



Kontinuierliche Architekturdokumentation im agilen Umfeld



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Ihr Sprecher

Falk Sippach (@sipsack)

Trainer, Berater, Entwickler



Co-Organisator

*Architektur
Agile Softwareentwicklung
Codequalität*



Feedback gern an: @sipsack, falk.sippach@oio.de

Java und XML

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen



Man kann zwar an vielen Stellen nachlesen, wie man Architekturdokumentation strukturiert. Aber auf der Suche nach einer praktikablen Handhabung zur Erstellung und Pflege enden die meisten Versuche in der WYSIWYG-Hölle einer Textverarbeitung oder im tiefen Schlund eines Wikis. In diesem Vortrag wollen wir uns anschauen, wie aufbauend auf bestehenden Tools und Textformaten eine möglichst redundanzfreie Dokumentation erstellt und für verschiedene Zielgruppen in ansprechenden Formaten ausgeliefert werden kann. Es wird dabei um Begriffe wie Continuous Documentation und Documentation as Code gehen.

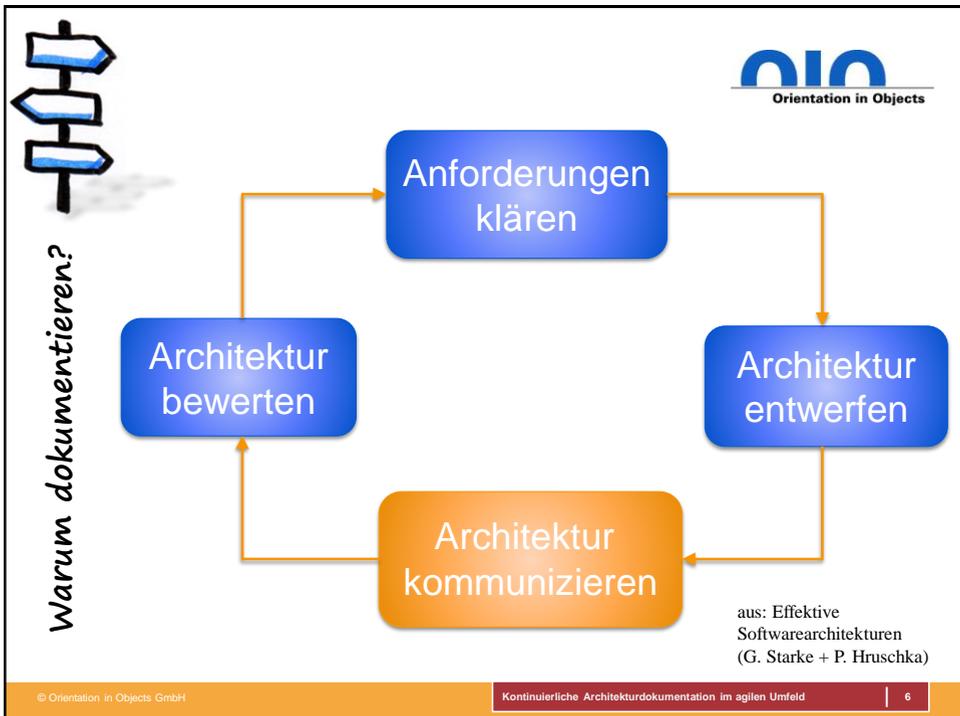
Doku-Smells

Die 7 Sünden der Architekturdokumentation
aus Entwicklersicht

Foto von jesperbkskov: <https://pixabay.com/en/fish-newspaper-food-russian-salted-224097/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 5



Gründe für eine Architektur-Dokumentation

Where's the fun in just knowing what the code is supposed to do?



Essential

Excuses for Not
Writing Documentation

Entwurfsunterstützung

Kommunikation

Bewertbarkeit

Frage nach Warum

Neue Mitarbeiter

ORLY?

@ThePracticalDev

Grafik von The Practical Dev: <https://github.com/thepracticaldev/orly-full-res/blob/master/notwritingdocs-big.png> (CC0 Public Domain Lizenz)



Doku-Smells

WYSIWYG

ABLAGE

REDUNDANZEN

TEXTWÜSTE

HANDARBEIT

TOTE DOKU

ELFENBEINTURM



Doku-Smells

WYSIWYG

ABLAGE

REDUNDANZEN

TEXTWÜSTE

HANDARBEIT

TOTE DOKU

ELFENBEINTURM

Hauptsache,
du machst es
nicht mit Word!

**Ich mach's
mit Latex.**

Kondome schützen.
Für eine Nacht oder mehr, mach's mit.

GIB AIDS
KEINE
CHANCE

STI
Sexuell
übertragbare
Infektionen

oio
Orientation in Objects

Textverarbeitung

Visio

Powerpoint

UML-Tools

Foto von Antrianas: <https://pixabay.com/en/telescope-view-distant-binoculars-1201497/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 11

Alternative Datenformate zu Textverarbeitung

oio
Orientation in Objects

Plain-Text

Plain-Text, leicht lesbar, einfach editierbar, automatisiert verarbeitbar

Wikis: kollaborativ

LaTeX/DocBook: eingeschränkte Lesbarkeit

Mindmaps: visuell, kurz & knapp

Richtext: Austauschformat

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 12

Markdown

Normaler Text wird so dargestellt wie eingegeben.

```
*Kursiv*, **Fett** und ***Fett kursiv***  
und ___Fett kursiv___
```

Markiert Text als ``Inline-Quelltext``

```
Ein Code-Block  
durch Einrückung  
mit vier Leerzeichen
```

```
* Ein Punkt in einer ungeordneten Liste  
* Ein Unterpunkt, um vier Leerzeichen
```

```
1. Ein Punkt in einer geordneten Liste  
2. Ein weiterer Punkt  
1. Noch ein Punkt bei mehrfacher Angabe
```

```
# Überschrift in Ebene 1  
#### Überschrift in Ebene 4
```

Normaler Text wird so dargestellt wie eingegeben.

Kursiv, **Fett** und **Fett kursiv** bzw. *Kursiv*, **Fett** und **Fett kursiv**

Markiert Text als `Inline-Quelltext`

```
Ein Code-Block  
durch Einrückung  
mit vier Leerzeichen
```

- Ein Punkt in einer ungeordneten Liste
 - Ein Unterpunkt, um vier Leerzeichen eingerückt

1. Ein Punkt in einer geordneten Liste
2. Ein weiterer Punkt
3. Noch ein Punkt bei mehrfacher Angabe derselben Ziffer

Überschrift in Ebene 1

Überschrift in Ebene 4

AsciiDoc

```
= AsciiDoc is Writing Zen  
Doc Writer <doc.writer@example.com>  
:icons: font
```

```
_Zen_ in the art of writing `plain text` with  
http://asciidoc.org[AsciiDoc].
```

```
[TIP]  
Use http://asciidoctor.org[Asciidoctor]  
experience.footnote:[Not to mention  
Then icon:twitter[role=aqua] about it]
```

```
== Sample Section
```

```
[square]  
* item 1  
* item 2
```

```
[source,ruby]  
----  
puts "Hello, World!"  
----
```

AsciiDoc is Writing Zen

Doc Writer – doc.writer@example.com

Zen in the *art* of writing plain text with [AsciiDoc](#).



Use [Asciidoctor](#) for the best AsciiDoc experience.¹¹ Then [♥](#) about it!

Sample Section

- item 1
- item 2

```
puts "Hello, World!"
```

¹¹ Not to mention the best looking output!

Welcher Markup-Typ bin ich?



Michael Simons
@rotnroll666



Folge ich

Think I'm more a Markdown person than AsciiDoc. The results are great, but Markdown needs less concentration to write.



Ralf D. Müller
@RalfDMueller



Folgen

@rotnroll666 but the possibilities of AsciiDoc are better! Include code directly from the repository, render plantUML, subdocuments etc...

Einheitliche Dokumentenstruktur (z. B. arc42)

1. Einführung und Ziele

Aufgabenstellung, Qualitätsziele, eine Kurzfassung der architekturelevanten Anforderungen (insb. die nichtfunktionalen), Stakeholder.

2. Randbedingungen

Welche Leitplanken schränken die Entwurfsentscheidungen ein?

3. Kontextabgrenzung

In welchem fachlichen und/oder technischen Umfeld arbeitet das System?

4. Lösungsstrategie

Wie funktioniert die Lösung? Was sind die fundamentalen Lösungsansätze?

5. Bausteinsicht

Die statische Struktur des Systems, der Aufbau aus Implementierungsteilen.

6. Laufzeitsicht

Zusammenwirken der Bausteine zur Laufzeit, gezeigt an exemplarischen Abläufen ("Szenarien")

7. Verteilungssicht

Deployment: Auf welcher Hardware werden die Bausteine betrieben?

8. Querschnittliche Konzepte und Muster

Wiederkehrende Muster und Strukturen. Fachliche Strukturen.

Querschnittliche, übergreifende Konzepte, Nutzungs- oder Einsatzanleitungen für Technologien. Oftmals projekt-/systemübergreifend verwendbar!

9. Entwurfsentscheidungen

Zentrale, prägende und wichtige Entscheidungen.

10. Qualitätsszenarien

Qualitätsbaum sowie dessen Konkretisierung durch Szenarien

11. Risiken

12. Glossar

Wichtige Begriffe.

Legende:

anforderungsbezogene Informationen

Strukturen der Lösung (Sichten)

übergreifende (technische) Informationen

besonders wichtige Entscheidungen

arc42 Templates

	Online / Wiki	Version	Download (via dropbox)
	arc42 template in English	6.1	<ul style="list-style-type: none"> • docx • The template as a Confluence-space: <ul style="list-style-type: none"> • for Confluence version 5.x or 6.x • for confluence version 4.3 • Enterprise-Architect (tm) Model (EAP), slim version. • AscIIDoc version (via GitHub) <p>We created an introductory example (HtmlSanityChecker) to give an impression how arc42 could look like ...</p>
	arc42 Template auf Deutsch	6.0	<ul style="list-style-type: none"> • docx • Beispiele für Qualitätsanforderungen ("Q-Szenarien") • Das Template als Confluence-Space: <ul style="list-style-type: none"> • für Confluence Version 4.3.x • für Confluence 5.x und 6.x • Enterprise-Architect (tm) Model (EAP) (schlanke Version.) • xmind-Version (Mindmap, to be published soon)
	arc42 Template Deutsch	7.0	<p>update currently under internal review. Structure will remain unchanged, numerous corrections to help text. To be released Jan/Feb 2016</p> <ul style="list-style-type: none"> • AscIIDoc (master version) • docx • xmind (Mindmap) • Confluence
	arc42 template in Spanish work in progress, Microsoft Word version currently completed.	5.0	<ul style="list-style-type: none"> • docx-Version

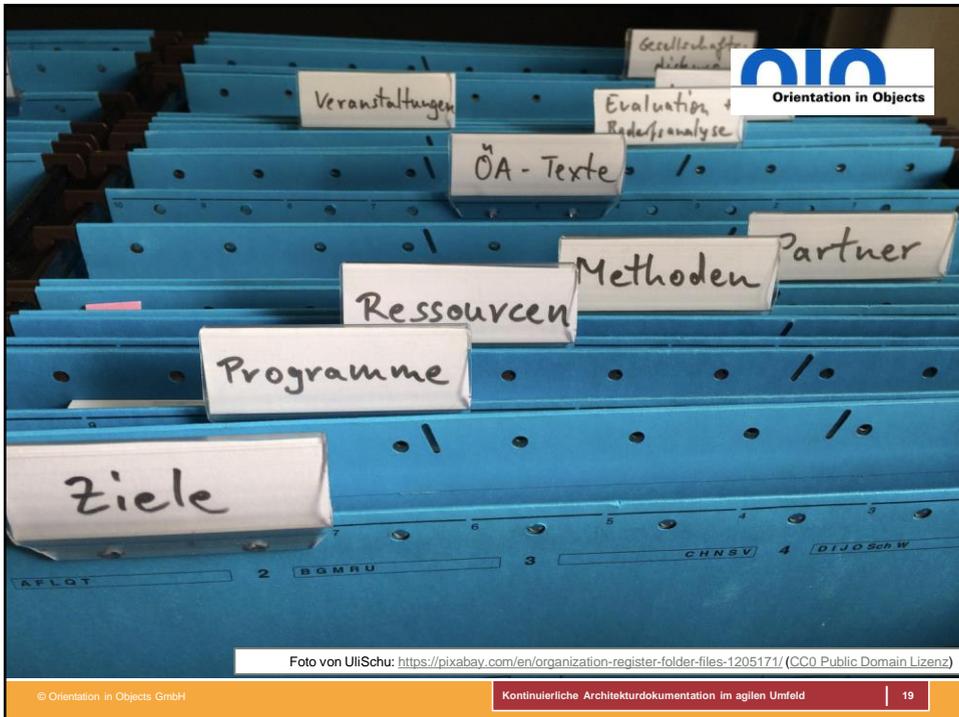


AsciiDoctor



Doku-Smells

- WYSIWYG
- ABLAGE
- REDUNDANZEN
- TEXTWÜSTE
- HANDARBEIT
- TOTE DOKU
- ELFENBEINTURM



Unser täglich Entwickler-Brot:

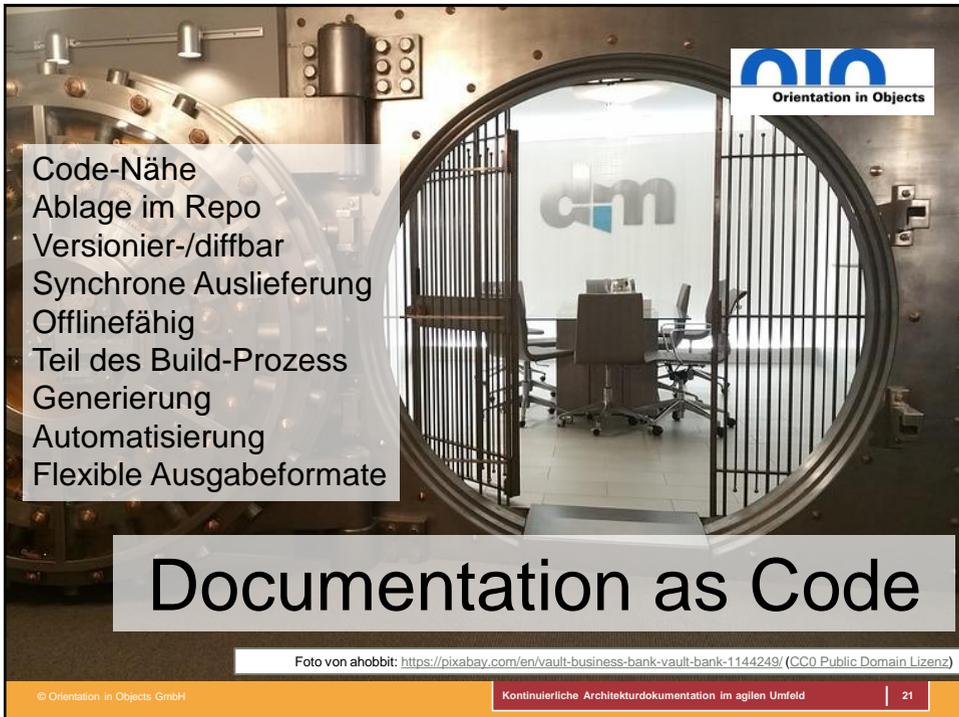
- Plain-Text
- Entwicklungs-umgebung
- Kommandozeilen-werkzeuge
- **Versions-verwaltung**

Orientation in Objects

Foto von geralt: <https://pixabay.com/de/unternehmer-start-start-up-karriere-696976/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 20



Orientation in Objects

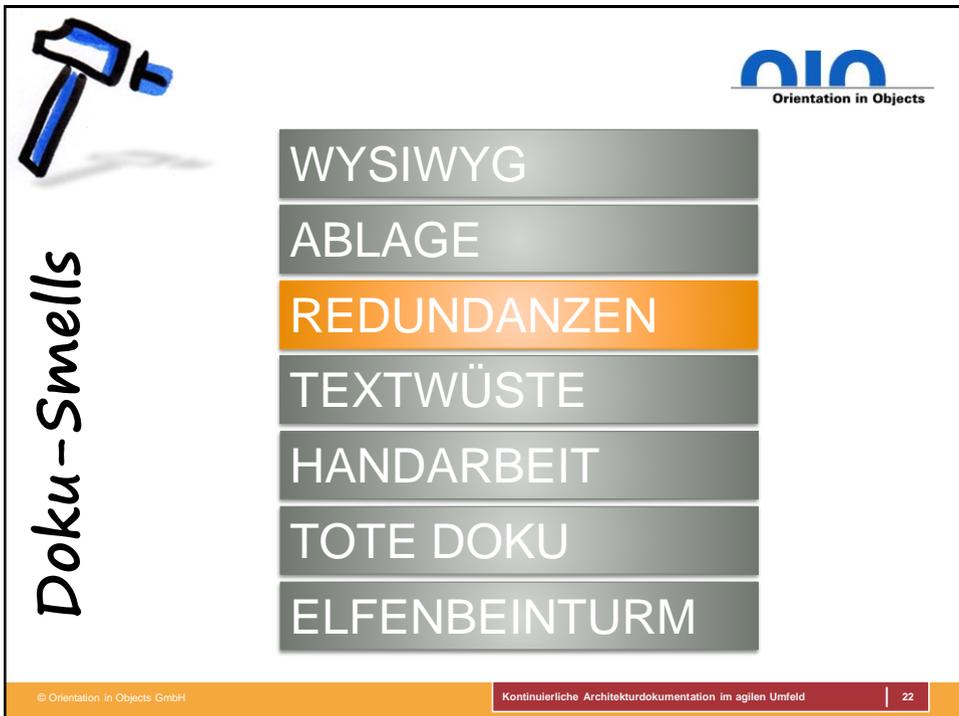
- Code-Nähe
- Ablage im Repo
- Versionier-/diffbar
- Synchrone Auslieferung
- Offlinefähig
- Teil des Build-Prozess
- Generierung
- Automatisierung
- Flexible Ausgabeformate

Documentation as Code

Foto von ahobbit: <https://pixabay.com/en/vault-business-bank-vault-bank-1144249/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 21



Orientation in Objects

Doku-Smells

- WYSIWYG
- ABLAGE
- REDUNDANZEN
- TEXTWÜSTE
- HANDARBEIT
- TOTE DOKU
- ELFENBEINTURM

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 22

A photograph of a beach with many colorful striped umbrellas (orange, green, blue, and white) set up on the sand. In the background, the blue sea is visible with several people swimming. The Orientation in Objects logo is in the top right corner.

**Single Source
of Truth**

Includes
Quellcode verlinken
Platzhalter einbetten

Foto von pcdazero: <https://pixabay.com/en/umbrellas-sea-holiday-summer-beach-921501/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 23

A photograph of several white wind turbines on a grassy hill under a clear blue sky. The Orientation in Objects logo is in the top right corner.

**Inhalte
generieren**

WSDL, Swagger
DB-Schema
Annotationen
JavaDoc
Markup generieren

Foto von ClassicallyPrinted: <https://pixabay.com/en/wind-turbine-energy-electricity-937715/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 24

Schnittstellenbeschreibung generieren



Swagger2Markup / swagger2markup Watch 15 Star 294 Fork 54

Code Issues 7 Pull requests 0 Pulse Graphs

A Swagger to AsciiDoc or Markdown converter to simplify the generation of an up-to-date RESTful API documentation by combining documentation that's been hand-written with auto-generated API documentation.

Spring REST Docs SUCCESS

Overview

The primary goal of this project is to make it easy to document RESTful services by combining content that's been hand-written using [Asciidoctor](#) with auto-generated examples produced with the [Spring MVC Test](#) framework. The result is intended to be an easy-to-read user guide, akin to [GitHub's API documentation](#) for example, rather than the fully automated, dense API documentation produced by tools like [Swagger](#).

JAX-RS Analyzer

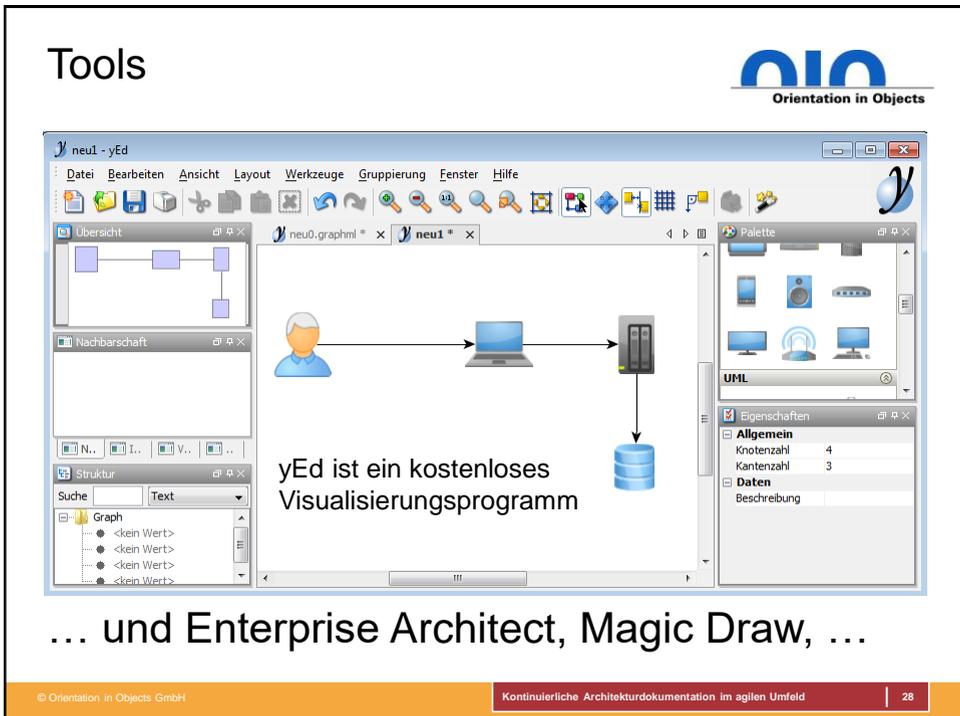
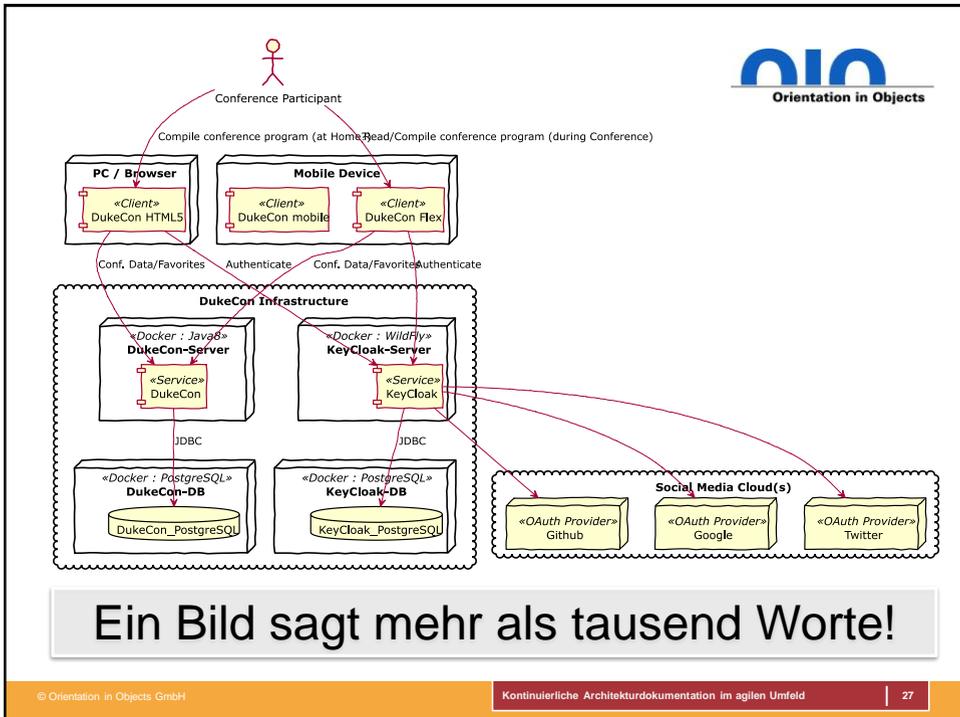
Generates an overview of all JAX-RS resources in a project by bytecode analysis.



Doku-Smells



- WYSIWYG
- ABLAGE
- REDUNDANZEN
- TEXTWÜSTE
- HANDARBEIT
- TOTE DOKU
- ELFENBEINTURM



Quelle für Bilder/Grafiken

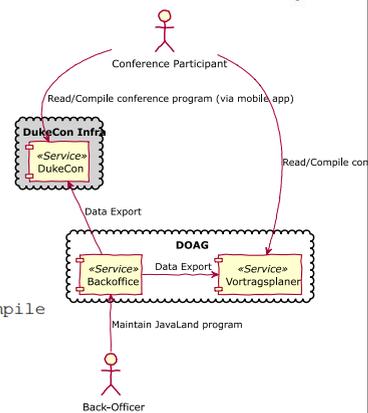
- Export aus UML-Modellen
- Manuelle Erstellung (Bildverarbeitung/Visualisierungsprogramme)

- Einbetten/Verlinken in Markup:

```
![Alternativtext] (bild.png "Bildtitel hier")  
image::bild.png[Alternativtext]
```

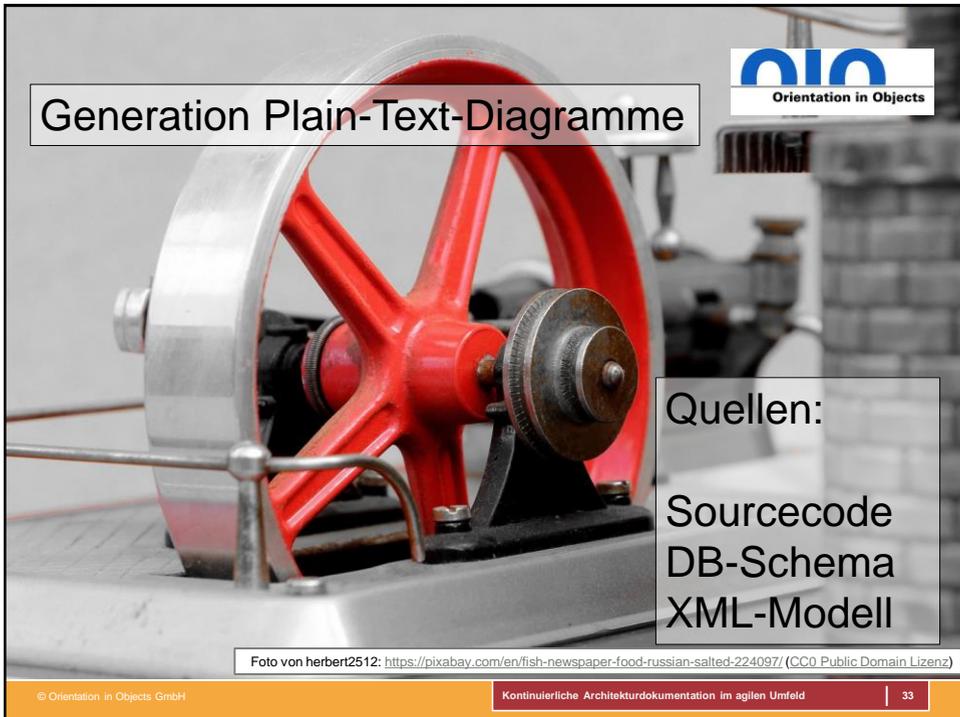
Diagramme als textuelle Beschreibung

```
@startuml  
skinparam handwritten true  
  
actor :Conference Participant:  
  
cloud "DukeCon Infra" #lightgray {  
    [DukeCon] <<Service>>  
}  
  
cloud "DOAG" {  
    [Backoffice] <<Service>>  
    [Vortragsplaner] <<Service>>  
}  
  
:Conference Participant: -right-> (DukeCon) : Read/Compile  
conference program (via mobile app)  
:Conference Participant: -right-> (Vortragsplaner) :  
Read/Compile conference program (old style)  
  
(Backoffice) -left-> (DukeCon) : Data Export  
(Backoffice) -left-> (Vortragsplaner) : Data Export  
  
actor :Back-Officer:  
:Back-Officer: -left-> (Backoffice) : Maintain JavaLand program  
@enduml
```



```
["plantuml", "dukecon-systemcontext", "svg"]
```

```
include:../systemcontext.puml []
```

Generation Plain-Text-Diagramme

Quellen:
Sourcecode
DB-Schema
XML-Modell

Foto von herbert2512: <https://pixabay.com/en/fish-newspaper-food-russian-salted-224097/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 33



Doku-Smells

WYSIWYG
ABLAGE
REDUNDANZEN
TEXTWÜSTE
HANDARBEIT
TOTE DOKU
ELFENBEINTURM

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld | 34



nio
Orientation in Objects

Agile Projekte

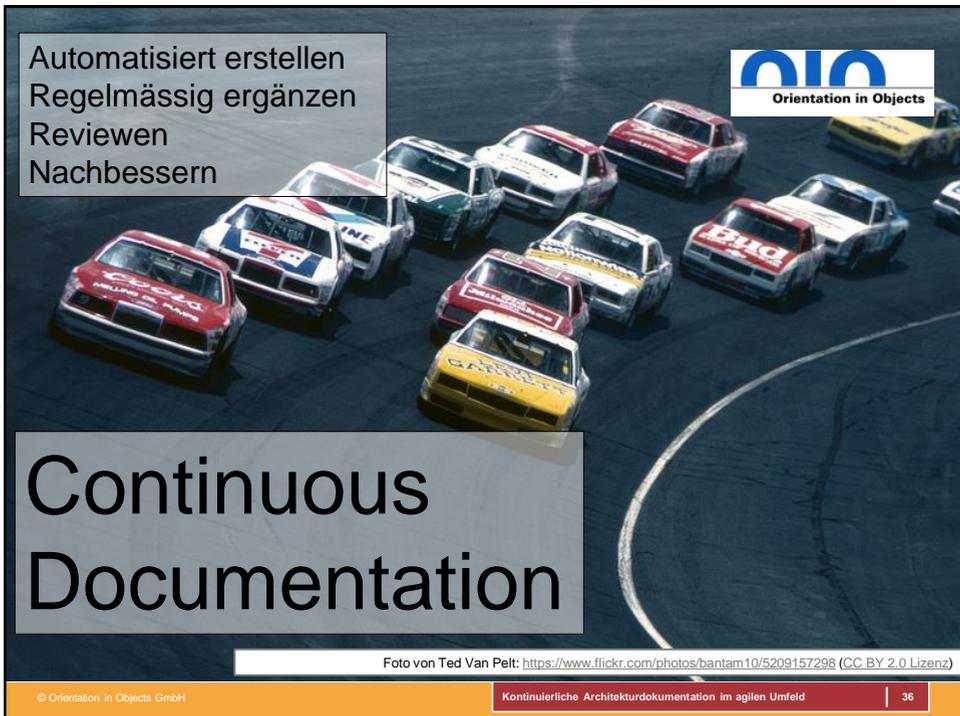
iterativ
kontinuierlich

Foto von Unsplash: <https://pixabay.com/de/fluss-bachlauf-bach-l%C3%A4ndlich-768654/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld

35



nio
Orientation in Objects

Automatisiert erstellen
Regelmässig ergänzen
Reviews
Nachbessern

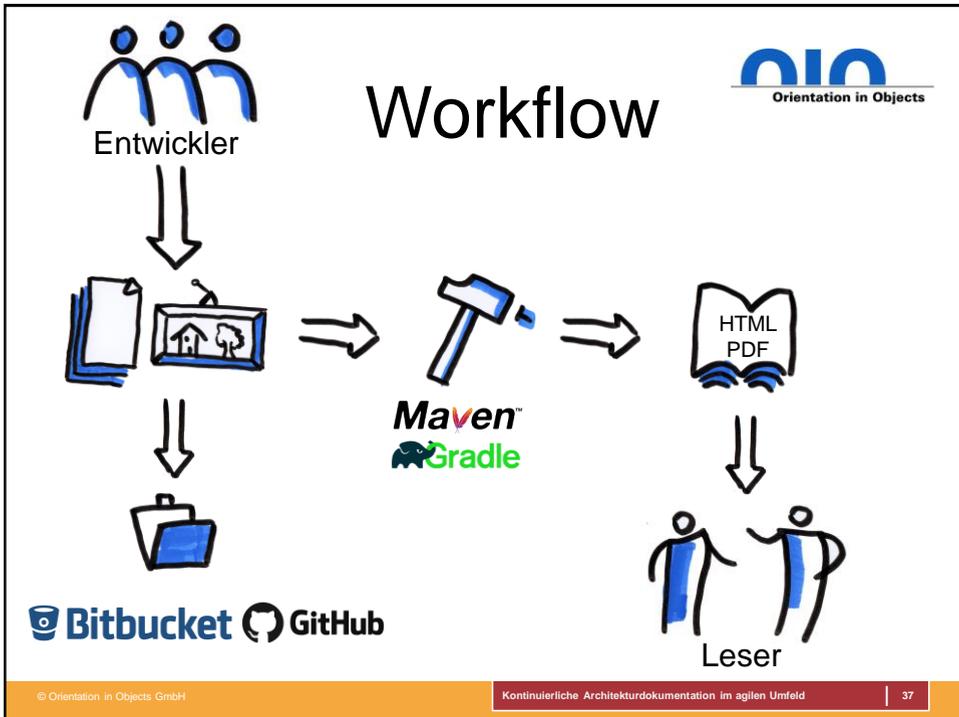
Continuous Documentation

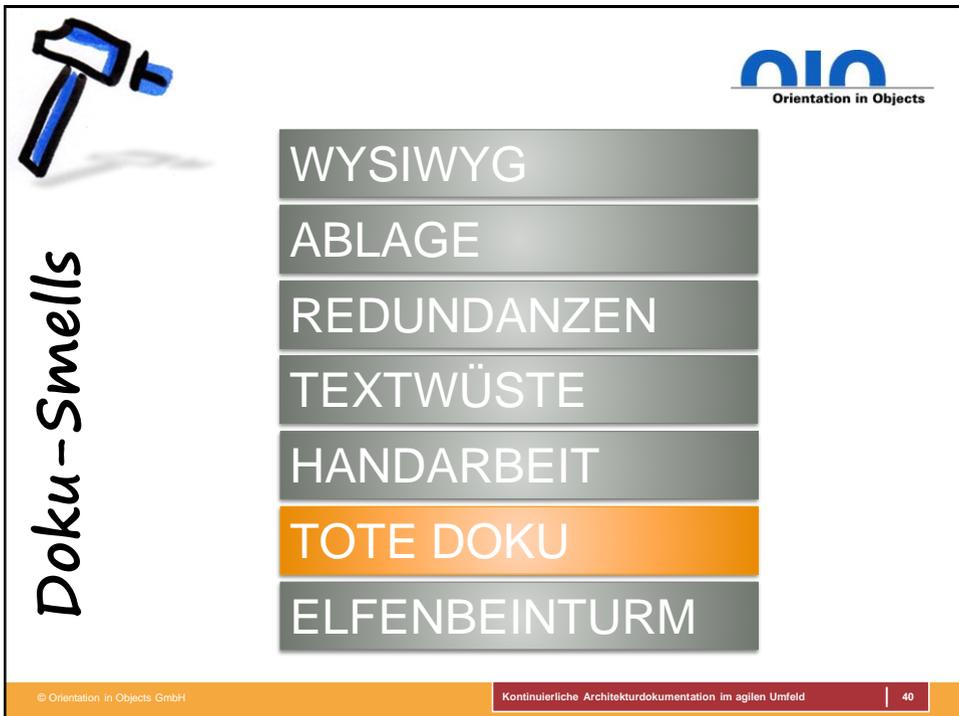
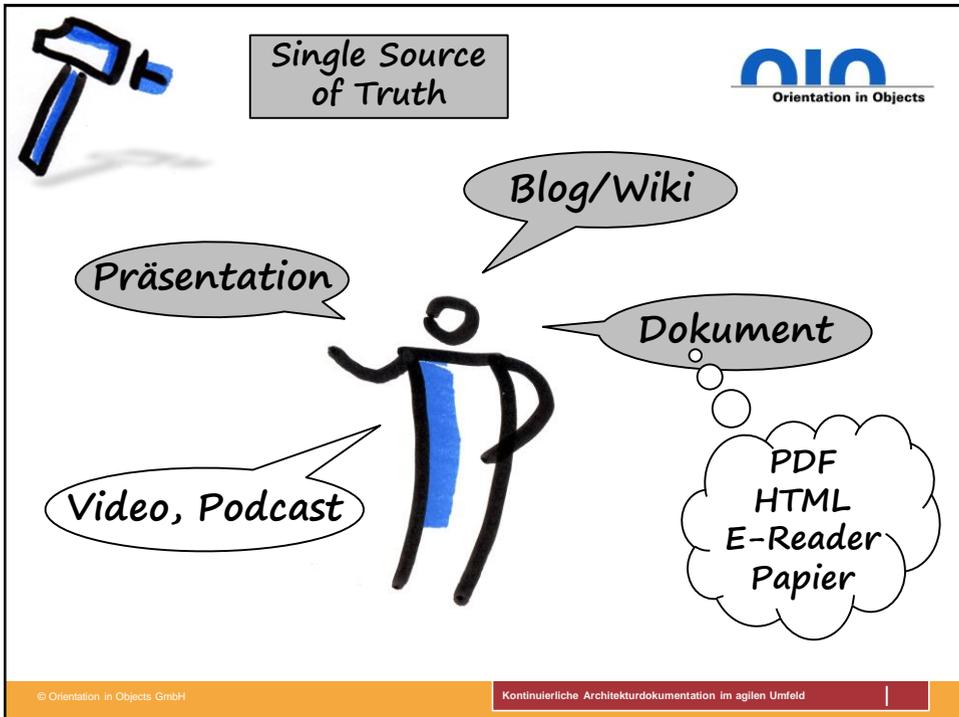
Foto von Ted Van Pelt: <https://www.flickr.com/photos/bantam10/5209157298/> (CC BY 2.0 Lizenz)

© Orientation in Objects GmbH

Kontinuierliche Architekturdokumentation im agilen Umfeld

36







nur so viel wie nötig
geringe Änderungsrate
Zielgruppenbedürfnisse
Feedback einpflegen

Pragmatik/Effektivität

Foto von PublicDomainPictures: <https://pixabay.com/de/ansager-audio-schwarz-kassette-316584/> (CC0 Public Domain Lizenz)
Foto von Devanath: <https://pixabay.com/de/bleistift-b%C3%BCro-design-kreative-1209544/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 41



Validierung

oio
Orientation in Objects

SanityChecks
Broken Links
PDEI Init

Foto von skeeze: <https://pixabay.com/en/military-honor-guard-inspection-642639/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 42

Ausführbare Dokumentation



Foto von Wikimedia: https://commons.wikimedia.org/wiki/File%3A2014_W6N_-_France_vs_Italy_-_Christelle_Le_Duff_5780.jpg (Creative Commons Attribution 3.0 Unported)

Ausführbare Tests
Einbettung in Dokument
Reportgenerierung

JGiven Report

search in scenarios

ALL SCENARIOS

SUMMARY
All Scenarios 3
Failed Scenarios 0
Pending Scenarios 0

TAGS
Page: Detail
Page: Start

CLASSES
org

All Scenarios

3 Successful, 0 Failed, 0 Pending, 3 Total (29.910s)

Group By Sort By Status Tags Classes

Detail Page Should show details page (13.411s) [Page: Detail](#)

Given start page opened in browser (6.257s)
When click on first talk (1.143s)
Then talk details are visible (302ms)
And talk abstract is visible (21ms)

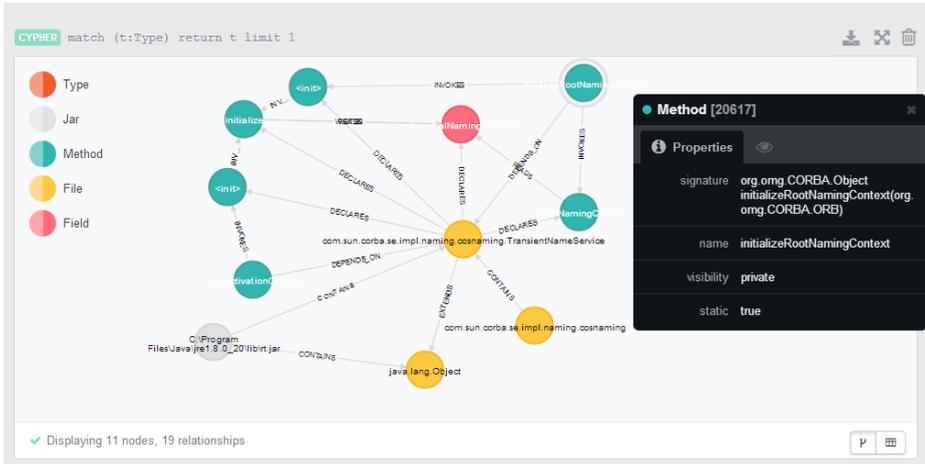
org.dukecon.server.gui.DetailPageTest

[Page: Start](#)

[Page: Start](#)

```
1 @Test
20 @DetailPageTag
21 public void should_show_details_page() throws InterruptedException {
22     given().start_page_opened_in_browser();
23     when().click_on_first_talk();
24     then().talk_details_are_visible()
25         .and().talk_abstract_is_visible();
26 }
```

Spezifikation



```

1 [[model:Default]]
2 [[role=group, includesConstraints="model:JpaEntityInModelPackage"]]
3 == JPA Model
4
5 The following constraints are verified:
6
7 - <<model:JpaEntityInModelPackage>>
8
9 == Constraints
10
11 [[model:JpaEntityInModelPackage]]
12 .All JPA entities must be located in packages named "model".
13 [source,cypher,role=constraint,requiresConcepts="jpa2:Entity"]
14 ----
15 MATCH
16 (package:Package) -[:CONTAINS]->(entity:Jpa:Entity)
17 WHERE
18 package.name <> "model"
19 RETURN
20 entity AS EntityInWrongPackage
21 ----

```

jQAssistant Report

Groups

#	Group Name	Date
1	default	2016-04-14T14:43:50
2	management:Default	2016-04-14T14:43:50
3	test:Default	2016-04-14T14:43:51
4	structure:Default	2016-04-14T14:43:52
5	model:Default	2016-04-14T14:43:55

Constraints

- Move the mouse over a constraint to view a description.
- Click on a failed constraint to open a details view.

#	Constraint Name	Count	Severity	Duration (in ms)
1	assertJ:TestMethodWithoutAssertion ✓	0	info	238
2	JUnit4:IgnoreWithoutMessage ✓	0	info	192
3	structure:ImplementationDependencies ✓	0	info	351
4	dependency:PackageCycles ✓	0	info	178
5	structure:ControllerDependencies ✓	0	info	46
6	structure:RepositoryDependencies ✓	0	info	40
7	structure:ServiceDependencies ✓	0	info	31
8	model:JpaEntityInModelPackage ✓	0	info	42

```
[[architecture:DefinedDependencies]
[plantuml,role=concept]
----
```

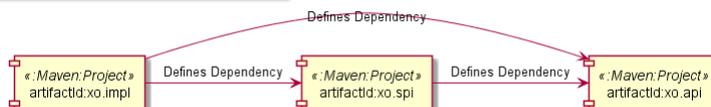
```
[artifactId:xo.impl] as impl <<:Maven:Project>>
[artifactId:xo.api] as api <<:Maven:Project>>
[artifactId:xo.spi] as spi <<:Maven:Project>>
```

```
impl -> api : Defines Dependency
impl -> spi : Defines Dependency
spi -> api : Defines Dependency
----
```

Architekturdefinition in PlantUML,
eingebettet in AsciiDoc

```
[[architecture:UndefinedDependencies]]
[source,cypher,role=constraint,requiresConcepts="architecture:DefinedDependencies"]
.There must not be dependencies between Maven
project which have not been defined.
```

```
----
MATCH
(p1:Maven:Project)-[:CREATES]->(:Artifact)-[:CONTAINS]->(t1:Type),
(p2:Maven:Project)-[:CREATES]->(:Artifact)-[:CONTAINS]->(t2:Type),
(t2)-[:DEPENDS_ON]->(t1)
WHERE NOT
(p1)-[:DEFINES_DEPENDENCY]->(p2)
RETURN
*
----
```





Doku-Smells

- WYSIWYG
- ABLAGE
- REDUNDANZEN
- TEXTWÜSTE
- HANDARBEIT
- TOTE DOKU
- ELFENBEINTURM**



Klassische
Architekten
Rolle

Foto von scottwebb: <https://pixabay.com/en/toronto-architecture-skyscraper-1594940/> (CC0 Public Domain Lizenz)



Architekt/Senior Developer als
Ratgeber/Mentor
und **Moderator**

Foto von JESHOOITS: <https://pixabay.com/de/warten-termin-zeitplan-zeit-eile-410328/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 51



Vorbereiten
Planen
Erinnern
Delegieren
Integrieren
Prüfen

Dokumentation braucht einen
Kümmerner

Foto von Berzin: <https://pixabay.com/en/ambulance-doctor-students-game-2166079/> (CC0 Public Domain Lizenz)

© Orientation in Objects GmbH | Kontinuierliche Architekturdokumentation im agilen Umfeld | 52



Doku-Smells

- WYSIWYG
- ABLAGE
- REDUNDANZEN
- TEXTWÜSTE
- HANDARBEIT
- TOTE DOKU
- ELFENBEINTURM



Zusammenfassung

- ~~WYSIWYG~~ Plain Text
- ~~ABLAGE~~ Documentation as Code
- ~~REDUNDANZEN~~ Generierung
- ~~TEXTWÜSTE~~ Mehr Grafiken
- ~~HANDARBEIT~~ Automatisierung
- ~~TOTE DOKU~~ Lebendige Dokument.
- ~~ELFENBEINTURM~~ Kümmerer



Szenario 1: Markdown, Pandoc, PlantUML, yEd



- Schreiben in Markdown in IDE (IntelliJ IDEA) inkl. Preview
- PDF-Erzeugung mit PanDoc über LaTeX-Zwischenschritt inkl. Corporate Design
- UML-Diagramme mit PlantUML, Live-Ansicht/Export aus IDE
- andere Diagramme mit yEd, Export als *.png
- Stash/Bitbucket Server als Repo
 - rendert Markdown direkt in Weboberfläche
 - readme.md Verlinkungen auf wichtige Dokumente

Szenario 2: AsciiDoctor, Maven, PlantUML



- Erstellen AsciiDoc und PlantUML in IDE mit Preview
- Maven-Plugin zum Erzeugen des HTML/PDF



AsciiDoctor

Maven™

Szenario 3: Generierung aus Quellcode



- Quellcode parsen
 - Reflection
 - Spring Kontext
 - ...
- in Unit-Test aus Klassen-Strukturen Diagramm-Markup erzeugen
 - z. B. PlantUML
 - als Text-Datei ablegen und in Markup-Dokumentation verlinken
- im Build-Prozess als Input für Markup-Konvertierung einlesen

Szenario 4: Schnittstellenbeschreibung



- Generierung aus WSDL, WADL, Swagger
- Einbindung in Build-Prozess
- Swagger2Markup
- JAX-RS Analyzer
- Spring REST Docs



Szenario 5: Ausführbare Dokumentation



- Quellcode-Struktur in Graph-Datenbank importieren
- Architektur-Regeln als Graph-Abfragen in AsciiDoc einbetten





Orientation in Objects

???

Fragen ?

???

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de



Orientation in Objects

**Vielen Dank für Ihre
Aufmerksamkeit !**

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de