

# Systematische Architekturdokumentation & Tools



Stand: April 2020

© Diese Unterlagen sind urheberrechtlich geschützt von Dr. Peter Hruschka und Dr. Gernot Starke.  
Jede Verwendung außerhalb der engen Grenzen des Urheberrechts ist ohne schriftliche Zustimmung der Autoren unzulässig und strafbar.  
Dies gilt insbesondere für Vervielfältigungen, Übersetzungen sowie Speicherung und Verarbeiten in elektronischen Systemen.

[www.arc42.de](http://www.arc42.de)

A blue curved line, similar to the one in the logo, located in the bottom right corner of the page.

# Ziele und Inhalt



Sie lernen

- Anforderungen an Architekturdokumentation
- Typische Architekturdokumente
- Organisationsformen für Architekturdokumentation
- Werkzeuge für Architekturdokumentation

Lernziele gemäß iSAQB CPISA-F:

- LZ 3-1: Qualitätsmerkmale technischer Dokumentation erläutern und berücksichtigen (R1)
- LZ 3-2: Softwarearchitekturen beschreiben und kommunizieren (R1)
- LZ 3-8: Architekturentscheidungen erläutern und dokumentieren (R2)
- LZ 3-9: Dokumentation als schriftliche Kommunikation nutzen (R2)
- LZ 3-10: Weitere Hilfsmittel und Werkzeuge zur Dokumentation kennen (R3)

Anforderung		Lösung
korrekt/ richtig	In Bezug auf <ul style="list-style-type: none"> <li>• Sourcecode</li> <li>• Schnittstellen</li> <li>• Aktuel</li> </ul>	<ul style="list-style-type: none"> <li>• Eine verantwortliche Person</li> <li>• Review / 4-Augen-Prinzip</li> </ul>
(effizient) wartbar	Leicht anpassbar	<ul style="list-style-type: none"> <li>• Abstraktion statt (zu viele) Details</li> <li>• Templates verwenden</li> <li>• Redundanz vermeiden</li> </ul>
verständlich	Für relevante Lesergruppen	<ul style="list-style-type: none"> <li>• Sichten verwenden</li> <li>• Bedürfnisse im Vorfeld klären</li> <li>• Reviews</li> </ul>
entsteht begleitend		<ul style="list-style-type: none"> <li>• Kontinuierlich dokumentieren</li> <li>• Templates</li> </ul>
angemessen, einfach	In Bezug auf <ul style="list-style-type: none"> <li>• Umfang</li> <li>• Detailtiefe</li> <li>• Abstraktion</li> </ul>	Angemessenheit anhand von Risiken, Komplexität, Stakeholdern, Kontext, Größe, Termin, Budget, Organisation, Team, Domäne und <N.N.> explizit bewerten.
auffindbar		<ul style="list-style-type: none"> <li>• Standard-Directory-Strukturen/-laufwerke, read-me, index.html, ...</li> </ul>

# Wichtige Architekturdokumente

## a) **Architektur-Referenzdokumentation** (Architektur-Repository)

- Zentraler „Behälter“
- Vorstrukturiert: Template(s)
- inkl. Erklärung von Begriffen und Methode  
(Handbuch-zur-Architekturdokumentation)
- Ermöglicht „Generierung“ spezifischer Ergebnisse, z.B.:
  - Betriebsdoku
  - Architekturpräsentation
  - <N.N.>

## b) Architekturpräsentation (Extrakt aus a.)

## c) Architekturtapete

## d) Doku-Landkarte

- sollte von Projektleitung erstellt werden

- Whitebox-Template**
1. Name
  2. Überblick (Diagramm!)
  3. Begründung
  4. Enthaltene Blackboxes
  5. Interne Schnittstellen
  6. Offene Punkte

- Blackbox-Template**
1. Name
  2. Zweck / Verantwortlichkeit
  3. Schnittstellen
  4. Ablageort / Datei
  5. Erfüllte Anforderungen
  6. Variabilität
  7. Offene Punkte

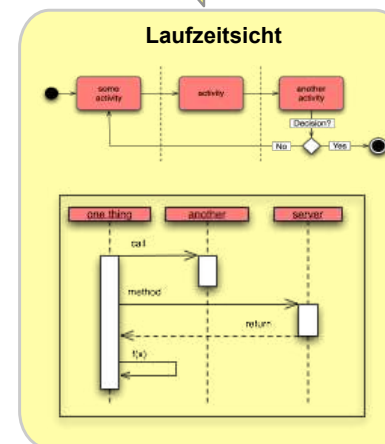
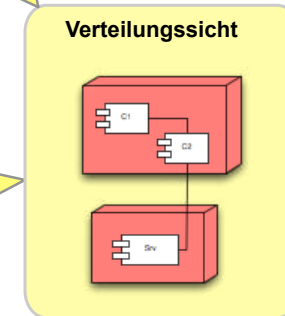
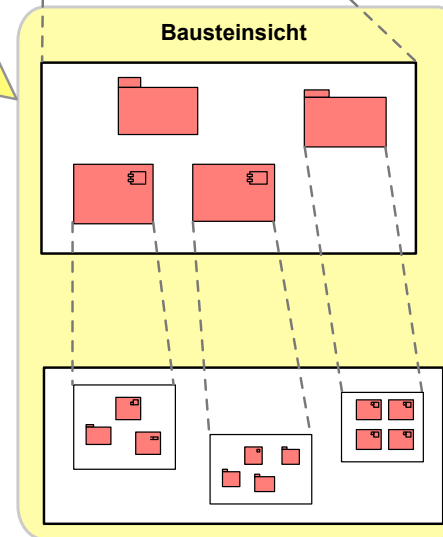
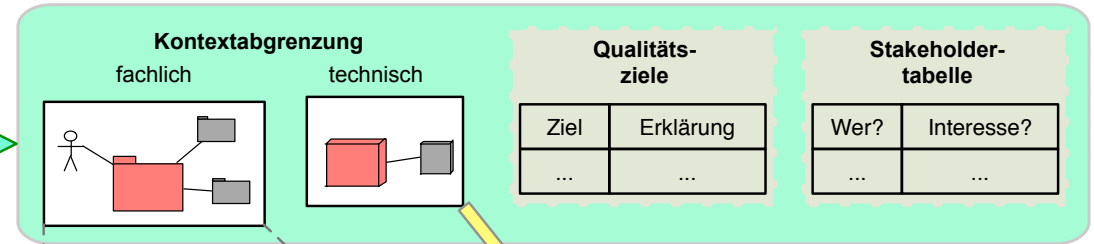
- Struktur von Entwurfsentscheidungen**
1. Was muss entschieden werden (Fragestellung)?
    - 1.1 In welchem Kontext?
  2. Entscheidungskriterien
  3. Mögliche Alternativen
  4. Wie wurde entschieden?
    - 4.1 Warum?
    - 4.2 Getroffene Annahmen
    - 4.3 Verworfen Alternativen
  5. Konsequenzen?
  6. Bekannte Risiken?
  7. Wer hat wann entschieden?

## ARC42 Architektur-Dokumentation

1. Einführung und Ziele
2. Randbedingungen
3. Kontextabgrenzung
4. Lösungsstrategie
5. Bausteinsicht
6. Laufzeitsicht
7. Verteilungssicht
8. Konzepte
9. Entwurfsentscheidungen
10. Qualitätsszenarien
11. Risiken & techn. Schulden
12. Glossar

- Szenarien zur Definition von Qualitätsanforderungen**
1. Qualitätsbaum
  2. Anwendungsszenarien/ Änderungsszenarien

- Struktur von Konzepten**
1. Ziele und Anforderungen
  2. Randbedingungen
  3. Scope / Kontext
  4. Lösung / Vorgehen
    - 4.a Strukturen & Abläufe
    - 4.b Beispiele inkl. Code
  5. Betrachtete Alternativen
  6. Risiken



# Kein Geld? Keine Zeit?

- Falls für Gesamtdoku kein Budget ...
- Highlights, die jeder über Architektur wissen muss
- Gliederung: Ähnlich Architekturpräsentation

Pragmatismus!

Einfach, kein Aufwand für Tools!

- Was soll das System leisten?
- Randbedingungen

## ■ Kontextabgrenzung

## ■ Bausteinsicht (Level 1)

- Laufzeitszenarien
- Verteilungssicht

## ■ Technische Konzepte

- Zentrale Entscheidungen

# Praxistipps Dokumentation

- **Schreibe und sprich Top-Down**
  - Stakeholder „abholen“
- Verwende standardisierte Struktur
  - Templates sind Dein Freund (arc42.de / arc42.org)
- **Kombiniere Bild mit Text**
- Formuliere eindeutig
  - Eindeutigkeit wichtiger als Interessantheit
- **Redundanzfrei** – aber komfortabel
  - Kontrollierte Wiederholung statt extensiver Querverweise
- Beginne sparsam
  - Erfrage Bedürfnisse der Stakeholder
- **ABNNC: accurate, but not neccessarily complete**
  - Mut zur Lücke
- Dokumentiere:
  - Technische Schuld
  - Überraschungen
  - Offene Punkte
  - Getroffene Annahmen
  - Gründe



# Werkzeuge zur Dokumentation

## (UML-) Modellierungswerkzeuge

(Enterprise Architect, VisualParadigm, MagicDraw, IdeasModeler...)

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>+ Effiziente Erstellung &amp; Pflege von Diagrammen</li> <li>+ Gute Navigierbarkeit in Modellen</li> </ul> | <ul style="list-style-type: none"> <li>- Mittelmäßige Integration in Code-Repository</li> <li>- Aufwändige/schlechte Reportgeneratoren</li> <li>- Schlecht für Text + Tabellen</li> </ul> |
|---|---|

## Textwerkzeuge

(Word, OpenOffice, ASCIIDoc, Markdown, LaTeX, DocBook)

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>+ (oft) universelle Verfügbarkeit</li> <li>+ gute Integration von Text, Tabellen + Diagrammen</li> </ul> | <ul style="list-style-type: none"> <li>- Teilweise schlechte Integration von Diagrammen</li> <li>- Teilweise schlechte Mehrbenutzerfähigkeit</li> <li>- Schlechte Skalierbarkeit auf große Dokumente</li> </ul> |
|---|---|

## Wiki

(Confluence, MediaWiki, Twiki, ...)

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>+ sehr gute Mehrbenutzerfähigkeit</li> <li>+ gute Suche/Navigierbarkeit</li> <li>+ gute Integration mit Issue Tracker</li> </ul> | <ul style="list-style-type: none"> <li>- Kaum Offline-Fähigkeit</li> <li>- Versionierung getrennt von Source</li> </ul> |
|---|---|

## Zeichenwerkzeuge

(Visio, Graffle, Gliffy, DrawIO, PowerPoint, Yed, ...)

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>+ attraktive Resultate</li> <li>+ hohe Flexibilität in Darstellung</li> </ul> | <ul style="list-style-type: none"> <li>- Keine Weiterverwendbarkeit von Diagrammen</li> <li>- Zeichnen statt modellieren</li> <li>- Keine Standardisierung</li> </ul> |
|--|---|



# (einige) Kriterien zur Werkzeugwahl

- **Einfachheit der Benutzung**

- Erstellung von Doku
- Finden / Lesen von Doku
- Volltextsuche

- Verfügbarkeit im Team

- Integration mit Code-Repository

- Merge-Fähigkeit der Inhalte

- **Integration Text, Tabellen und Diagramme**

- Robustheit / Ausfallsicherheit

- Rechte & Rollenkonzept

- Zugriffsschutz

- ▶ Automatisierbarkeit

- Generierung benötigter Artefakte (Reportgenerator)
- Integration in Daily-Build
- Offene Datenformate

- ▶ UML-(Metamodel) Compliance

- ▶ Flexibilität der Modellierung

- ▶ Lizenz- und Betriebskosten

- Lizenzmodell

- ▶ **Multi-User-Fähigkeit**

- ▶ Offline-Fähigkeit



# Mit arc42 im Projekt...

Team



## Projektdokumentation

(gemeinsames mentales  
Modell)

Eher  
volatil

„Gärtner“



## Systemdokumentation

stabil

# Mit arc42 im Projekt...

Team



„Gärtner“



## Projektdokumentation

~arc42 „locker“

Diskussionen

Implementation-Guide,

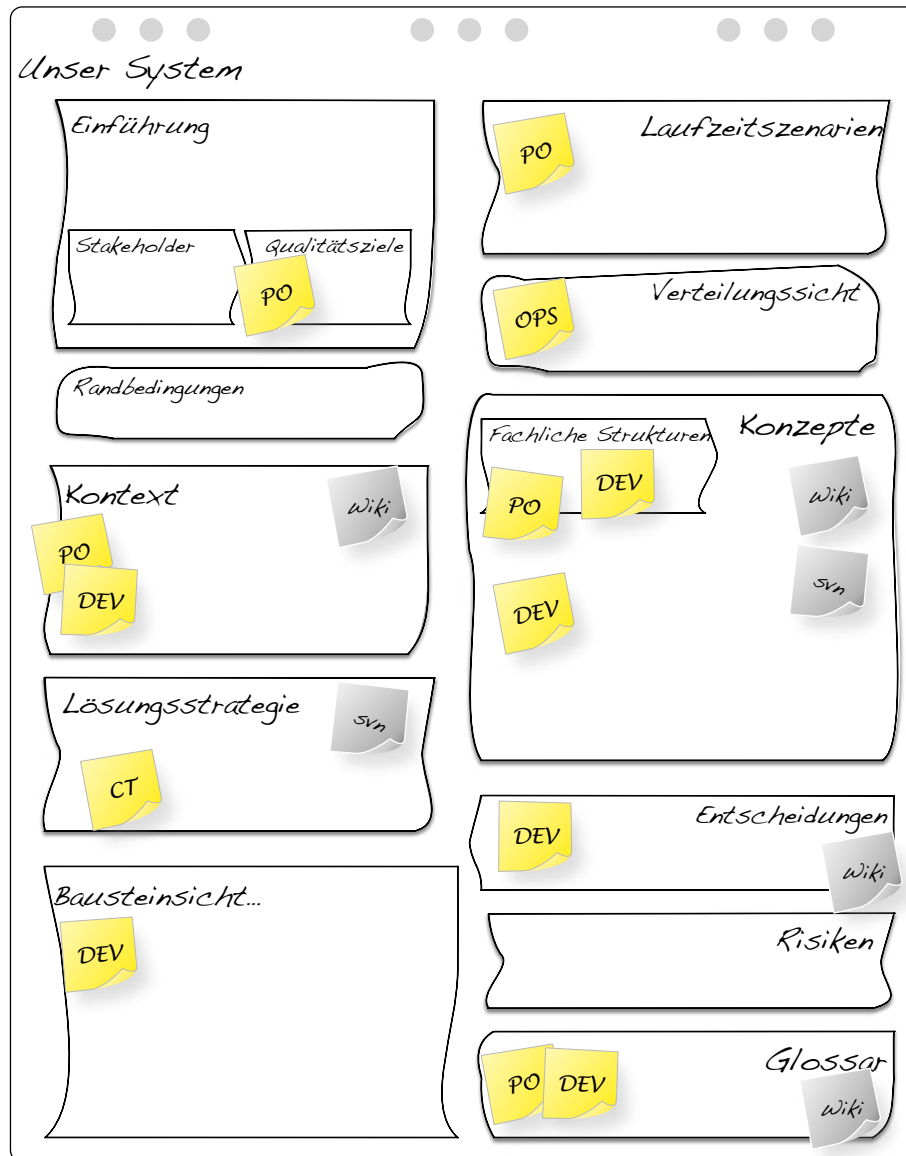
Tasks / Issues

## Systemdokumentation

1. Einführung und Ziele
2. Randbedingungen
3. Kontextabgrenzung
4. Lösungsstrategie
5. Bausteinsicht
6. Laufzeitsicht
7. Verteilungssicht
8. Konzepte
9. Entwurfsentscheidungen
10. Qualitätsszenarien
11. Risiken
12. Glossar

Weitere relevante Infos...  
(Betrieb/Admin, Test, Release...)

# Nachdokumentation..

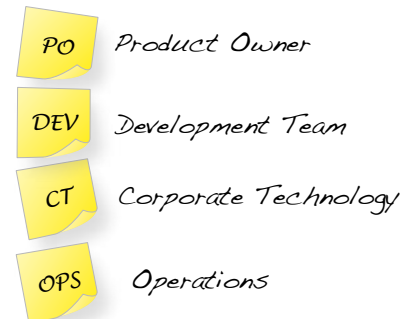


## Timebox 1 (15min):

- Was brauchen wir?
- Was haben wir schon?

## Timebox 2: (15min)

- In kleinen Gruppen Resultate skizzieren



# Architekturdokumentation und Tools

- Egal, welches Tool Sie einsetzen:
  - Teile der Dokumentation können verantwortlich vom Entwicklungsteam erstellt werden.
  - **ABER: Architekten verantwortet die Zusammenführung von Teilergebnissen zu einer Gesamtarchitektur(dokumentation) und deren konsistenten Stil.**

„One Throat to Choke“

(Adrenalin Junkies & Formular Zombies, DeMarco et al.)

# Zusammenfassung



„Dokumentieren“ ist wichtige Architektentätigkeit:

- Dokumentieren macht Entscheidungen über längere Zeit nachvollziehbar
- Viele Informationen/Entscheidungen stehen nicht im Quellcode (obwohl sie Konsequenzen für diesen Code besitzen)

Tipps:

- Gründe festhalten
- (Zu viele) Details vermeiden
- Template verwenden (z.B. arc42)

