



Die automatisierte Erstellung und Pflege von Code-Dokumentationen stellt oft eine Herausforderung dar. Für BELIMO haben wir eine Lösung entwickelt, die sich nahtlos in deren CI/CD-Pipeline integrieren lässt und die automatisierte Dokumentation von .NET- und SPS-Code ermöglicht.

**KUNDE**

BELIMO Automation AG
Brunnenbachstrasse 1
8340 Hinwil
www.belimo.com

PROJEKT

Die BELIMO Automation AG mit Hauptsitz im schweizerischen Hinwil wurde 1975 gegründet und zählt heute zu den weltweit führenden Anbietern im Bereich der Antriebslösungen für Heizungs-, Lüftungs- und Klimaanlage. Das Unternehmen entwickelt, produziert und vertreibt innovative Produkte für die Steuerung von Energieflüssen in Gebäuden.

BELIMO bietet massgeschneiderte Lösungen, die Energieeffizienz und Raumkomfort verbessern. Als global tätiges Unternehmen legt BELIMO grossen Wert auf Qualität, Innovation und Kundennähe. Die Fertigung wesentlicher Komponenten erfolgt im eigenen Haus, was höchste Standards in Technik und Zuverlässigkeit garantiert.

Folgende Dienstleistung durfte AVM Engineering AG für die BELIMO Automation AG umsetzen:

➔ Automatisierte Generierung von Source-Code-Dokumentation mit Hilfe von DocFx

Generierung der Dokumentation

Das Grundbedürfnis ist, ein auf .NET basiertes Softwareprojekt zu dokumentieren. Das heisst, es soll eine einheitliche, aktuelle und korrekte Dokumentation erstellt werden, die Einsicht über die Funktionalitäten des Projekts gibt. Dazu soll der gesamte Prozess, die Dokumentation zu generieren, automatisiert ablaufen.

Als Grundlage für die Lösung dient DocFx – ein Open-Source-Projekt aus dem Hause Microsoft. Es hat die Fähigkeit, aus .NET-Code direkt eine strukturierte und einheitliche Dokumentation zu generieren. Darin werden Klassen und die darin enthaltenen Elemente übersichtlich festgehalten.

Der Entwickler kann mit sogenannten "XML-Kommentaren" die Elemente genauer spezifizieren:

```
/// <summary>
///     This method is used to calculate the sum of two numbers.
/// </summary>
/// <param name="a">
///     The first number.
/// </param>
/// <param name="b">
///     The second number.
/// </param>
/// <returns>
///     The sum of the two numbers.
/// </returns>
/// <remarks>
///     Please note that it can not multiply or divide the numbers.
/// </remarks>
```

Abb.: XML-Kommentare

Diese Kommentare werden dann ebenfalls in die Dokumentation integriert, um dem Leser einen besseren Einblick über die Funktionsweise und die korrekte Verwendung der Komponenten zu geben. So ist der Nutzer nicht mehr dazu gezwungen, den Code in einer IDE zu öffnen und im Detail anzusehen. Die Abhängigkeiten und Anwendung einer API können also nachvollziehbarer und zugänglicher dargestellt werden.

Im Detail läuft dieser Generierungsvorgang folgendermassen ab:

1. Das .NET-Projekt wird gebaut
2. DocFx generiert anhand des Projekts eine Reihe an YAML-Dateien, die alle notwendigen Informationen über die Klassen, Methoden, Parameter etc. enthalten
3. Die YAML-Dateien werden anschliessend in HTML-Dateien konvertiert, welche in diesem Schritt untereinander verknüpft werden
4. Diese HTML-Dateien können dann im Browser angesehen werden. Anhand der Verknüpfungen können dann zwischen den Komponenten navigiert werden.

Strukturierter Text

BELIMO hat zusätzlich noch weitere Softwareprojekte, die sie ebenfalls dokumentieren möchten. Diese sind jedoch nicht .NET-basiert, sondern beinhalten SPS-Code beziehungsweise strukturierten Text. Da DocFx mit dieser Art von Input nicht umgehen kann, wurde ein zusätzlicher Parser implementiert. Dieser übernimmt den initialen Konvertierungsvorgang von DocFx und übersetzt den Input-Code in die genannte YAML-Struktur, die identisch ist mit dieser, die automatisch aus .NET-Projekten generiert wird. Diese Dateien werden dann als Input an DocFx gegeben, woraus so die Dokumentation erstellt wird.

Versionierung

Dokumentationen von vergangenen Projektversionen müssen festgehalten werden. Dazu wurde ein Versionierungskonzept umgesetzt, bei der jede neue Version der Dokumentation, die durch eine neue Version der dazugehörigen Software entsteht, persistiert wird. Dadurch ist sichergestellt, dass auch ältere Versionen der Dokumentation jederzeit nachvollziehbar und zugänglich bleiben.

CI / CD

Um sicherzustellen, dass der Betriebsaufwand so gering wie möglich gehalten werden kann, ist der Dokumentationsprozess vollständig automatisiert und ist nahtlos in die bestehende Release-Pipeline eingebunden.

Der Output, also die HTML-Struktur, welche DocFx generiert, wird in ein Artifactory geladen. Dort werden die Dateien dann für den Webserver bereitgestellt.

Was ist in Zukunft möglich?

Das umgesetzte Projekt sollte auch dazu dienen, aufzuzeigen, welche Möglichkeiten bestehen, um Prozesse zu automatisieren.

Es öffnet die Tür für weitere Projekte und Erweiterungen wie beispielsweise, das Integrieren von Klassendiagrammen. DocFx ist in der Lage, Diagramme, unter anderem PlantUML, in der Website gerendert einzubetten.

So könnte zum Beispiel, zusätzlich zum Code, auch die Code-Struktur dokumentiert werden.

Aus den Klassen könnte ein Klassendiagramm in Form eines PlantUML-Diagramms erstellt werden, welches dann direkt in die Dokumentation miteingebaut werden kann. Und das auch wieder vollkommen automatisiert.

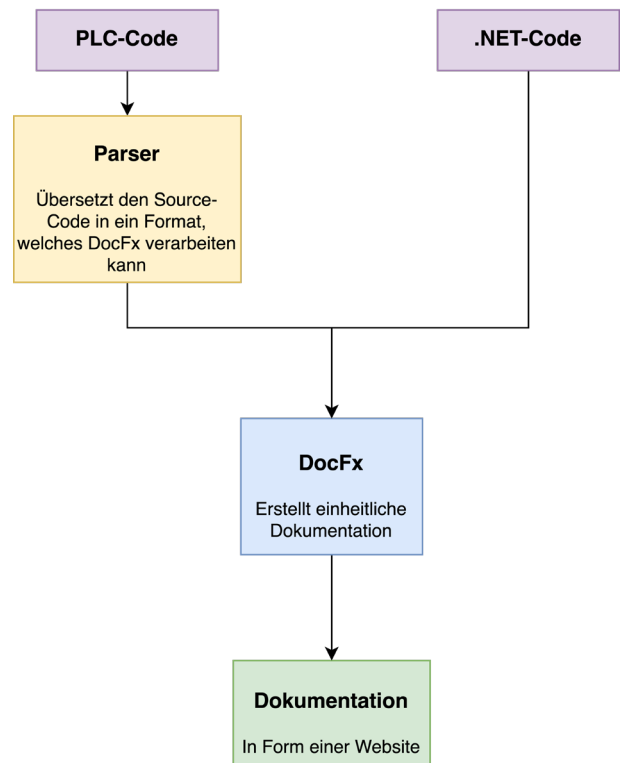
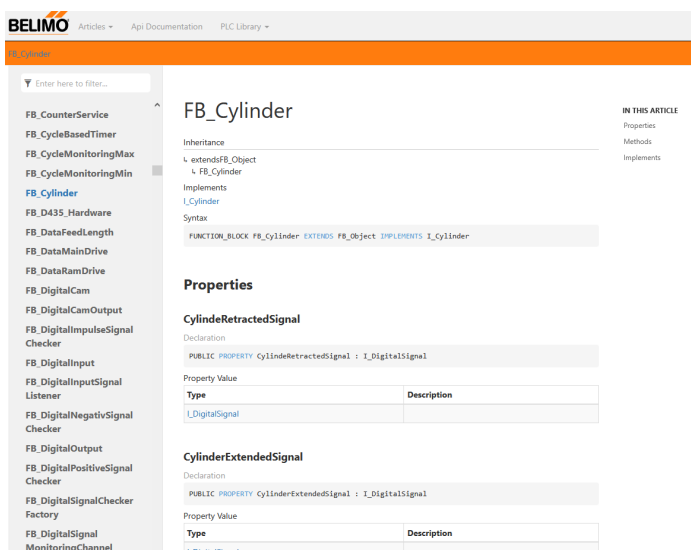
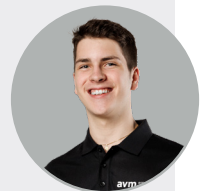


Abb.: Generierungsprozess



FAZIT

Die Automatisierung eines zuvor manuellen Prozesses ist eine äusserst lohnende Aufgabe. Es ist faszinierend zu beobachten, wie die einzelnen Komponenten harmonisch zusammenwirken, um ein gemeinsames Ziel zu erreichen, bis schliesslich am Ende der Pipeline ein fertiges Produkt entsteht.



AVM Engineering AG
Mario Soller

mario.soller@avm.swiss
+41 71 544 60 50

Abb.: Automatisch generierte Dokumentation eines Funktionsbausteins mithilfe von DocFx