



Ulrike Parson

(E-Mail: upa@parson-com.com)

ist Technische Redakteurin für Software und seit 2006 Eigentümerin des Hamburger Dienstleistungsunternehmens parson communication. Sie schreibt seit 16 Jahren Dokumentation für IT-Lösungen, sowohl für Endanwender als auch Entwickler.

IM TEAM SCHREIBEN: DOKUMENTATION VON TECHNISCHEN SCHNITTSTELLEN MIT WIKIS

An die Dokumentation einer Web-Services-API, die von Entwicklern bei externen Kunden und Partnern genutzt wird, werden hohe Anforderungen gestellt: Sie soll stets aktuell und vor allem verständlich sein, damit sie korrekt und zweckgemäß genutzt wird. Dabei ist zu berücksichtigen, dass die Entwickler die Dokumentation auf ganz unterschiedliche Art und Weise nutzen. Leider tun sich viele Entwickler mit dieser zielgruppenorientierten Art der Dokumentation schwer. Der Artikel stellt einen interdisziplinären Ansatz vor, der die kollaborativen Möglichkeiten eines Wiki-Systems nutzt, um gemeinsam eine API-Dokumentation zu erstellen und zu pflegen.

Dieser Artikel richtet sich an Entwickler, die eine technische Programmierschnittstelle (API) entwerfen und den Anwendern dieser Schnittstelle Dokumentation liefern „dürfen“. Er richtet sich aber auch an diejenigen, die in ihrem Unternehmen für die (technische) Dokumentation von Softwareprodukten verantwortlich sind. Beiden Zielgruppen soll der Bericht zeigen, wie sie ihre jeweiligen Stärken in die Erstellung und Pflege der Dokumentation einbringen können. Dazu versetzen wir uns im Folgenden gedanklich in ein Beispielunternehmen: die *Best Software GmbH* (im Folgenden kurz *Best*).

Dokumentation einer extern genutzten Web-Services-API

Best stellt ein Softwareprodukt namens *Fantastik* her. *Fantastik* wird agil entwickelt, um schnell auf die Änderungswünsche der Kunden reagieren zu können. Kleine, interdisziplinär besetzte Teams erstellen alle zwei Wochen eine neue Version der Software. Nicht jede dieser Versionen wird tatsächlich ausgeliefert. Erst wenn ein marktfähiger Mix an neuen Funktionen verfügbar ist, wird ein neues Release veröffentlicht und an die Kunden und Partner von Best ausgeliefert. Diese nutzen *Fantastik* intensiv und liefern immer

wieder neue Ideen an die Produktmanager der Firma Best. Die Fülle neuer Ideen liegt in der offenen Schnittstelle von *Fantastik* begründet, über die Kunden und Partner das Softwareprodukt sehr gezielt nutzen und es in einigen Bereichen sogar erweitern können. Diese Schnittstelle ist als Sammlung von Web-Services implementiert. Die Web-Services-API muss für die Nutzung bei den Kunden und Partnern dokumentiert werden.

Im Rahmen der internen Produktentwicklung wird der Code mithilfe von Javadoc dokumentiert (vgl. [Ora]). Javadoc ist ein Software-Dokumentations-

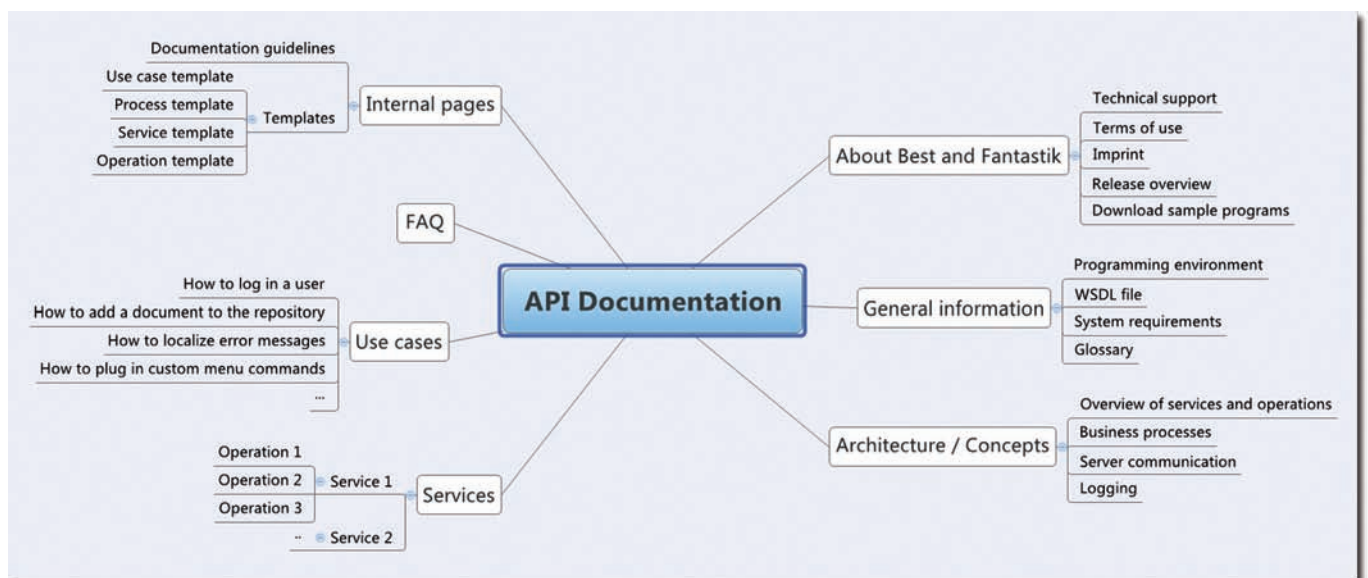


Abb. 1: Wiki-Struktur als Mindmap.

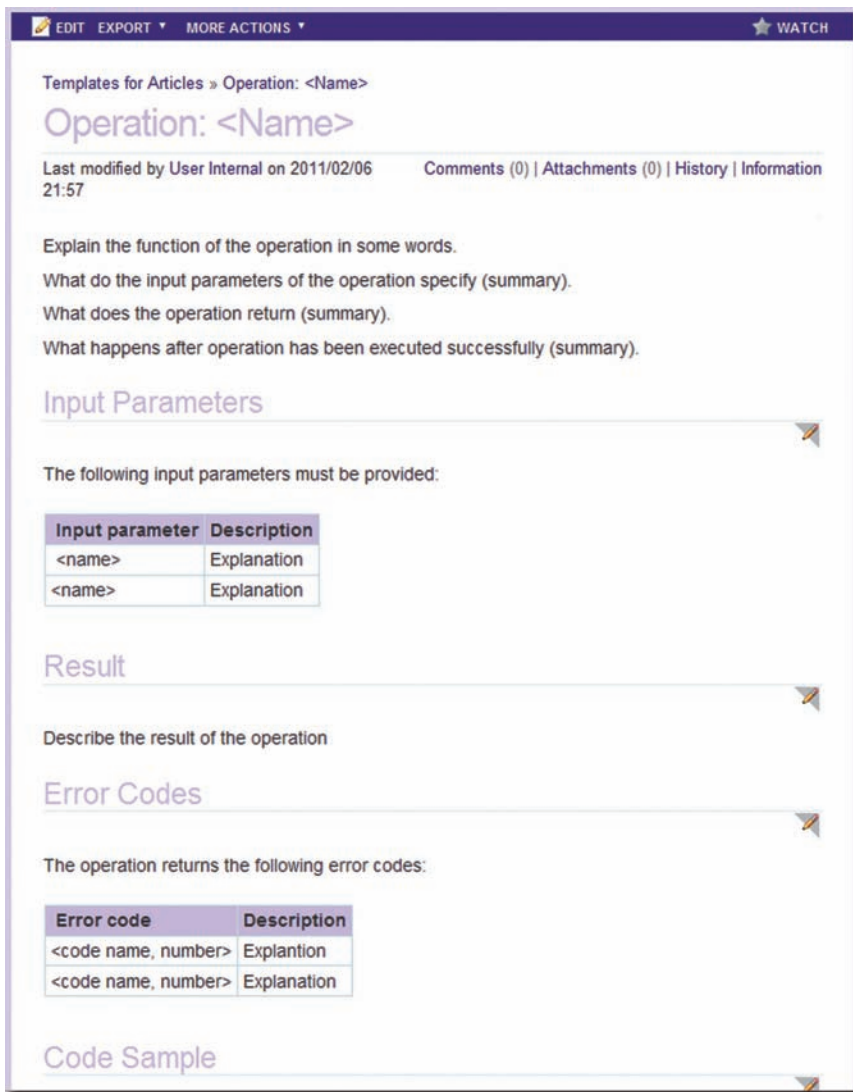


Abb. 2: Beispiel-Inhaltsvorlage für die Beschreibung von Operationen.

werkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt. Die im internen Code angewendeten Standards für die Dokumentation sollen auch für die externe API gelten. Allerdings können bei einer Web-Services-API keine Javadoc-ähnlichen Code-Kommentare eingesetzt werden. Best möchte seinen Kunden und Partnern außerdem die Möglichkeit bieten, Feedback zur API und deren Dokumentation zu geben – und das möglichst einfach und schnell.

Die technische Lösung: Ein Wiki

So entstand die Idee, die Dokumentation der API in Form eines Wiki verfügbar zu machen. Der offensichtlichste Vorteil eines Wiki-Systems ist die schnelle Aktualisierung. Ein Wiki-Artikel kann direkt über

die web-basierte Benutzungsoberfläche bearbeitet werden. Die Grundsyntax für die Formatierung der Wiki-Seiten ist einfach und kann schnell erlernt werden. Viele Wikis bieten darüber hinaus umfangreiche Formatierungsmöglichkeiten, beispielsweise für Tabellen. Damit kann die API-Dokumentation schnell und einfach aktuell gehalten werden. Aber auch die zweite Anforderung (Feedback durch Kunden und Partner) lässt sich einfach realisieren, weil bei vielen Wiki-Systemen die Möglichkeit der Kommentierung von Artikeln zum Standard-Funktionsumfang gehört oder einfach nachgerüstet werden kann.

Ursprünglich waren Wikis als offene Kollaborationswerkzeuge konzipiert, bei denen jeder jede Seite nach eigenem Ermessen verändern kann. Wie das bekannteste Wiki-System „Wikipedia“

(vgl. [Wiki-a]) zeigt, hat diese Offenheit jedoch ihre Grenzen. Auch bei der Dokumentation einer kommerziellen API möchte man den Schreibzugriff beschränken. Die meisten Wiki-Systeme bieten ein Rechte- und Rollensystem, das bestimmte Systemfunktionen an Benutzerrollen bindet, die wiederum mit den Wiki-Benutzern assoziiert werden. Darüber hinaus können Artikel mit einem Status versehen werden. Damit lässt sich ein einfacher Workflow für die Erarbeitung und Aktualisierung von Inhalten realisieren.

Ein Wiki ist immer nur so gut wie seine Inhalte, die schnell und zielgerichtet gefunden werden müssen, wenn das Wiki nicht zum Informationsgrab verkommen soll. Durch das Kategorisieren und Verschlagworten der Wiki-Artikel kann die Trefferquote bei der Suche im Wiki deutlich verbessert werden. Daneben muss natürlich auch eine Volltextsuche über den gesamten Inhalt des Wiki möglich sein.

Viele der beschriebenen Funktionen haben die Wiki-Systeme im Laufe der Jahre von ihren großen Geschwistern, den ausgewachsenen *Content-Management-Systemen (CMS)*, gelernt. Tatsächlich erinnern heutige Wiki-Systeme nur noch entfernt an das einfache Wissensverwaltungswerkzeug WikiWikiWeb (vgl. [Wiki-b]), das Ward Cunningham im Jahr 1994 für die Publikation von Entwurfsmustern (*Design Patterns*) entwickelt hatte. Mit Wikipedia und neuerdings WikiLeaks (vgl. [Wiki-c]) hat diese Softwaregattung weltweite Bekanntheit und Beliebtheit erlangt. Aber beide Beispiele zeigen auch deutlich, dass eine solide und einfach benutzbare technische Plattform allein noch nicht ausreicht, um aus einem Wiki eine nützliche Informationsquelle zu machen. Dazu bedarf es einer Gruppe von Redakteuren, die Informationen recherchieren sowie diese zielgruppengerecht aufarbeiten und publizieren. Bei einem solchen Tätigkeitsprofil denken wohl die wenigsten an Softwareentwickler. Diese gelten eher als dokumentationsscheu. Wie aber bekommt Best seine Web-Services-API ins Wiki?

Die fachliche Lösung: Ein interdisziplinäres Team

Für die Umsetzung wird ein Team aus Technischen Redakteuren und Entwicklern gebildet. Beide Gruppen bringen ihr berufsspezifisches Know-how in das Projekt ein:



- Technische Redakteure sind spezialisiert auf die Strukturierung, Formulierung und Verschlagwortung von Informationen sowie zielgruppengerechtes Schreiben.
- Entwickler repräsentieren die Zielgruppe der Dokumentation und liefern das Domänenwissen aus der Softwareentwicklung. In der Regel entwickeln sie auch die Codebeispiele oder Beispielprogramme für die API-Dokumentation.

Das Vorgehen

Schritt 1: Zielgruppenbestimmung

Damit die API-Dokumentation für die Kunden und Partner effektiv nutzbar ist, stellt sich das Team zuerst die Fragen: „Wer nutzt die API? Wer ist unsere Zielgruppe?“ Anhand von Personas werden unterschiedliche Entwicklertypen mit verschiedenen Anforderungen herausgearbeitet:

- *Petra* fängt sofort an zu programmieren, liest nur Codekommentare und benutzt Codeschnipsel und Beispielprogramme als „Abgucker“.
- *Georg* liest erst einmal in Ruhe die Informationen zum technischen Konzept und zur Architektur der API. Wenn er anfängt zu programmieren, liest er Beschreibungen von Programmieraufgaben.
- *Nicolas* liest nur bei Bedarf, d. h. wenn er eine bestimmte Aufgabe lösen muss und dazu Informationen in der Dokumentation sucht.

Die Personas machen klar, dass es in der API unterschiedliche Informationsklassen geben muss:

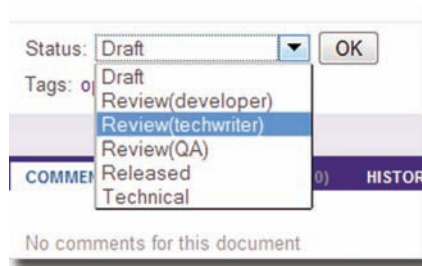


Abb. 3: Artikelstatus setzen.

- Konzeptionelle Informationen zur Architektur und Struktur der API (für Georg).
- Beschreibung von Programmieraufgaben inklusive Codeschnipseln und technischen Prozessen (für Nicolas und Georg).
- Detaillierte Referenzinformationen zu einzelnen Operationen und Services, ähnlich einer Code-Dokumentation mit Javadoc sowie Beispielprogramme (für Petra).

Schritt 2: Wiki-Struktur festlegen

Für die Erstbefüllung des Wikis erarbeiten die Technischen Redakteure eine Grundstruktur für die Wiki-Themen (siehe [Abbildung 1](#)). Die verschiedenen Informationsklassen spiegeln sich in der Grundstruktur wider – so gibt es auf oberster Ebene unterschiedliche Ordner für konzeptionelle Informationen, Code-Informationen und Programmieraufgaben (*Use-Cases*).

Schritt 3: Vorlagen für Informationsklassen erarbeiten

Für jede Informationsklasse erstellt das Team eine Vorlage, damit alle Artikel im Wiki einheitlich strukturiert sind, auch wenn unterschiedliche Personen daran

arbeiten (siehe [Abbildung 2](#)). Neben den strukturellen Vorgaben gibt es Dokumentationsrichtlinien und ein Glossar, das die zu verwendenden Begriffe festlegt.

Schritt 4: Workflow festlegen und technisch umsetzen

Die Dokumentation einer API kann nur zusammen von Redakteuren und Entwicklern geschrieben werden. In unserem Beispiel-Wiki nutzt das Team einen speziellen Artikelstatus, um einen Workflow abzubilden (siehe [Abbildung 3](#)).

1. Der Entwickler legt den Artikel auf Grundlage einer Vorlage an. Während der Bearbeitung bleibt der Artikel im Status „Draft“.
2. Der Entwickler stellt den Artikel auf „Review(techwriter)“ und schickt ihn so zur Überarbeitung.
3. Der Technische Redakteur überarbeitet den Artikel hinsichtlich Strukturierung, Terminologie und Dokumentationsrichtlinien. Anschließend stellt er den Artikel auf „Review(QA)“, damit die Qualitätssicherungsabteilung den Text und das Codebeispiel prüfen kann.
4. Die Qualitätssicherung stellt geprüfte Artikel auf „Released“.

Wenn der Artikel von einem technischen Redakteur angelegt wird, kommt im zweiten Workflow-Schritt der Status „Review(developer)“ zur Anwendung. Alle Beteiligten können mithilfe eines speziellen Plug-Ins nach dem Status suchen und sich so ihre Aufgabenliste zusammenstellen (siehe [Abbildung 4](#)).

Schritt 5: Wiki füllen

Alle Vorbereitungen sind getan – nun kann das Team endlich schreiben. Das Wiki füllt

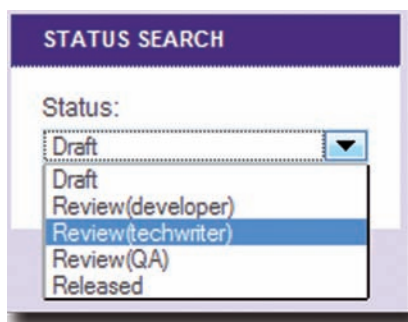


Abb. 4: Artikelstatus suchen und Suchergebnisse anzeigen.



Abb. 5: Tag-Cloud mit Schlagwörtern.

sich langsam und die Qualitätssicherungsabteilung gibt alle Artikel nach und nach frei.

Die Entwickler tragen vor allem Codeschnipsel, Beispielpprogramme und Programmieraufgaben bei. Die technischen Redakteure sorgen für eine saubere Sprache und machen die Informationen im Wiki für die verschiedenen Zielgruppen leichter zugänglich:

- Mithilfe von *Tags* werden Artikel verschlagwortet, um schnell alle relevanten Artikel zu einem Thema zu finden (siehe Abbildung 5).
- Mithilfe von Eltern-Kind-Beziehungen zwischen den Artikeln wird die eigentlich flache Artikelstruktur des Wiki in eine hierarchische Struktur gebracht, um die Themen ähnlich wie in einem Buch anzuordnen. Eine „Brotkrumen-Navigation“ im Kopf jeder Seite weist auf alle übergeordneten Artikel (siehe Abbildung 6).
- Die Hauptnavigation im Wiki wird wie ein Inhaltsverzeichnis aufgebaut.

Schritt 6: Dokumentation veröffentlichen

Nun kann die Dokumentation von Fantastik veröffentlicht werden – dazu werden alle freigegebenen Seiten auf einen öffentlichen Bereich (*Space*) kopiert, auf den die Kunden und Partner von Best zugreifen können. In diesem Space sind die Seiten für Kommentare freigeschaltet, sodass die externen Entwickler ihr Feedback zur Dokumentation geben können (siehe Abbildung 7).



Abb. 6: Brotkrumen-Navigation im Artikelkopf.

Schritt 7: Dokumentation pflegen

Jede Dokumentation lebt länger, als man glaubt. Ein Wiki bleibt aber nur am Leben, wenn seine Inhalte aktuell sind und das Feedback der Anwender kontinuierlich eingepflegt wird. An dieser Stelle kommt der Wiki-Gärtner (vgl. [Fli08]) ins Spiel:

- *Der Wiki-Gärtner sät Ideen:* Er legt neue Artikel an, strukturiert das Wiki bei Erweiterungen der API neu, fügt *Tags* für die Verschlagwortung hinzu, erzeugt und überarbeitet Inhaltsvorlagen und beantwortet Fragen von Entwicklern und Technischen Redakteuren zur Organisation des Wiki.
- *Der Wiki-Gärtner jätet Unkraut:* Er löscht Duplikate und veraltete Artikel, fasst Artikel zusammen, arbeitet User-Feedback ein und löscht unangebrachte oder überholte Kommentare.
- *Der Wiki-Gärtner und alle Wiki-Nutzer ernten die Früchte ihrer Arbeit:* Sie werden belohnt mit einem Wiki voller Wissen.

Bei einem großen Wiki können die Gärtnerarbeiten auf verschiedene Personen aufgeteilt werden, die jeweils einen Bereich des Wiki pflegen. Wichtig ist, dass die Firma Best die Rolle des Wiki-Gärtners fest besetzt hat, sodass sowohl die Nutzer des Wiki als auch die Entwickler und Redakteure bei Best einen festen Ansprechpartner haben. So ist die Hürde, das Wiki zu nutzen, gering. Die Wiki-Inhalte bleiben aktuell und spiegeln das Feedback der Nutzer wider.

Fazit

Unser Beispiel-Wiki zeigt, dass technische Dokumentation eine interdisziplinäre Aufgabe ist, die nur mit dem Know-how des gesamten Teams zu einem nützlichen Ergebnis führt. Da jeder seine Stärken einbringen kann, wird Dokumentation auf diese Weise von den Entwicklern nicht als lästig empfunden. Wikis bieten eine ideale Plattform, um gemeinsam Wissen zu erarbeiten und weiterzugeben. Sie sind auch ideal für agile Prozesse mit häufigen Updates der Dokumentation, da sich Änderungen schnell umsetzen und veröffentlichen lassen. Aber ohne Pflege verkommt auch das schönste Wiki und endet als Informationshalde. Ein Wiki-Gärtner, der das Wiki hegt und pflegt, ist deshalb entscheidend für den langfristigen Erfolg. ■

Literatur & Links

- [Fli08] A. Flicker, K. Schmieder, Tobias Ahlert, Informationen säen, Wissen ernten, in: technische kommunikation 3/2008, siehe: http://www.tekom.de/index_neu.jsp?url=/servlet/ContentrollerGUI?action=voll&id=2416
- [Ora] Oracle, Javadoc Tool, siehe: www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html
- [Wiki-a] Wikipedia, siehe: de.wikipedia.org
- [Wiki-b] Ward Cunninghams WikiWikiWeb, siehe: <http://c2.com/cgi/wiki?WelcomeVisitors>
- [Wiki-c] WikiLeaks, siehe: wikileaks.org

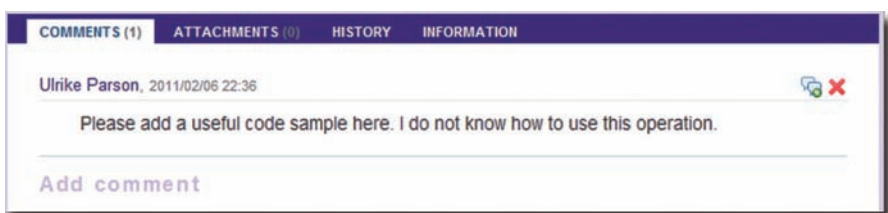


Abb. 7: Kommentare zu einem Artikel.