

# Der Widerspenstigen Zähmung - Dokumentation in APEX Projekten

**Dietmar Aust**  
**Opal-Consulting**  
**Köln**

## **Schlüsselworte:**

Oracle APEX, MS Word, Dokumentation

## **Einleitung**

Die Erstellung von Dokumentation für eine Applikation ist eine sehr unbeliebte Tätigkeit. Es ist langweilig, anstrengend und sie ist bereits veraltet, sobald wir sie erstellt haben.

In dieser Session präsentieren wir ein Vorgehen, um APEX Applikationen zu dokumentieren, ein wichtiger Bestandteil ist die automatische Generierung von Dokumentation basierend auf dem Oracle Data Dictionary sowie dem APEX Data Dictionary.

Dieses Verfahren bietet eine Lösung zu der Fragestellung, wie die Dokumentation einer Applikation sowie deren Änderungen bestmöglich und mit geringem Aufwand dokumentiert werden können. Wie viele Details sind notwendig, wie dokumentieren wir Veränderungen an der Applikation, wie können wir den Ablauf und die Prozesse derart gestalten, dass die Erstellung von Dokumentation leicht fällt?

Für eine standardisierte Vorgehensweise verwenden wir Dokumentenvorlagen und arbeiten sie als Checklisten ab.

Im Rahmen eines Delta-Releases beschreiben wir natürlich im Detail die Änderungen an der Software, jedoch wird selten die Gesamtspezifikation aktualisiert. Durch ein geeignetes Vorgehen kann dies unterstützt werden.

Um in einem MS Word Dokument statische Texte, Diagramme und eingebettete Dokumente gemeinsam mit den aktuellen Definitionen in der Datenbank zu vermischen, haben wir eine wiki-ähnliche Syntax entwickelt. So können wir z.B. mittels (:apex.page page\_id=1 :) oder (:db.table name=DEMO\_ORDERS:) die Struktur und die Kommentare aus dem Data Dictionary per Extraktion direkt in unser MS Word Dokument einbetten. Dies erlaubt eine viel stärkere Ausrichtung der Dokumentation am Source Code als bisher.

## **Hintergrund – die Herausforderungen**

In zahlreichen Software-Projekten sind uns immer wieder folgende Fragestellungen in Bezug auf Dokumentation begegnet:

- Zu wenig oder zu viel Dokumentation, die richtige Granularität ist schwer zu treffen.
- Die Dokumentation veraltet schnell, wird selten auf dem aktuellen Stand gehalten.
- Delta vs. Gesamtdokumentation, wie kann ich die Veränderungen am System beschreiben und gleichzeitig eine vollständige und aktuelle Dokumentation der gesamten Funktionalität pflegen?
- Die Dokumentation muss oft im MS Word Format abgeliefert werden.
- Redundanzen sind sehr schwer aufzulösen, dadurch entstehen Widersprüche in der Dokumentation.

In vielen Projekten wird zu wenig dokumentiert, teilweise sogar gar nicht. Demgegenüber gibt es auch immer wieder Projekte, in denen versucht wird, zu detailliert zu konzipieren und dokumentieren.

## Zielsetzung für eine bessere Lösung

Aus den oben genannten Herausforderungen haben wir folgende Zielsetzungen abgeleitet:

### Die relevanten Teile mit minimalem Aufwand dokumentieren

Wie können wir die den Umfang der Dokumentation so gering wie möglich halten, so dass er immer noch aussagekräftig ist? Dafür müssen wir die relevanten Teile einer Applikation dokumentieren.

### Don't make me think!

Ich möchte auch dann ordentlich dokumentieren, selbst wenn der Druck im Projekt steigt. Daher müssen wir einen Prozess für die Dokumentation implementieren, der sich stark an Vorlagen und Checklisten orientiert.

### Delta vs. full documentation

Implementierung eines Prozesses, der sicherstellt, dass die Gesamtdokumentation nach Auslieferung eines Delta-Releases auch aktualisiert wird.

## Zielgruppen für Dokumentation

Für wen schreiben wir eigentlich Dokumentation? Es gibt unterschiedliche Zielgruppen in einem Software-Projekt mit jeweils unterschiedlichen Bedürfnissen:

- Customer / Business User
- Developer
- Test-Team
- IT-Operations
- 3rd Level Support
- End-User of the application

Wir haben festgestellt, dass die wesentliche Information für alle beteiligten Zielgruppen identisch ist. Es geht immer um die Business Rules der Applikation, welche fachliche Anforderungen wurden auf Seite x in dieser APEX Applikation umgesetzt? Was genau soll passieren, wenn dieser Button gedrückt wird? Wer darf überhaupt diesen Button drücken?

Dabei ist das **WAS** weitaus wichtig als das **WIE**. Die genaue Implementierung kann ich mir immer erschließen, wenn ich mir den Source Code ansehe. Gerade für das Testen muss ich wissen, was das erwartete Ergebnis einer Benutzer-Interaktion sein soll.

## Vorstellung des Konzeptes anhand der APEX Beispiel-Applikation

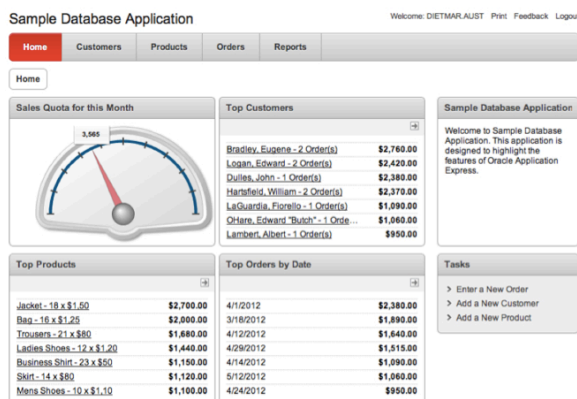


Abbildung 1: APEX Beispiel-Applikation

In jedem APEX Workspace steht standardmäßig die APEX Beispiel-Applikation zur Verfügung. Anhand dieser sollen die Konzepte vorgestellt und diskutiert werden (siehe Abbildung 1).

Im Rahmen des Vortrages werden wir die aktuelle Version dokumentieren sowie die Dokumente besprechen, um eine Änderung am System vorzunehmen.

## Aufbau der Dokumente

Wir verwenden üblicherweise folgende Dokumente mit den unterschiedlichen Zielsetzungen (Abbildung 2):



Abbildung 2: Übersicht der Dokumente

Um den Gesamtprozess zu vereinfachen (und die Dokumentation mit minimalem Aufwand zu realisieren) sorgen wir dafür, dass das System Design Dokument und die Physische Implementierung (APEX Applikationen, das physische Datenmodell, etc.) so ähnlich wie möglich sind.

Dadurch werden wir in die Lage versetzt, nach der ersten Implementierung den aktuellen Stand des Design Dokuments wieder aus der physischen Implementierung abzuleiten über ein Verfahren des Reverse Engineering.

Dabei werden die aktuellen Strukturen (APEX Seiten, Prozesse, Validierungen, Berichte, etc.) sowie die hinterlegten Kommentare aus dem APEX Data Dictionary ausgelesen und zur Dokumentation verwendet. Dies gilt analog für das Datenmodell (Tabellen, Views) wie auch die Datenbank-Packages.

## WordWiki

Um das Reverse Engineering zu automatisieren und die tatsächliche Erstellung der Dokumentation zu vereinfachen, haben wir die sogenannte WordWiki Funktionalität entwickelt.

Dabei gehen wir hin und betten in ein Word Dokument mit einer speziellen (an Wikis angelehnten) Syntax Anweisungen ein, um diese dann letztlich aus der Entwicklungsdatenbank live zu extrahieren.

Dies geschieht über einen Makro-gesteuerten Rendering Prozess, der das Template kopiert, das Dokument linear verarbeitet und für jeden Wiki-Eintrag einen http Aufruf startet, um die Metadaten des benannten Objektes (APEX Seite, Datenbank-Tabelle bzw. Datenbank-Package) zu extrahieren.

Auf der anderen Seite haben wir eine APEX Applikation erstellt, die die jeweils gewünschten Metadaten für ein Objekt ermittelt und im HTML Format zurückliefert.

Über diesen Ansatz sind wir in der Lage, von einem initial statischen MS Word Dokument (bevor der Source Code vorhanden ist, wird ein technisches Design erstellt) sanft zu einem weitestgehend aus Kommentaren im Programmcode generierten Dokument zu gelangen.

Weiterhin können wir damit ebenfalls statischen Inhalt (Diagramme, eingebettete MS Office Dokumente, etc.) mit dynamischen Inhalt (Tabellen, Spalten, Kommentare) kombinieren.

## Fazit

In diesem Vortrag stellen wir unsere gewonnenen Erfahrungen in der Dokumentation von APEX Projekten dar. Wir arbeiten mit Dokumentenvorlagen und Checklisten, um die Durchführung so weit es geht, zu vereinfachen.

Nach der initialen Implementierung (Release 1.0) wird die Dokumentation immer stärker in den Source Code verlagert.

Um dabei trotzdem am Ende eine Dokumentation im MS Word Format zu erhalten, haben wir die WordWiki Funktionalität entwickelt. Damit können wir makrogesteuert Metadaten der Applikationsbestandteile (APEX Seiten, Tabellen, Views, Packages, etc.) aus dem Sourcecode extrahieren und in das MS Word Dokument integrieren.

Kontaktadresse:

Dietmar Aust  
Zum Tilmeshof 11  
D-50859 Köln

Telefon: +49(0)173-5322 955  
Fax: +49(0)221-17099759  
E-Mail: [dietmar.aust@opal-consulting.de](mailto:dietmar.aust@opal-consulting.de)  
Internet: <http://www.opal-consulting.de>  
BLOG: <http://daust.blogspot.com>  
APEX-Tools: <http://www.opal-consulting.de/tools> (zur freien Nutzung und kostenlos)  
APEX-Buch: <http://apex-xe-praxis.de/>