetzen. Die eigentliche Interaktivität wird durch inhaltsneutrale Skripting-Komponenten ergänzt.

4.3 Smartphone-Funktionen nutzen

Neben inhaltszentrierten Erweiterungen kann auch auf Funktionen der Endgeräte zurückgegriffen werden, um eine mobile Dokumentation interaktiver zu gestalten. Dazu kann auf ein umfangreiches Repertoire an Schnittstellen aus dem HTML5-Umfeld zurückgegriffen werden. Der Zugriff auf Ortungsdienste und Beschleunigungssensoren der Hardware sowie das Initiieren von Telefonaten und Nachrichten kann neben weiteren Funktionen wie dem Ansteuern von Kamera oder Vibrationsalarm in eine App integriert werden.

In mobilen Dokumentationen lassen sich diese Funktionen nicht immer ohne Weiteres sinnvoll einsetzen, doch über kleine Er-

Automatisierte
Anreicherungen

Mehrwert für
Anwender

weiterungen, die für Anwender einen echten Mehrwert darstellen, sollte immer nachgedacht werden. So ist es denkbar, die in der Einleitung herbeigesehnte App mit mobilen Funktionen zu ergänzen:

- Automatische Lokalisierung der nächsten Werkstatt
- Klickbare Rufnummern zu Service/Pannendienst
- Unterstützte Erkennung von Cockpit-Symbolen (per Kamera)

Die aufgeführten Funktionen können nur teilweise automatisch generiert werden, in den meisten Fällen ist Maarbeit gefordert. Werden die Gerätefunktionalitäten jedoch unabhängig vom Inhalt in einem App-Template untergebracht, haben sie keinen Einfluss auf eine automatisierte Publikation und ihr Entwicklungsaufwand fällt nur einmal an.

4.4 Multimedia-Inhalte einbinden

Neben der Funktion als mobile Dokumentation möchten viele Firmen ihre App auch als erweitertes Schaufenster für Marketingmaterialien nutzen. So besteht der Wunsch, Produktvideos oder 3D-Illustrationen zu integrieren, die nicht im CMS verwaltet werden. In der Regel haben solche Dateiartern ihre eigenen Verwaltungssysteme mit eigener Versionsverwaltung (z. B. MAMS) oder liegen im Dateisystem. Können diese externen Inhalte in den Publikationsprozess integriert werden oder sollte in diesem Fall die Erstellung komplett außerhalb des CMS erfolgen?

Externe Inhalte integrieren

4.5 Die goldene Mitte

Erfolgreich umgesetzte Projekte verschiedener Branchen zeigen, dass wie so oft die goldene Mitte der richtige Weg zu sein scheint: Klassischer Modulcontent kommt aus dem CMS; Spezialinhalte und Skript-Komponenten werden in einem zusätzlichen Publikationsschritt manuell eingearbeitet. Kann dabei auf festgelegte Ablagestrukturen oder andere Verwaltungssysteme zurückgegriffen werden, wird auch das Single-Source-Prinzip nicht verletzt. Das Ausspielen der App kann als weitere Publikationsstrecke im CMS angelegt werden und die App mit einer halbautomatischen Nachbearbeitung endgültig veröffentlicht werden (Abb. 3).

Doch soll eine App überhaupt als statische Publikation betrachtet werden? Werden damit nicht die eigentlichen Vorteile eines so dynamischen Mediums verspielt?

5 Der Weg aufs Endgerät

5.1 Immer online?

Ein Kriterium, mit dem sich mobile Apps einteilen lassen, ist die Abhängigkeit von einer Internetverbindung. Sogenannte Offline-Apps

werden mit dem gesamten Inhalt publiziert und auf das Endgerät geladen, was hohe Zuverlässigkeit, aber auch großen Speicherverbrauch zur Folge hat. Online-Apps hingegen beziehen ihre Inhalte per Internet, sind dadurch aktuell und flexibel, können jedoch bei einer fehlenden Netzverbindung nur sehr eingeschränkt oder gar nicht benutzt werden.

5.2 Die Auslieferung

Die Unterscheidung zwischen Online- und Offline-Apps ist eng mit zwei Paradigmen verknüpft, die eine ähnliche Einteilung aus der Sicht des Content Managements treffen. Man differenziert hier die beiden Konzepte „Content Publishing“ und „Content Delivery“.

5.2.1 Content Publishing

Beim Content Publishing wird der Content aus dem CMS ausgespielt, zusammengestellt und verpackt. Als gängigstes Beispiel dient das klassische Print-PDF. Sollen die enthaltenen Informationen aktualisiert werden, muss das PDF neu publiziert und ausgetauscht werden. Im App-Bereich kann dies mit einem Update der App im App-Store verglichen werden. Eine neue Version muss aktiv durch den Nutzer installiert werden. Die Stärken des Verfahrens liegen in der eindeutigen Publikationsversionierung, einer garantierten Verfügbarkeit und der Erfahrung, die in der Technischen Dokumentation vorhanden ist.

5.2.2 Content Delivery

Content Delivery bezeichnet hingegen die Bereitstellung der jeweiligen Informationsmodule in dem Moment, in dem der Leser sie anfordert. Dem gleichen Prinzip folgen Webserver, die eine Website erst im Moment des Aufrufs an den Nutzer senden. Diese On-Demand-Auslieferung wird immer häufiger von agilen Softwareherstellern mit schnellen Entwicklungszyklen in ihren Online-Hilfen eingesetzt. Durch die Möglichkeit, Inhalte zentral zu aktualisieren, bleibt die Dokumentation stets auf dem neuesten Stand.

Da Inhalte beim Single Source Publishing modulbasiert vorliegen, können nun auch Module einzeln publiziert werden. Dadurch reduziert sich der nötige Netzwerkverkehr auf die tatsächlich geänderten Module und nicht wie beim Publishing auf die gesamte Dokumentation. Einige Systemhersteller bieten bereits Content-Delivery-Erweiterungen für ihre CMS an.

5.3 Clever kombiniert

Von einer reinen Content-Delivery-Lösung, also einer Online-App, ist im Bereich der Technischen Dokumentation auf Grund der potenziellen Unzuverlässigkeit und der dadurch unklaren rechtlichen Lage (z. B. DIN EN ISO 12100 2011 Ziffer 6.4.5.3e) abzuraten. Als eine sehr flexible Lösung erweist sich dagegen die Kombination der beiden Methoden: Der Content wird zunächst in einer initialen Versi-

Effiziente Auslieferung
der Inhalte

on mit der App ausgeliefert. Diese prüft in regelmäßigen Abständen, ob Aktualisierungen oder neuer Content vorliegen, und bezieht diese dann bei einer bestehenden Internetverbindung von einem Content-Delivery-System (idealerweise dem CMS selbst).

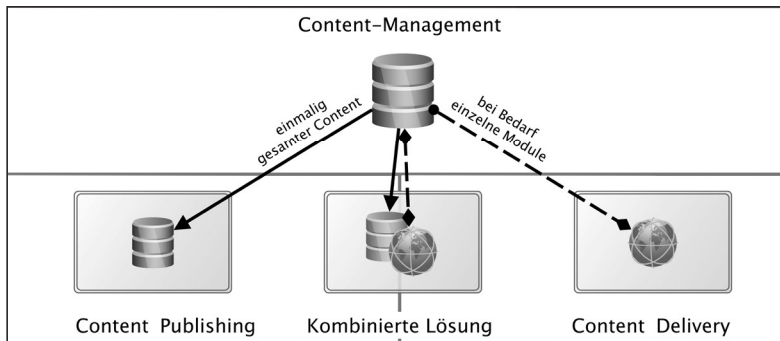


Abb. 1: Verschiedene Varianten der Auslieferung von Content an den Endanwender

6 Mobile Apps

6.1 Durchblick im App-Dschungel

Aus technischer Sicht unterscheidet man drei verschiedene Bauweisen von mobilen Apps: native, hybride und Web-Apps. Webbasierte Apps sind plattformunabhängig, basieren vollständig auf Webtechnologien und werden im Browser ausgeführt. Native Apps sind speziell für eine Plattform programmiert und besitzen erweiterten Zugriff auf Hardware und Funktionen des Endgeräts (Schober 2012).

Hybride Apps sind native Rahmen für Web-Apps, die dadurch Zugriff auf erweiterte Funktionen der Hardware erlangen. Diese Funktionen werden über einheitliche und webbasierte Schnittstellen von Frameworks² wie Apache Cordova oder Adobe PhoneGap in der Entwicklungsphase bereitgestellt und bei der Publikation in ihre plattformspezifischen Äquivalente umgewandelt. In Kombination mit UI-Frameworks³ wie jQuery Mobile, Ionic oder Sencha Touch können so relativ einfach hybride Apps mit Webtechnologien erstellt werden, die sich in ‚Look and Feel‘ von einer nativen App kaum unterscheiden.

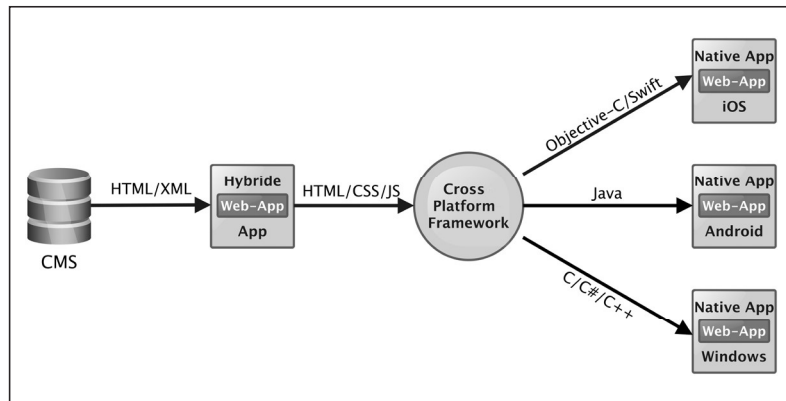
Hybride Apps unterscheiden sich im Publikationsprozess nur geringfügig von webbasierten Lösungen und sind dadurch prädestiniert für den Einsatz in einer Publikationsstrecke. In der zukünftigen Entwicklung der mobilen Dokumentation spielen sie eine zentrale Rolle (Ziegler 2013).

Hybride Apps als
Mittelweg

² Die Bezeichnung solcher Frameworks ist noch nicht vereinheitlicht. In dieser Arbeit werden sie nach ihrer Ausrichtung als „Cross Platform Frameworks“ bezeichnet.

³ UI-Frameworks stellen Werkzeuge zum Erstellen von Benutzeroberflächen bereit.

Abb. 2: Generierung von nativen Varianten einer hybriden App (mit Technologien)



6.2 Nicht nur effizient: Die hybride App

Doch warum überhaupt eine hybride App generieren, wenn eine moderne Web-App die meisten Anforderungen an die Darstellung von mobiler Dokumentation schon erfüllt? Steht der zusätzliche Publikationsschritt in einem positiven Verhältnis zum späteren Nutzen?

Viele Gründe sprechen dafür, die Extra-Meile zu gehen und sich näher mit dem Thema hybride Apps zu beschäftigen. Die drei dominierenden Betriebssysteme Android, iOS und Windows Phone beherrschen im Jahr 2014 zusammen ca. 99 % des Markts (IDC 2014) und betreiben jeweils eigene App-Stores. Für Nutzer sind diese App-Stores der zentrale Anlaufpunkt zum schnellen und unkomplizierten Bezug von Apps; für Anbieter stellen sie eine beliebte Plattform zum Vertrieb der eigenen Lösung dar.

In das Angebot der App-Stores aufgenommen werden jedoch nur native Apps, die den plattformspezifischen Vorgaben der Anbieter entsprechen. Soll die eigene App auch für den Großteil der Smartphone-Nutzer sichtbar und auffindbar werden, führt der Weg nur über die offiziellen App-Stores und damit über eine native bzw. hybride App.

Doch nicht nur die Präsenz im virtuellen Verkaufsregal spricht für native Apps. Auch ein fester Platz des Icons auf dem Bildschirm des Nutzers kann durch den einfachen Zugriff zu häufigerer Nutzung führen. Zwar erlauben fast alle Betriebssysteme, Internet-Links als sogenannte Web-Clips auf dem Bildschirm abzulegen, doch nur wenige kennen oder nutzen diese Funktion.

Native Apps bauen eine engere Bindung zum Nutzer auf und werden tendenziell öfter verwendet als webbasierte Varianten mit demselben Funktionsumfang. Auch das Prestige einer eigenen App spielt oft eine große Rolle. Welche Bedeutung dem im Rahmen einer mobilen Dokumentation zugemessen wird, muss individuell entschieden werden.

App-Stores als Vertriebsplattform

7 Aus vielem eines

Bei den zahlreichen Möglichkeiten und Varianten, die es zur Erstellung mobiler Apps im Umfeld der Technischen Dokumentation gibt, kann kein allgemeingültiges Vorgehen definiert werden. Betrachtet man jedoch die typischen Anforderungen einer Redaktion an eine zumindest teilautomatisierte Generierung von Apps, dann stellen sich einige Lösungswege als besonders geeignete Kombinationen der Teilschritte heraus.

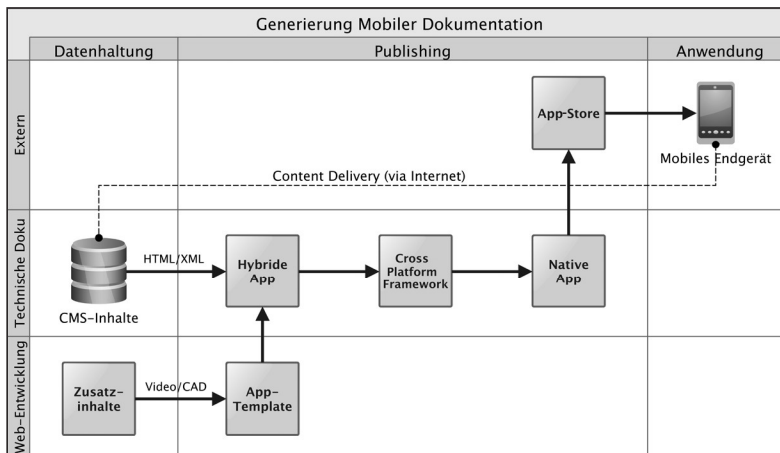


Abb. 3: Eine mögliche Publikationsstrecke für mobile Dokumentations-Apps

In dem in Abb. 3 gezeigten Beispiel werden die zuvor erläuterten Best Practices angewendet: Aus dem CMS werden die Inhalte zunächst modulweise als HTML ausgespielt. Die Inhaltsdateien werden mit externen Multimediainhalten und einem vorgefertigten App-Template zusammengeführt und anschließend durch ein Cross Platform Framework als native App in einem App-Store veröffentlicht.

Die Inhalte der mobilen Dokumentation werden im Gesamten auf das Endgerät publiziert, Aktualisierungen können nachträglich über das Internet auf das Endgerät eingespielt werden. Varianten dieser Lösung sind das Erstellen einer Online-App, die ohne Inhalte ausgeliefert wird und diese erst beim Start der App per Content Delivery bezieht oder das Publizieren einer Web-App auf einem eigenen Webserver. Hierbei entfällt die Umwandlung mit einem Cross Platform Framework.

Viele Wege möglich

8 Fazit

Das Generieren mobiler Dokumentation ist nicht nur möglich, es lohnt sich auch. Moderne Webtechnologien und weit entwickelte

Cross Platform Frameworks verkürzen den Entwicklungsprozess erheblich und erlauben zumindest eine Teilautomatisierung des Publikationsprozesses. Das Single-Source-Prinzip kann mit geringem Aufwand auch auf komplexe Apps übertragen werden; eine Integration in bestehende Redaktionsprozesse wird dadurch zum Kinderspiel. Ist nun noch das passende App-Konzept für die eigenen Anforderungen gefunden, steht einer mobilen Dokumentation nichts mehr im Weg.

9 Literatur

- Adobe (2014): PhoneGap: About the Project. <http://phonegap.com/about> [23.07.2014].
- Apache Software Foundation (2014): Apache Cordova. <http://cordova.apache.org> [23.07.2014].
- DIN Deutsches Institut für Normung e.V. (2011): DIN EN ISO 12100. Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze – Risikobeurteilung und Risikominderung. Berlin: Beuth.
- Drewer, Petra / Ziegler, Wolfgang (2011): Technische Dokumentation. Übersetzungsgerechte Texterstellung und Content-Management. Würzburg: Vogel.
- Drifty Co. (2014): Ionic: Advanced HTML5 Hybrid Mobile App Framework. <http://ionicframework.com/> [23.07.2014].
- Grünwied, Gertrud (2014): Die Online-Hilfe wird mobil. In: technische kommunikation 2, 31-34.
- IDC (2014): Smartphone OS Market Share, Q1 2014. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> [23.07.2014].
- jQuery Foundation (2014): jQuery Mobile. A Touch-Optimized Web Framework. <http://jquerymobile.com> [23.07.2014].
- Leicht, Jerome (2013): Mobile Produktinformation. In: technische kommunikation 4. <http://www.tekom.de/fachartikel/elektronische-informationsprodukte/mobile-produktinformation.html> [23.08.2014].
- Schober, Martin (2012): Mobil, mehrsprachig und multimedial – Grundlagen von HTML5. In: technische kommunikation 6. <http://www.tekom.de/fachartikel/elektronische-informationsprodukte/mobil-mehrsprachig-und-multimedial.html> [23.08.2014].
- Sencha Inc. (2014): Sencha Touch. Mobile App Development Framework. <http://www.sencha.com/products/touch/> [23.07.2014].
- Ziegler, Wolfgang (2013): Content Management in der Technischen Kommunikation. Ein Überblick. In: Hennig, Jörg / Tjarks-Sobhani, Marita (Hrsg.): Content Management und Technische Kommunikation. Stuttgart: tcworld (= tekom Schriften zur Technischen Kommunikation 18) 11-25 .