

HOCHWERTIGE ENTWICKLUNGSDOKUMENTATION AUTOMATISCH MIT ASCIIDOC

| von MARTIN SERNOW

Je größer ein Softwareprojekt, desto höher die Wahrscheinlichkeit, dass das Anpassen der Entwickler-, System- und Releasedokumentation von den Bearbeitern vergessen wird. Sei es, weil zu viel Dokumentation existiert oder weil sich das Projekt in einer kritischen Phase

befindet und die „goldenen Henkel“ ganz sicher später nachgeholt werden – so jedenfalls der gute Vorsatz.

Die Lösung für dieses Problem wäre, den Medienwechsel zwischen IntelliJ+Git und Word+SVN möglichst gering zu halten

und die Dokumentation dort abzulegen, wo sich die dokumentierte Problematik befindet. Hier kann die Auszeichnungssprache AsciiDoc Abhilfe schaffen. Wir verwenden es, um den Releaseletter, die Installationsanleitung, das Systemhandbuch und den Systementwurf durch AsciiDoc automatisiert generieren zu können. Durch die Umstellung von einem Worddokument auf mehrere AsciiDoc-Dokumente ergeben sich verschiedene Vorteile, die nachfolgend erläutert werden.

WAS IST ASCIIDOC?

AsciiDoc ist eine Auszeichnungssprache, mit der man schnell und einfach formatierte Dokumente erstellen kann. AsciiDoc kann mithilfe des Tools AsciiDoctor¹ direkt in den Softwareerstellungs- und Build-Prozess integriert werden. Aus der Code-Dokumentation werden Lieferartefakte erzeugt, die mit spezifischen Formatvorlagen für unterschiedlichste Ausgabeformate hinterlegt werden können. Dadurch entfällt die neben oder nach der Entwicklungs- und Releasephase anfallende zusätzliche Dokumentationsphase. Die erzeugte Dokumentation kann durch viele Werkzeuge (zum Beispiel IntelliJ, Eclipse, Visual Studio) geöffnet, erweitert und angesehen werden, ohne dass weitere Anwendungen gestartet werden müssen.



WELCHE PROBLEME BEI DER SYSTEM- UND RELEASEDOKUMENTATION WERDEN DURCH ASCIIDOC GELÖST?

Softwareentwickler nutzen für ihre Arbeit eine Entwicklungsumgebung (zum Beispiel IntelliJ) sowie gegebenenfalls weitere Entwicklungswerkzeuge, wie



Eine Auszeichnungssprache ist eine maschinenlesbare Sprache für die Gliederung und Formatierung von Texten. So kann man mit einfachen Mitteln wie ***Fett*** schreiben, mit **TIP:** ein Icon einfügen und per `image:icons/avatar.png[title="Avatar"]` sogar ein Bild ins Dokument einfügen. Ziel ist es, mit wenig Aufwand ein einheitliches und gut formatiertes Dokument zu schreiben.²

zum Beispiel Konsole, Datenbankviewer und Browser und verwalten ihre Arbeit in einer modernen Versionsverwaltung (zum Beispiel Git). Die Dokumentation dagegen findet über MS Word und die Verwaltung über eine „klassische“ Versionsverwaltung statt (zum Beispiel svn). Der an dieser Stelle entstehende Medienbruch und zusätzliche Aufwand führen häufig dazu, dass die Anpassung oder Aktualisierung der Dokumentation verschoben oder sogar ganz vergessen wird. Auch die Suche nach Informationen in der Dokumentation erweist sich oft als schwierig, da viele Entwicklungsumgebungen nicht sehr leistungsfähig in der separat abgelegten Dokumentation suchen können.

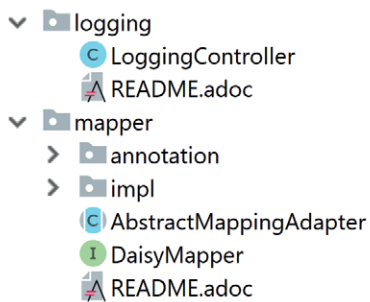


Abbildung 1: Projektstruktur mit integrierter ADOC-Datei

Dagegen liegen bei der Verwendung von AsciiDoc die Dateien für die Dokumentation direkt beim Code, im selben Verzeichnis wie die implementierte Klasse.

Andere Entwickler, die sich genau mit diesem Code befassen, können problemlos das entsprechende AsciiDoc-Dokument aufrufen, das entsprechende Informationen für die Arbeit an der Klasse bereitstellt. Zudem können Änderungen aufgrund von Change Requests oder Bugs sofort und an Ort und Stelle vorgenommen werden.

Code und Dokumentation können in einem gemeinsamen Review-Prozess der Entwicklung geprüft werden. Den Reviewern stehen beide Prüfobjekte gemeinsam zur Verfügung, und sie erkennen am Fehlen des AsciiDoc-Dokuments, dass eine Dokumentation noch aussteht.

Ein weiteres Problem ist das einheitliche Layout von MS-Word-Dokumenten. Da es sehr viele Möglichkeiten und Optionen gibt, ein Dokument „wunschgemäß“ zu formatieren, steigt die Vielfalt der „Layouts“ und Formatierungen, je länger ein Dokument bearbeitet wird und je mehr Personen daran arbeiten. Die Entwickler verlieren sich immer mehr darin, Design über Inhalte zu stellen.

AsciiDoc bietet zwar ebenfalls eine gute Auswahl an Dokumentdesign- und Formatierungsmöglichkeiten, allerdings wird das Design erst bei der Erzeugung des Artefakts (wie PDF, HTML, DOCX, PPT) auf Basis von Formatvorlagen (Templates) von AsciiDoctor über den Inhalt „gestülpt“. So müssen und können sich Entwickler vor allem den Inhalt konzent-

rieren. In vielen Fällen schreibt AsciiDoc sogar selbst vor, wie bestimmte Informationen formatiert sein müssen (zum Beispiel für Überschriften, Code Snippets, Bilder oder Tabellen).

Grafiken sind bei der System- und Entwicklerdokumentation oft ein wichtiges Mittel, um die Technik dahinter verständlich zu machen. Man möchte die Medien so ablegen, dass sie im generierten Dokument richtig angezeigt werden. Bei MS Word werden die Medien einfach per Drag & Drop in das Dokument „gezogen“. Allerdings ist danach das Aktualisieren der Bilder schwierig. Dafür bietet Word nur an, die Medien über eine Referenz zu ihrem Ablageort zu speichern. AsciiDoc ermöglicht das analog über die Referenzierung via URL-Notation des Ablagepfads. Dabei reicht auch eine relative Referenz mit Bezug zu einem zentralen Webserver aus, so dass man die Medien zentral bearbeiten, zum Download anbieten und Duplikate vermeiden kann.

Das Ausgabeformat von AsciiDoctor ist standardmäßig auf PDF und HTML5 beschränkt. Das Tool Pandoc³, das oft für Medienkonvertierungen über die Kommandozeile verwendet wird, bietet weitere Ausgabeformate an. Zusätzlich kann man Dokumente in mehreren Formaten exportieren: So kann man für Projektmitarbeiter die Dokumentation auf einem Siteserver hosten, den Beteiligten ein PDF schicken und gegebenenfalls für die nachträgliche Bearbeitung noch ein Docx bereitstellen. Um jedoch Kommentare von Nutzern zurück ins System zu spielen und dort weiterzuverarbeiten, wäre ein noch zu erstellendes Tool notwendig, das diese Kommentare aus dem dem Dokument extrahiert und zum Beispiel als To-dos an das entsprechende AsciiDoc-Dokument anhängt.

WIE INTEGRIERT MAN DEN BUILD-PROZESS IN SEIN PROJEKT?

Ein einfaches Beispiel zeigt, wie man eine Datei in einem Build Tool wie Maven integrieren kann, damit beim Build-Process das entsprechende Dokument mitgeneriert und an die Beteiligten ausgeliefert werden kann:

- Im ersten Schritt bindet man das asciidoctor-maven-Plugin ein, am besten im Build Process mit dem Profil asciidoc, damit man über dieses Profil steuern kann, wann man die Dokumente generiert.
- Als Nächstes legt man die verschiedenen Konvertierungen (executions) fest – in diesem Fall für den Systementwurf

```

<plugin>
  <groupId>org.asciidoctor</groupId>
  <artifactId>asciidoctor-maven-plugin</artifactId>
  <version>1.6.0</version>
  <executions>
    <execution>
      <id>generate-systementwurf-as-docbook</id>
      <phase>generate-resources</phase>
      <goals>
        <goal>process-asciidoc</goal>
      </goals>
      <configuration>
        <backend>docbook</backend>
        <doctype>book</doctype>
      </configuration>
    </execution>
  </executions>
</plugin>
<configuration>
  <sourceDirectory>.</sourceDirectory>
  <sourceDocumentName>SYSTEMENTWURF.adoc</sourceDocumentName>
  <outputDirectory>target/generated-docs</outputDirectory>
  <headerFooter>true</headerFooter>
  <logHandler>
    <outputToConsole>>false</outputToConsole>
    <failIf>
      <severity>DEBUG</severity>
    </failIf>
  </logHandler>
</configuration>

```

Abbildung 2: Projekt Object Model um ADOC Plugin erweitert

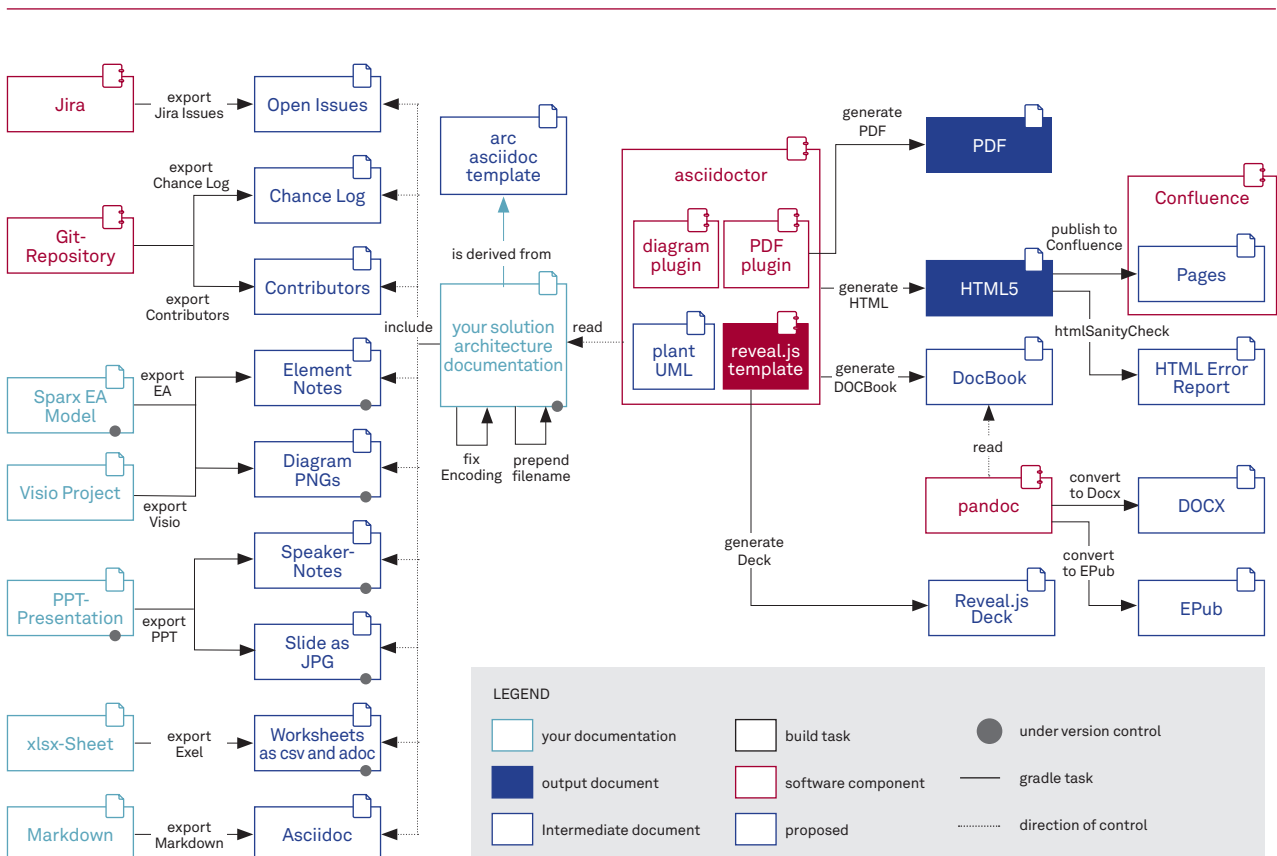


Abbildung 3: Übersicht der Möglichkeiten von DocToolchain®

DocBook (dies kann später zu dem Word-Dokumententyp docx konvertiert werden). In der Konfiguration sagt man mit dem Tag „backend“, dass man ein docbook wünscht. Für das DocBook-Dokument kann man html oder auch pdf verwenden, natürlich immer mit dem entsprechenden Doctype.

- Beliebig weitere executions kann man erzeugen, um entweder den Systementwurf in weitere Formate zu konvertieren oder auch andere Dokumente zu erzeugen.
- In der Konfiguration werden allgemeine, für alle executions gültige Werte übergeben, wie beispielsweise das Ausgabeverzeichnis (hier outputDirectory), welche Logeinträge geschrieben werden und woher die Quelldateien kommen.

DOCTOOLCHAIN

Doch was, wenn man mit mehreren Tools wie Sparx EA, Git, Jira oder Visio arbeitet und daraus die Inhalte in seinem AsciiDoc -Dokument anzeigen und aktuell halten möchte?

Das Werkzeug DocToolchain⁴ ermöglicht es, dynamisch noch mehr entwicklungsrelevante Informationen automatisiert in die AsciiDoc-Dokumente einzufügen. Ursprung der mittlerweile zahlreichen, automatisch Dokumentationsfunktionen unterstützenden DocToolchain war der Wunsch, Diagramme aus dem Enterprise Architekt⁵ in AsciiDoc-Dokumente einzubinden (siehe Abbildung 3).

Beispielszenario

Ein gutes Beispiel für „Documentation am Code“ aus der Projektpraxis ist der Einsatz von AsciiDoc im Ticketprozess. Änderungen im Programmcode aufgrund eines Tickets werden über ein spezielles Commit in Git abgebildet. Abschließende Commits wurden um das Kennzeichen „#releaseletter“ erweitert. DocToolchain sammelt automatisch für den Releaseletter alle Commits aus der Git-Historie und schreibt sie in den Releaseletter. Somit werden wichtige-Feature-Entwicklungen oder Fehlerkorrekturen im Releaseletter verzeichnet, ohne dass ein Release-Verantwortlicher die entsprechenden Tickets suchen und manuell hinzufügen muss.

ZUSAMMENFASSUNG

Mit dem Wechsel von MS Word auf eine Dokumentation direkt am Code durch AsciiDoc werden den Entwicklern manuelle Arbeiten abgenommen. Dies mindert das Fehlerpotenzial und versetzt Entwickler in die Lage, Code schneller zu verstehen und, neben JavaDoc⁷, besser zu dokumentieren. Der Release-Prozess wird durch die automatische Zusammenstellung von Release-Dokumentationen schlanker, die manuelle Arbeit entfällt zum großen Teil. Alles in allem „lebt“ die Entwicklungsdokumentation und wird stetig aktualisiert und publiziert. ●

1 <https://asciidoc.org/> (abgerufen am 26.10.2020).

2 Eine genaue Auflistung der AsciiDoc Formatierungen siehe hier: <https://asciidoc.org/docs/asciidoc-syntax-quick-reference/> (abgerufen am 30.03.2021).

3 <https://pandoc.org/> (abgerufen am 30.03.2021).

4 <https://doctoolchain.github.io/docToolchain/> (abgerufen am 26.10.2020).

5 <https://www.sparxsystems.de/> (abgerufen am 26.10.2020).

6 https://doctoolchain.github.io/docToolchain/#_overview_of_available_tasks (abgerufen am 26.10.2020).

7 <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html> (abgerufen am 26.10.2020).