

OZG-Umsetzung

Wie kann eine Organisation
ihr Vorgehen schnell optimieren?

Andreas Mieth, 14.02.2021



Wie kann eine Organisation ihr OZG-Vorgehen schnell optimieren?

- Zusammenfassung
- Analyse des aktuellen Stands
- Optimierung
 - Methodische Grundlagen
 - Maßnahmen
- Nächste Schritte
- Kontakt



Zusammenfassung: Wie kann eine Organisation ihr OZG-Vorgehen schnell optimieren?

- Warum?
 - Die OZG-Umsetzung ist ein Großprojekt: viele Ergebnisse, Beteiligte, Versionen, Jahre
 - Ein Ad-hoc-Vorgehen¹ ist riskant, vor allem im Bezug auf Skalierung und Nachnutzung
- Wie? – Methodische Grundlagen
 - Die Optimierung basiert auf nutzer-zentriertem Design und Software Engineering
- Wie? – Maßnahmen
 - Die Optimierung erfolgt in kurzen Inkrementen mit Feedback-Zyklen, ausgerichtet auf Risiko-Reduzierung und positives Kosten/Nutzen-Verhältnis
 - Durch Kanban-ähnliche Projektsteuerung ermöglicht sie sichere Navigation in volatilem Umfeld, Kosten-Nutzen-Optimierung und kontinuierliche Verbesserung
 - Durch Model Driven Development (MDD) fördert sie Skalierung und Nachnutzung

¹ Gewohntes Vorgehen ungeachtet der Größe und Komplexität des Vorhabens



Warum?

Die OZG-Umsetzung ist ein Großprojekt: viele Ergebnisse, Beteiligte, Versionen, Jahre
Ein Ad-hoc-Vorgehen ist riskant, vor allem im Bezug auf Skalierung und Nachnutzung

OZG: digitale Angebote öffentlicher Leistungen

Die Interaktion zwischen Bürgerinnen, Bürgern und Unternehmen mit der Verwaltung soll in Zukunft deutlich

- **schneller,**
- **effizienter und**
- **nutzerfreundlicher**

werden. Das Gesetz zur Verbesserung des Onlinezugangs zu Verwaltungsleistungen

Onlinezugangsgesetz (OZG)

verpflichtet daher Bund, Länder und Kommunen,

bis Ende 2022

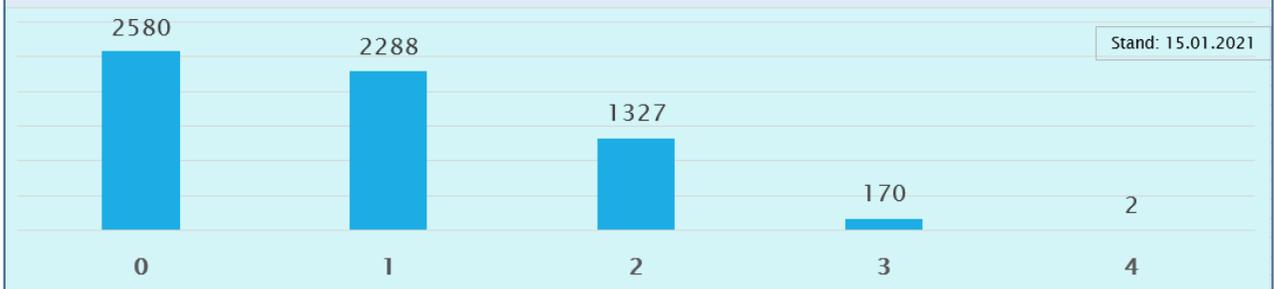
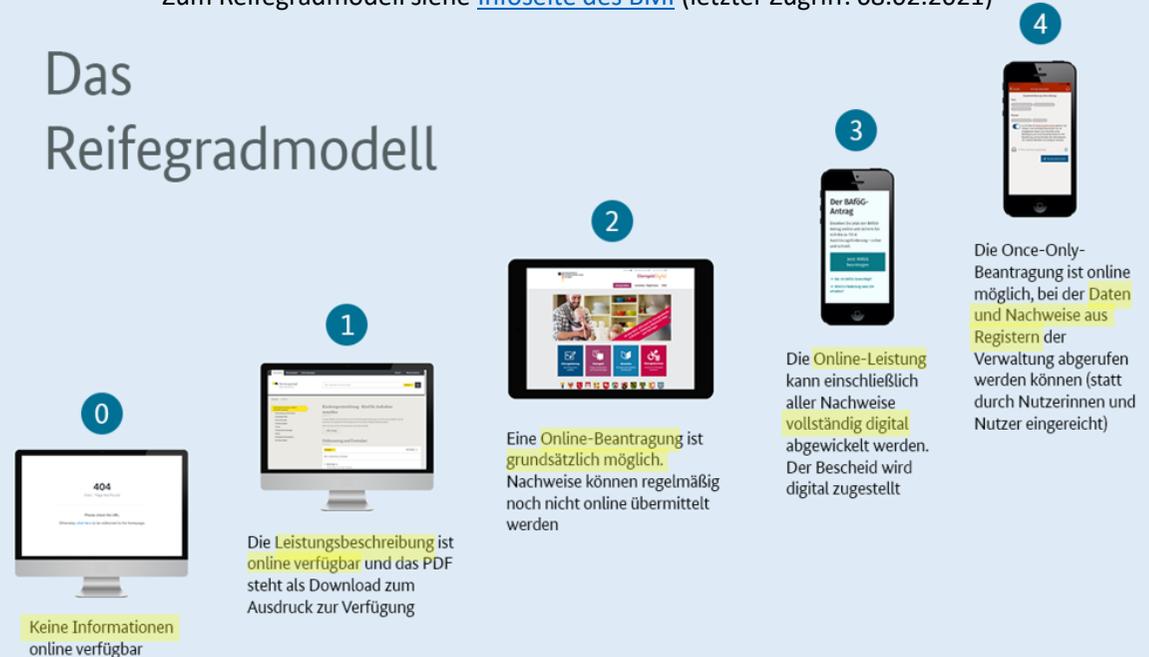
ihre Verwaltungsleistungen über Verwaltungsportale auch digital anzubieten.

Quelle: [BMI](#) (letzter Zugriff: 02.02.2021)

14.02.2021

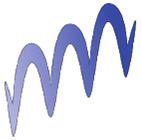
Zum Reifegradmodell siehe [Infoseite des BMI](#) (letzter Zugriff: 08.02.2021)

Das Reifegradmodell

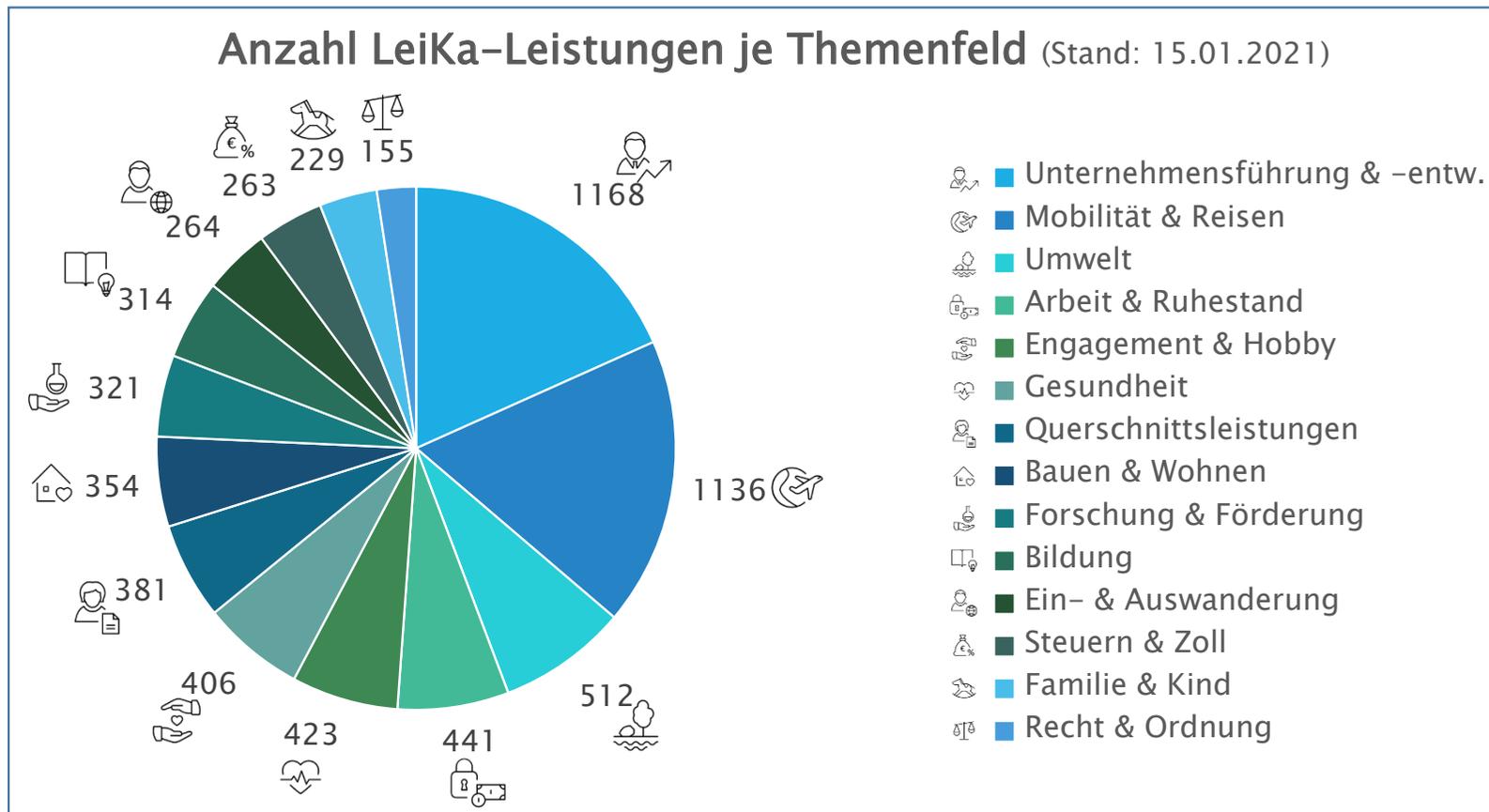


OZG-Optimierung, Version 1.0

5



Die OZG-Umsetzung umfasst mehr als 6000 öffentliche Leistungen



Jeder Punkt steht für eine LeiKa-Leistung; Stand: 15.01.2021

**Bund
oder
Land**

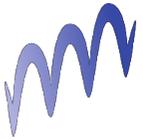
**1845 LeiKa-
Leistungen**

14.02.2021

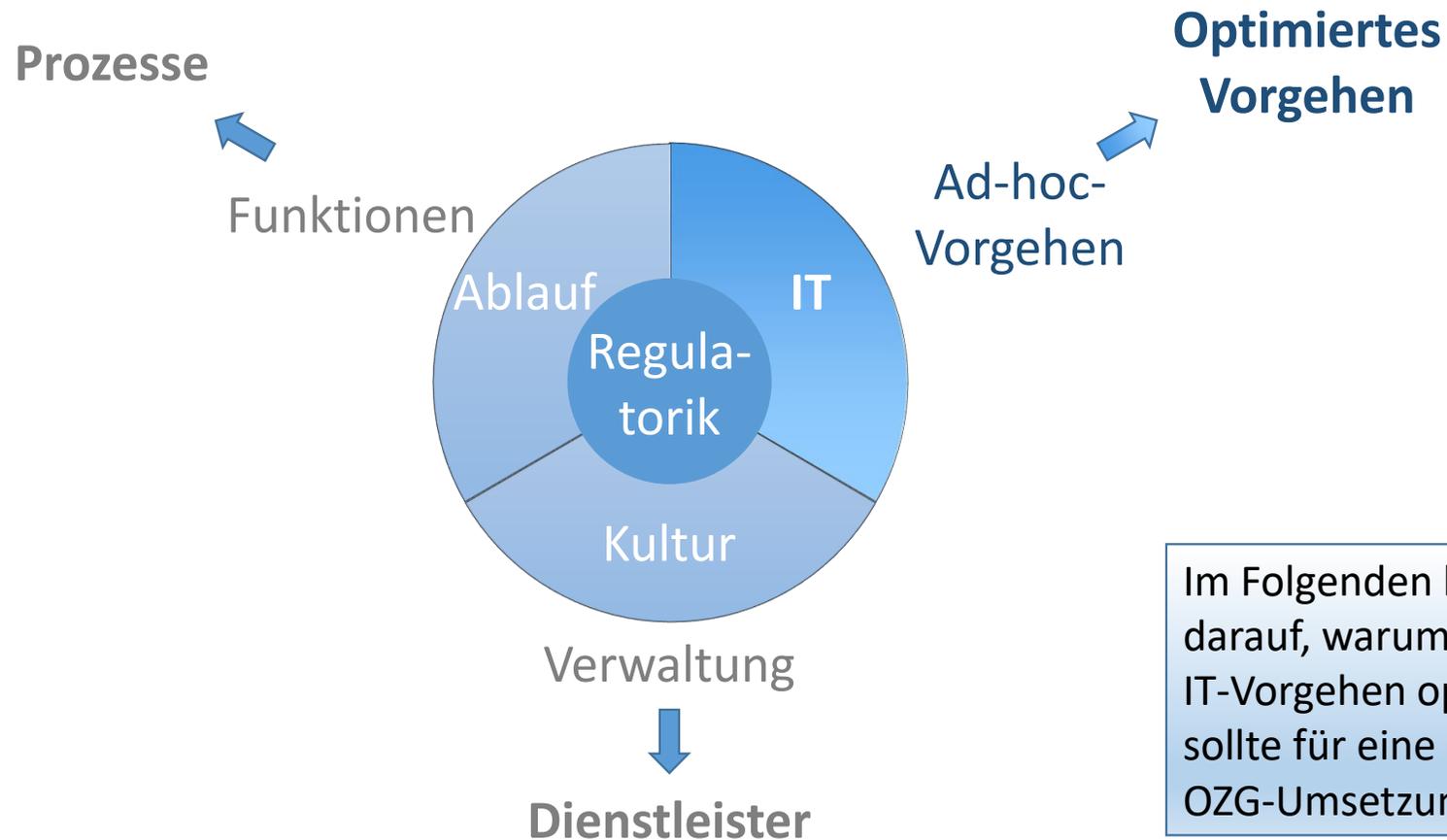
**Land
oder
Kommune**

**4522 LeiKa-
Leistungen**

OZG-Optimierung, Version 1.0

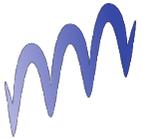


Das OZG erfordert einen komplexen Wandel



Im Folgenden liegt der Fokus darauf, warum und wie das IT-Vorgehen optimiert werden sollte für eine erfolgreiche OZG-Umsetzung

Ein Ad-hoc-Vorgehen ist risikoreich im Bezug auf Skalierung und Nachnutzung



Ad-hoc-Vorgehen	Nachteile
Keine fachliche Spezifikation , sondern techniknahe Umsetzungs-Vorlagen	Lückenhafte Spezifikation des Fachwissens erschwert Nachnutzung und Wartung; Techniknähe erhöht Aufwand bei Produkt-Update und Produkt-Ablösung
Viele Redundanzen in den Umsetzungs-Vorlagen	Höherer Entwicklungs- und Wartungs-Aufwand; Gefahr verschiedener Lösungen für dasselbe Problem
Keine Spezifikation der Nutzungs-Kontexte	Umsetzung teilweise organisations-spezifisch => erschwert Nachnutzung
Keine Spezifikation der Kanal-Kontexte	Umsetzung teilweise kanal-spezifisch (Online, PDF,...) => erschwert Nachnutzung
Manuelle Durchführung schematischer Aufgaben	Ist fehlerträchtig und erhöht so den Aufwand für Test und Fehlerbehebung

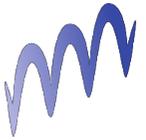


Wie?

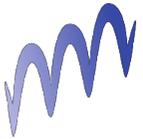
Methodische Grundlagen

Die Optimierung basiert auf nutzer-zentriertem Design und Software Engineering

Die Maßnahmen zur Optimierung werden aus strategischen Zielen abgeleitet



Digitale Leistungen werden nur dann ein Erfolg, wenn man sie gerne benutzt



Anbieter-zentriert: Such-Maschine



OZG-Pendant:

Verwaltungs-zentrierte Vorgänge und Formulare

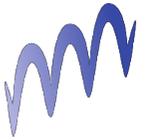
Nutzer-zentriert: Finde-Maschine

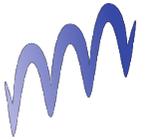


OZG-Pendant:

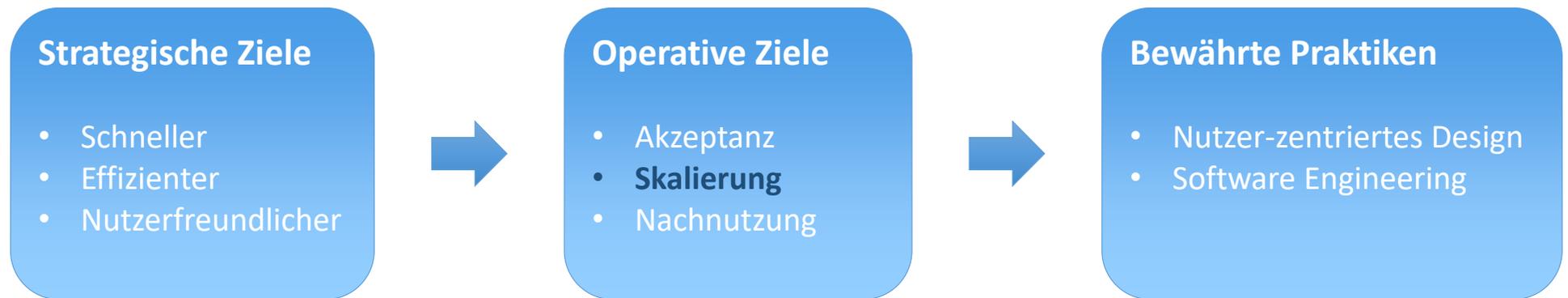
Digitale Leistungen entworfen für die Nutzer

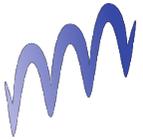
Nutzer von digitalisierten Verwaltungsleistungen sind nicht nur die direkten Benutzer



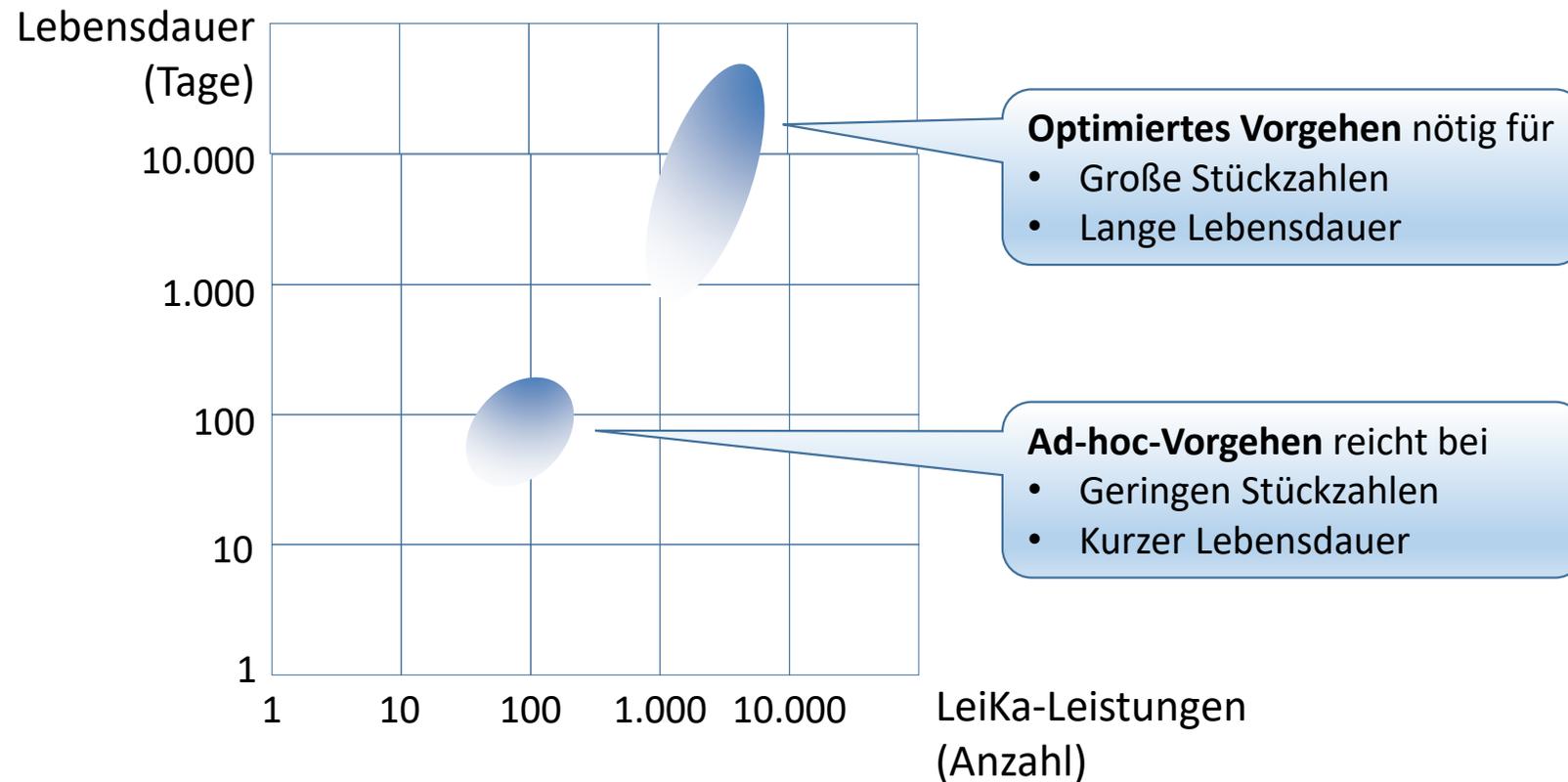


Ein skalierbares Vorgehen sichert den Erfolg bei großen Mengen und langer Nutzung

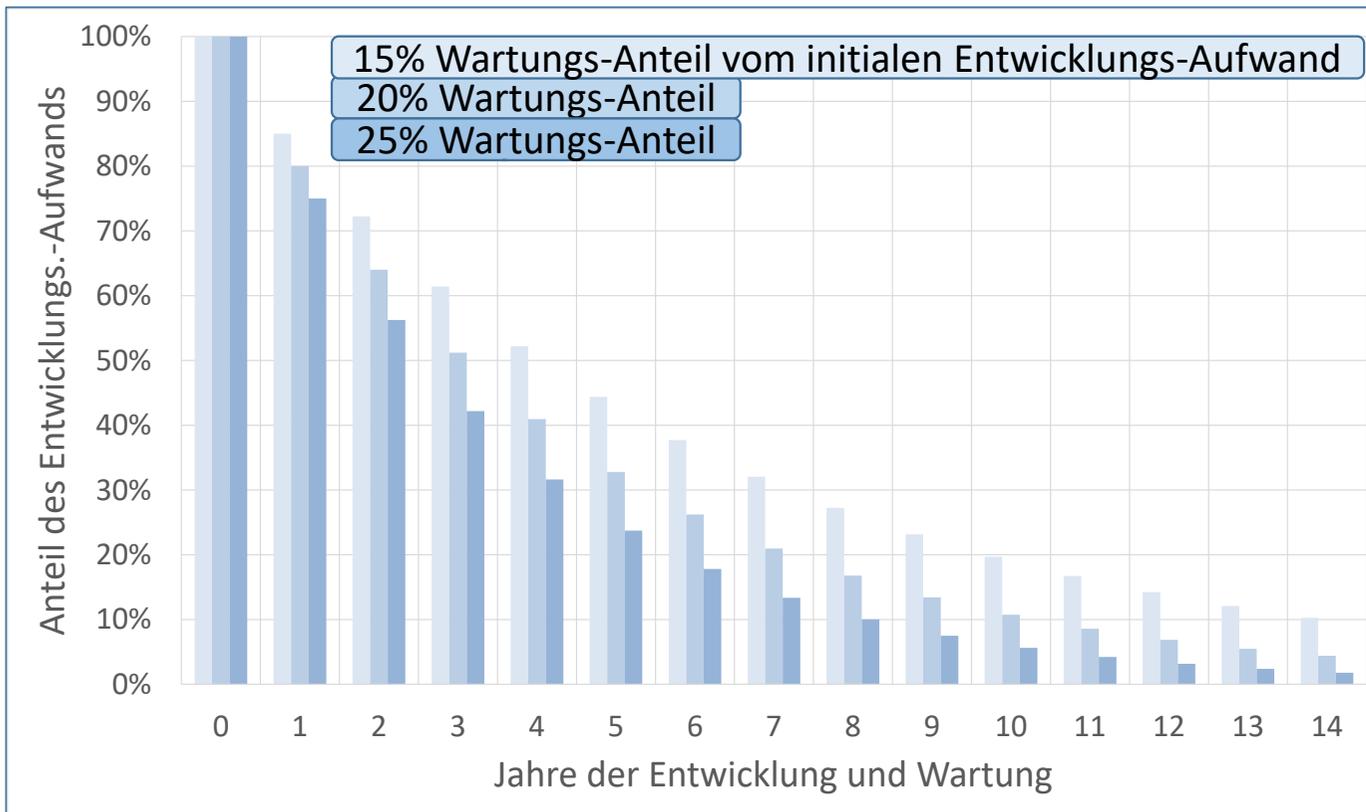
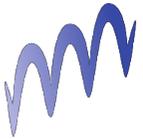




Die OZG-Umsetzung soll um mehrere Größenordnungen wachsen



Der Entwicklungs-Anteil schrumpft im Lauf der Zeit stark gegenüber der Wartung

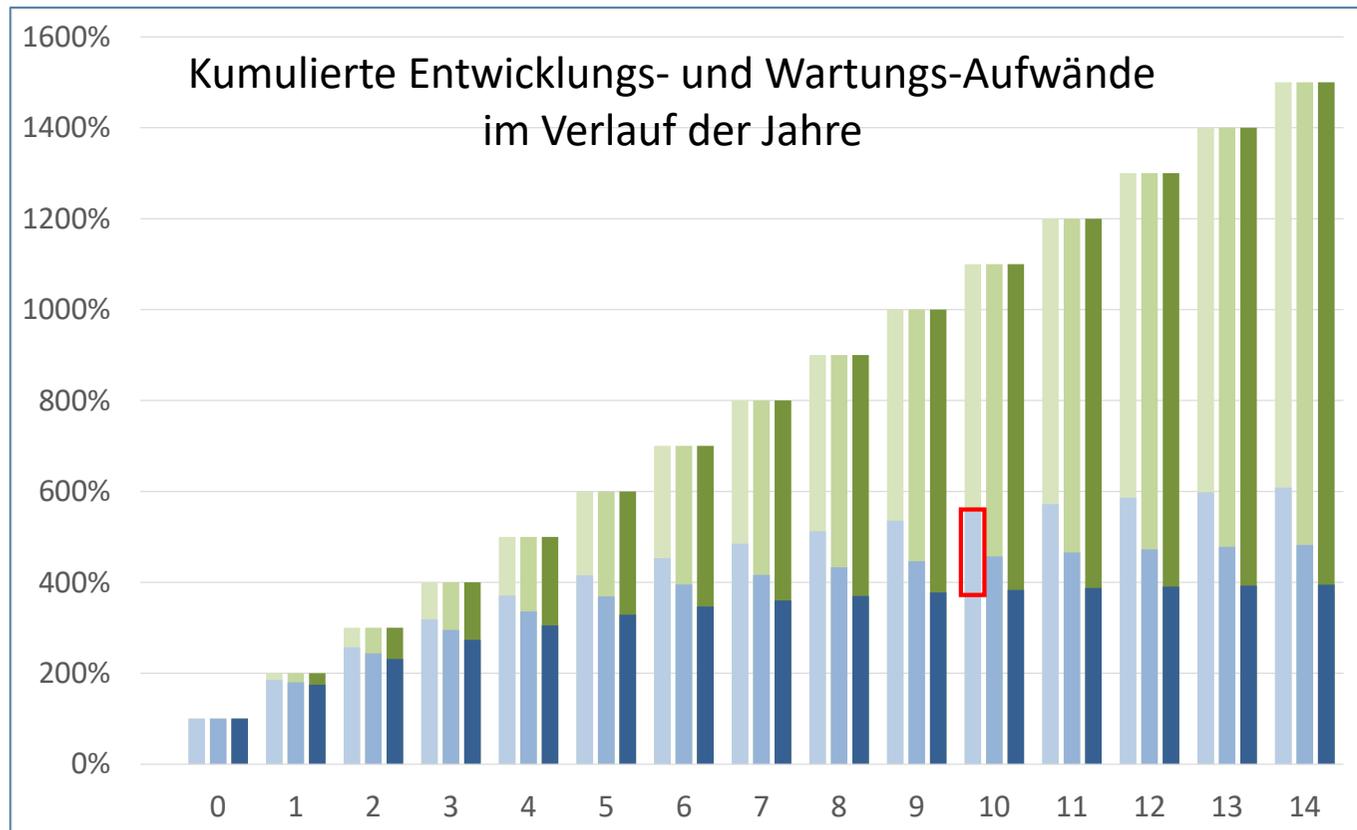
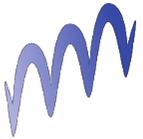


Prämisse:
Gesamt-Kapazität = 100%

Kleine Änderungen des Wartungs-Anteils wirken sich deutlich aus auf den Entwicklungs-Anteil.

Daher lohnen sich selbst geringe Verbesserungen der Wartbarkeit.

Schon nach 3 bis 4 Jahren ist kaum noch Entwicklung möglich aufgrund der Wartung

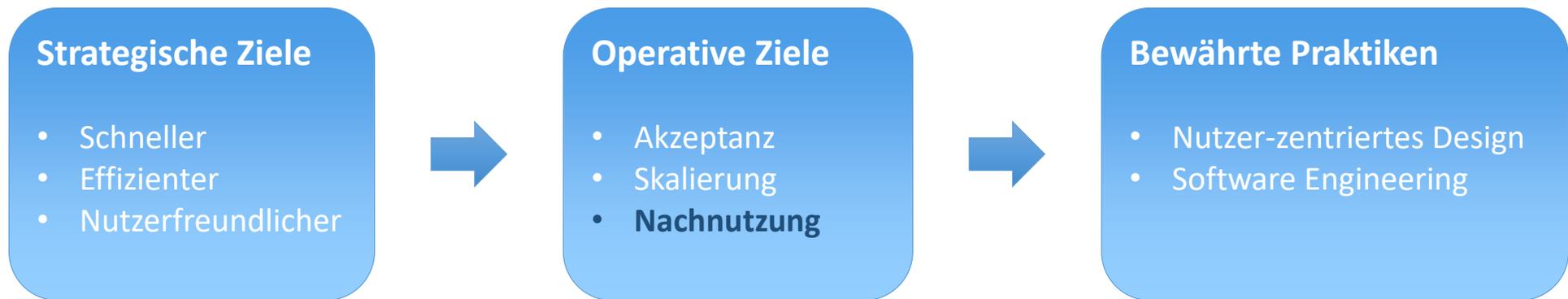
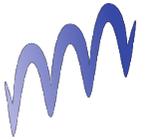


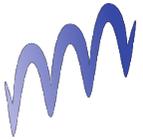
	Wartungs-Anteil		
	15%	20%	25%
Wartung			
Entwicklung			

Prämisse:
Gesamt-Kapazität = 100%

Ein geringer Wartungs-Anteil zahlt sich aus: 15% statt 25% bedeuten, dass innerhalb von 10 Jahren fast 2 Jahres-Budgets mehr für die Entwicklung bereitstehen.

Erfolgreiche Nachnutzung fördert die Qualität und reduziert den Aufwand





Die Nachnutzung kann auf verschiedenen Ebenen stattfinden

Nachnutzung „im Großen“

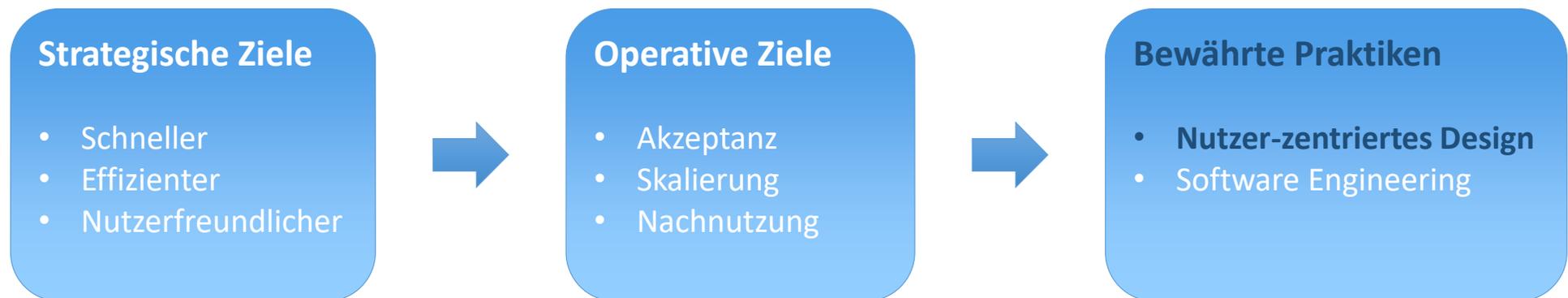
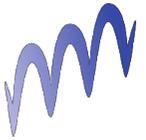
- [Modell A](#)
 - Einer für Alle/Viele
- [Modell B](#)
 - Nachnutzbare Software dezentral betrieben
- [Modell C](#)
 - FIM-basierte Eigenentwicklung

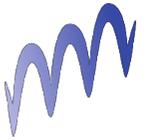
Nachnutzung „im Kleinen“

- Redundanzfrei spezifizieren
 - Mit Datentypen, Bausteinen usw.
 - Ggf. redundante Prüf-Sichten generieren
- Redundanzfrei umsetzen
 - Programmierung gemäß DRY*
 - Ggf. redundanten Code aus nicht-redundanten Quellen generieren

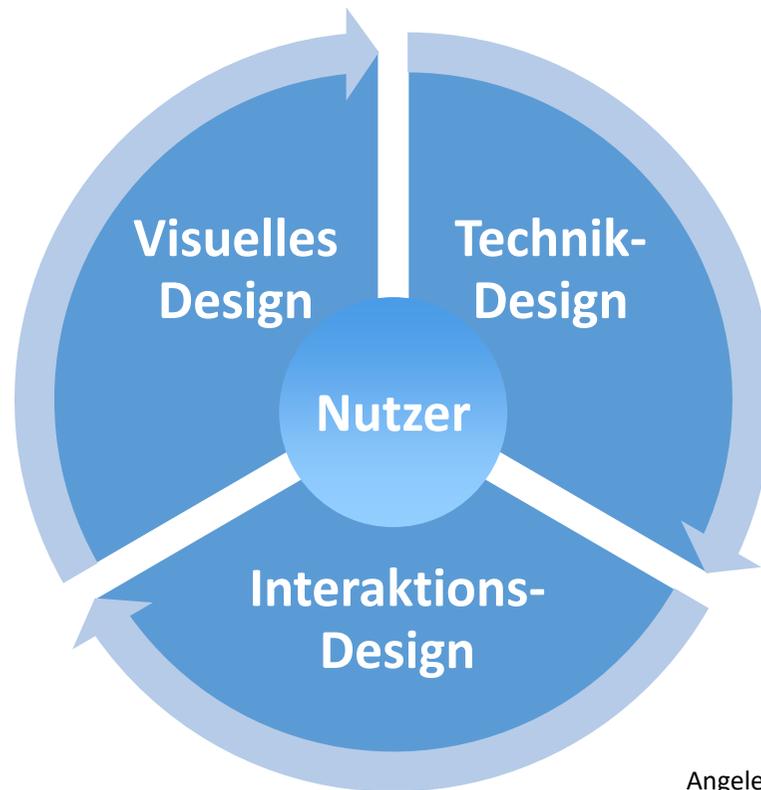
* Don't Repeat Yourself; siehe „The Pragmatic Programmer“, 2. Auflage, 2020.

Die operativen Ziele werden erreicht durch Anwendung bewährter Praktiken



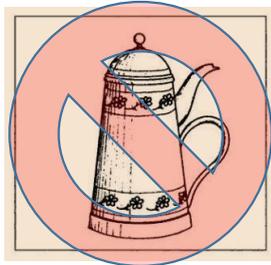
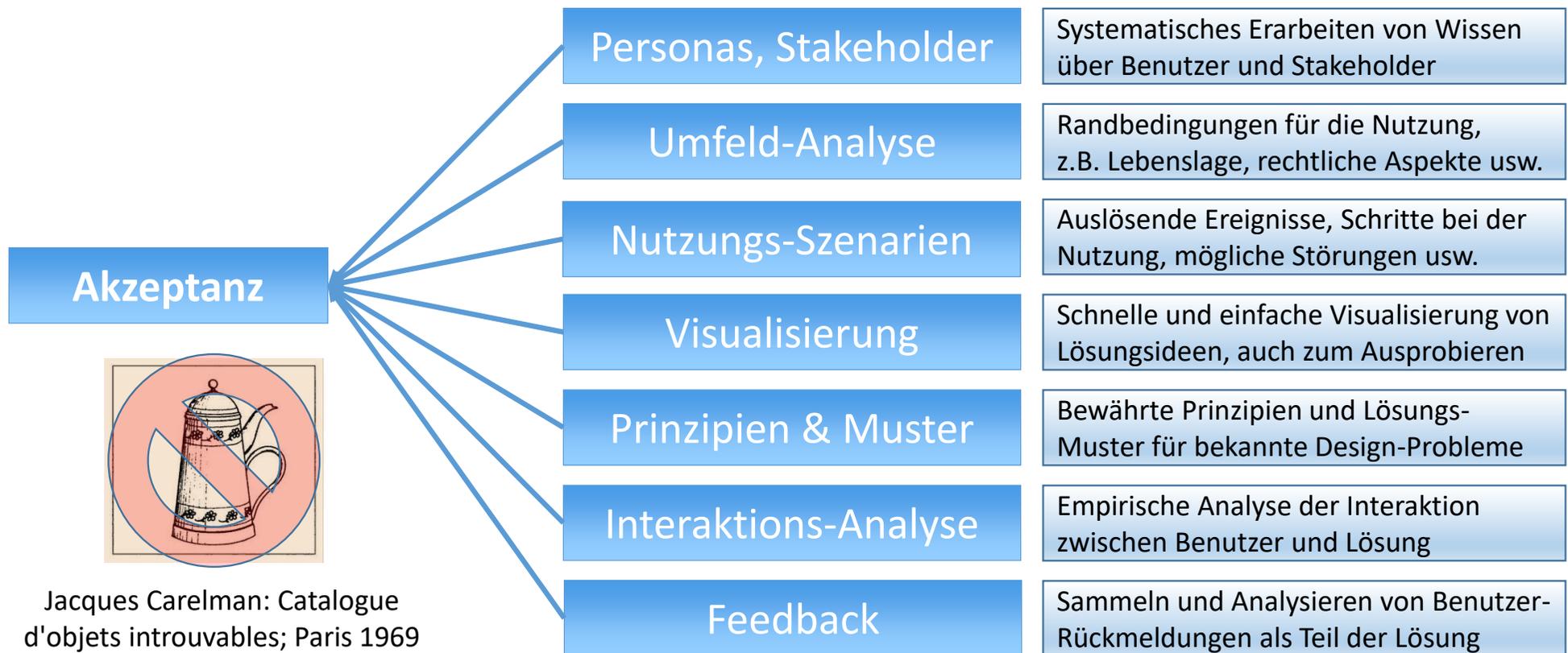
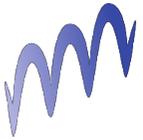


Nutzer-zentriertes Design geht von den Zielen und Bedürfnissen der Nutzer aus



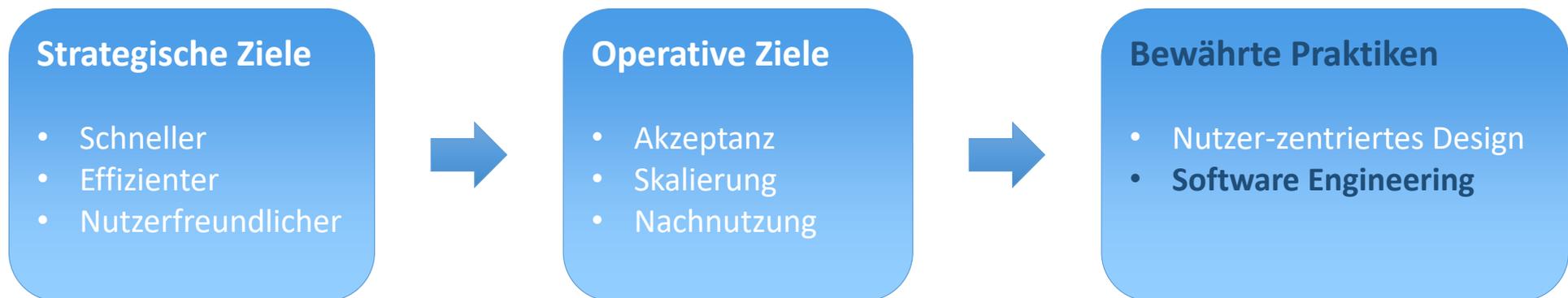
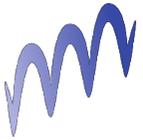
Angelehnt an Kim Goodwin: „Designing For the Digital Age“; Wiley 2009.

Nutzer-zentriertes Design verfügt über zahlreiche bewährte Praktiken

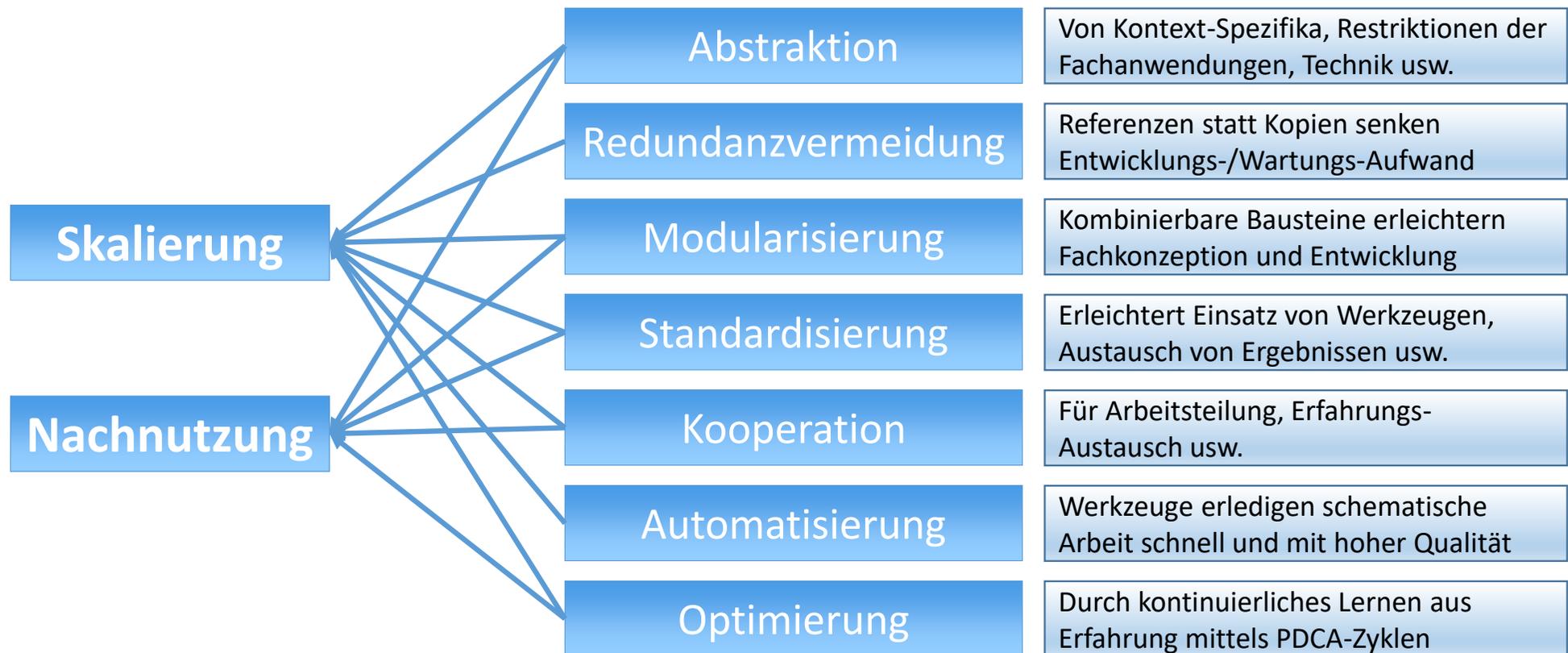
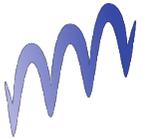


Jacques Carelman: Catalogue d'objets introuvables; Paris 1969

Software Engineering bietet weitere Praktiken zum Erreichen der operativen Ziele



Pragmatisches Software Engineering fördert Skalierung und Nachnutzung





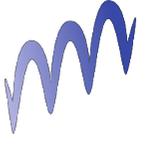
Wie? Maßnahmen

Inkrementelle Optimierung

Die Optimierung erfolgt in kurzen Inkrementen mit Feedback-Zyklen, ausgerichtet auf Risiko-Reduzierung und positives Kosten/Nutzen-Verhältnis

Kaban

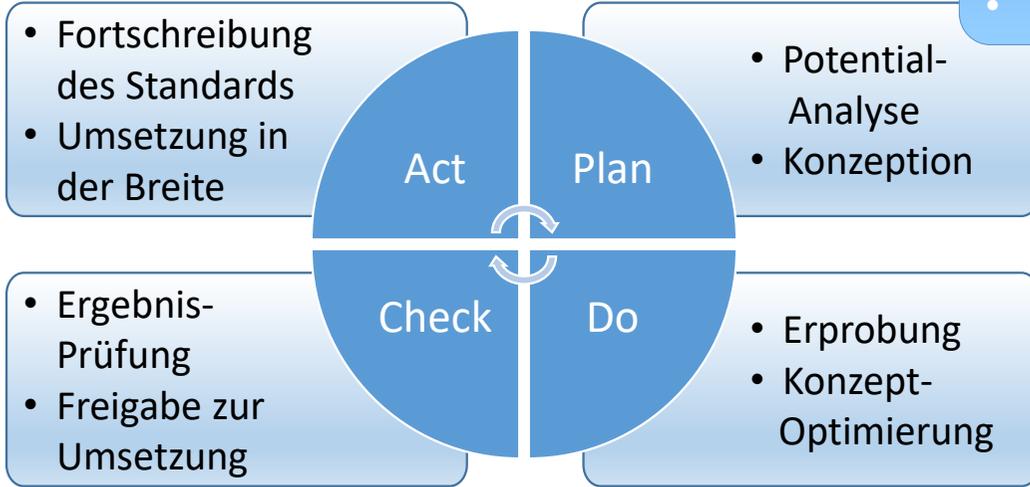
MDD



Die Optimierung erfolgt durch PDCA-Zyklen

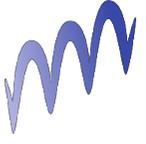
Optimiertes Vorgehen für

- Große Stückzahlen
- Lange Lebensdauer



Ad-hoc-Vorgehen für

- Geringe Stückzahlen
- Kurze Lebensdauer



Kurze PDCA-Zyklen reduzieren das Risiko und optimieren das Kosten-/Nutzen-Verhältnis

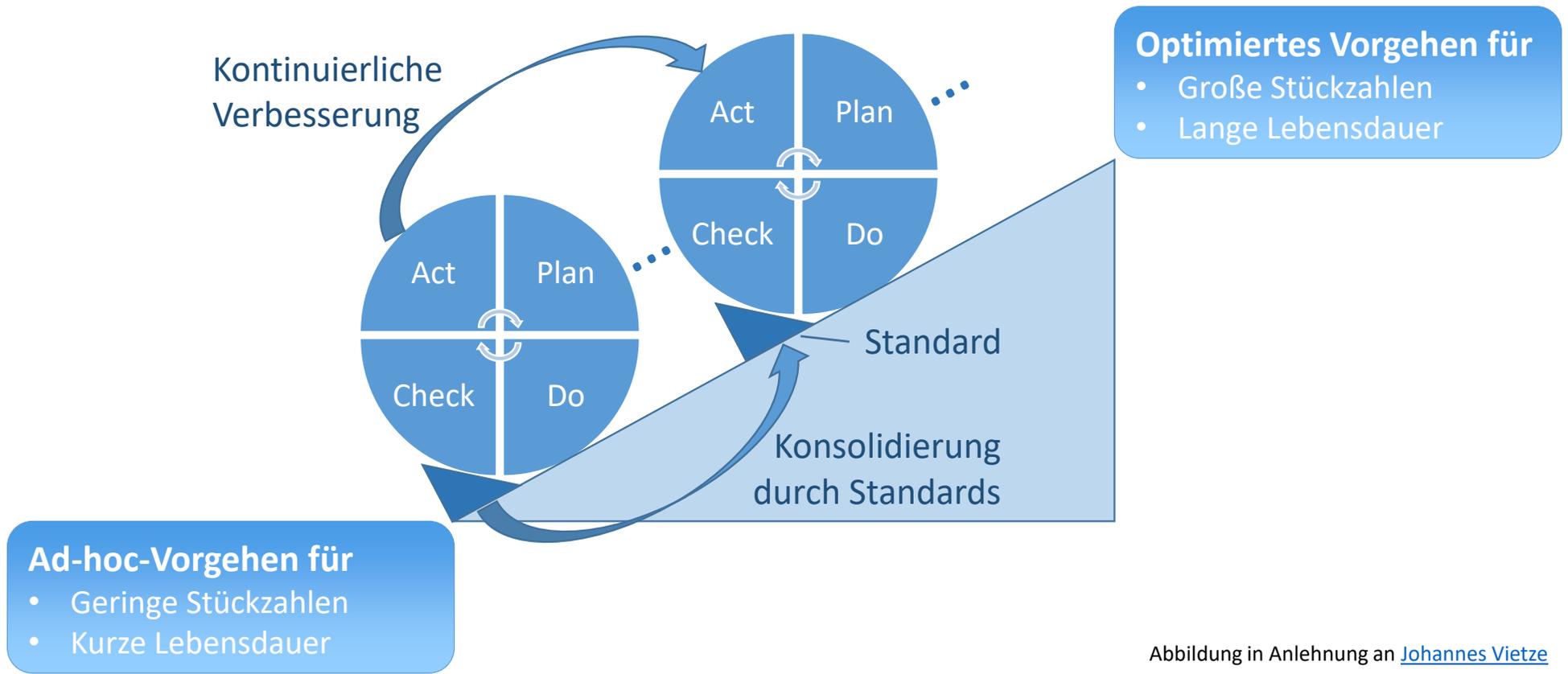


Abbildung in Anlehnung an [Johannes Vietze](#)



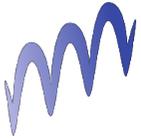
Optimierung Maßnahmen

Inkrementelles Vorgehen

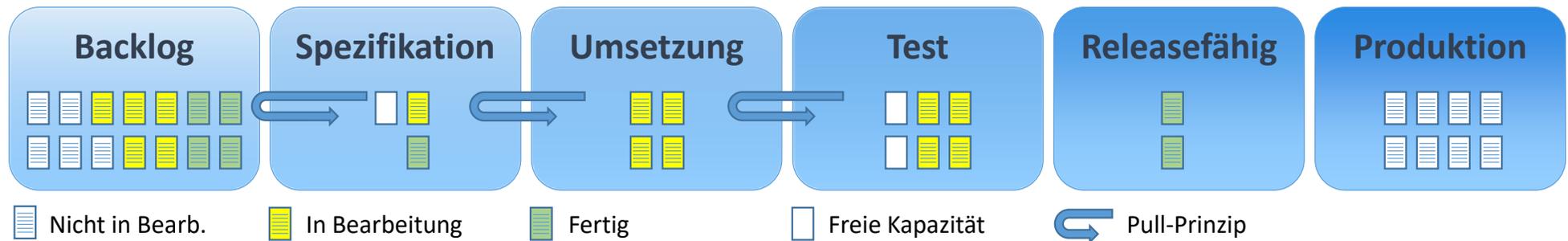
Kaban

Eine Kanban-ähnliche Projektsteuerung ermöglicht eine sichere Navigation in volatilem Umfeld, Kosten-Nutzen-Optimierung und kontinuierliche Verbesserung

MDD



Ein Kanban-ähnliches Vorgehen optimiert die Wertschöpfung der OZG-Umsetzung



Vorgehen	Nutzen
Kosten-Nutzen-Optimierung	Berücksichtigt verschiedene Kosten-/Nutzen-Arten, z.B. Verzugskosten (CoD)
Visualisierung der Arbeit	Ermöglicht Koordination aller Beteiligten und Gesamt-Optimierung
Möglichst späte Backlog-Bearbeitung	Nutzt neueste Erkenntnisse für „Fahren auf Sicht“ in volatilem Umfeld
Limitiertes Work-in-Progress	Erhöht Effizienz, Qualität, Termintreue
Steuerung des Arbeitsflusses	Beseitigt Engpässen, Wartezeiten, Hindernisse
Individuelle Prozess-Regeln	Berücksichtigen Organisations-Spezifika und machen Vorgehen explizit
Feedback-Mechanismen	Machen Mängel sofort sichtbar => vermeiden teure Folge-Probleme
Kontinuierliche Verbesserung	Nutzt Auslastungs-Lücken zur Investition in Gesamt-Optimierung



Optimierung Maßnahmen

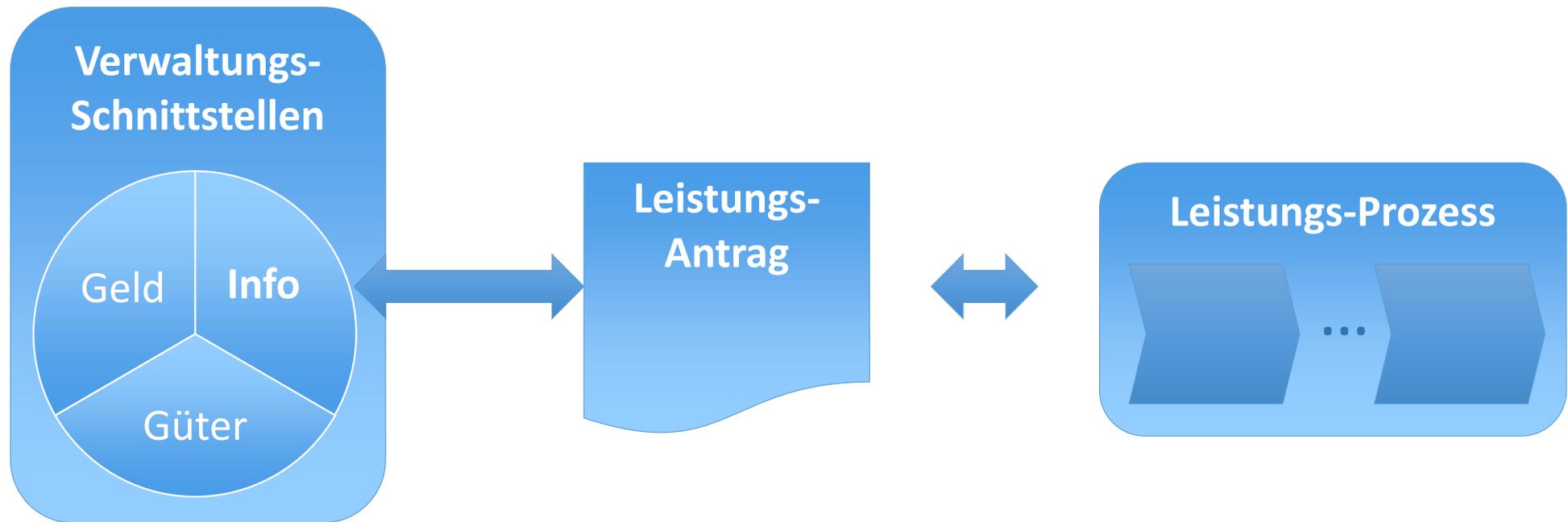
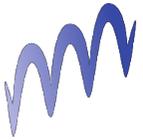
Inkrementelles Vorgehen

Kaban

MDD

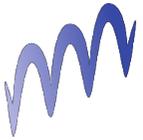
Model Driven Development (MDD) fördert Skalierung und Nachnutzung

Leistungen der Verwaltung werden über Informations-Schnittstellen beantragt



Neben Informations-Schnittstellen besitzen Verwaltungen auch Finanz-Schnittstellen (z.B. für Gebühren) und materielle Schnittstellen (z.B. für einen neuen Personalausweis). Um eine Leistung vollständig zu erbringen (Ende-zu-Ende), braucht man oft eine Kombination verschiedener Arten von Verwaltungs-Schnittstellen.

Informations-Schnittstellen sind relativ stabil im Bezug auf die benötigten Daten



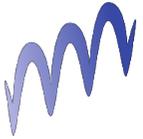
Antrags-Spezifikation

- Fachliches Datenmodell
- Struktur und Inhalt
- Kontexte
- Regeln
- Aktionen

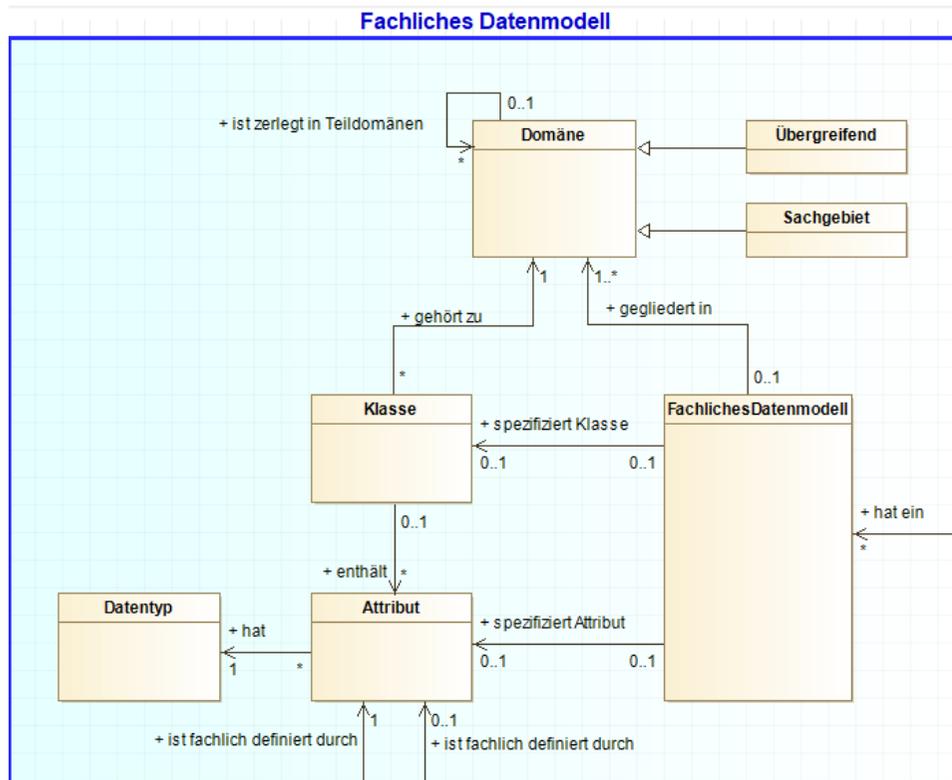
Beispiel Wohnsitz-Ummeldung:
Daten ähnlich wie vor 100 Jahren



Quellen: Wikimedia Commons (Foto), [SZ](#) (Karte)

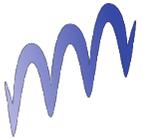


Für die Fachliche Datenmodellierung hat sich UML als Standard etabliert



- Modelle werden durch Domänen in handliche Sachgebiete gegliedert
 - Z.B. Meldewesen, Personenstandswesen usw.
- UML-Nutzungskonzept
 - Legt diejenige Untermenge von UML fest, die man tatsächlich benötigt
- UML-Modellierungs-Werkzeuge
 - Open Source, z.B. Modelio
 - Proprietär, z.B. MID Innovator, Sparx Enterprise Architect

Struktur und Inhalt eines Leistungs-Antrags werden in rein fachlicher Form spezifiziert



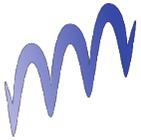
Antrags-Spezifikation

- Fachliches Datenmodell
- **Struktur und Inhalt**
- Kontexte
- Regeln
- Aktionen

- Reines Verwaltungs-Wissen
 - Unabhängig von Formular-Server, PDF-Generator, Chatbot, usw.
- Fachliche Bausteine
 - Antrag¹
 - Kapitel: Personendaten,...
 - Infobox: Datenschutzerklärung, ...
 - Feld: Familienname, ...
 - Feldgruppe: Adresse, Fahrzeugdaten, ...²

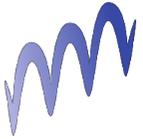
¹ Geschachtelte Anträge, z.B. Steuererklärung (Mantelbogen, Anlage S, SO, ...)

² Kardinalität 1: z.B. nur eine Adresse; Kardinalität n: z.B. mehrere Fahrzeuge



Die Struktur eines Leistungs-Antrags besteht aus einer Hierarchie von Bausteinen





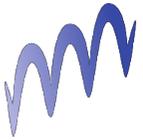
Fachliche Datentypen spezifizieren

Fachwissen kompakt und nachnutzbar

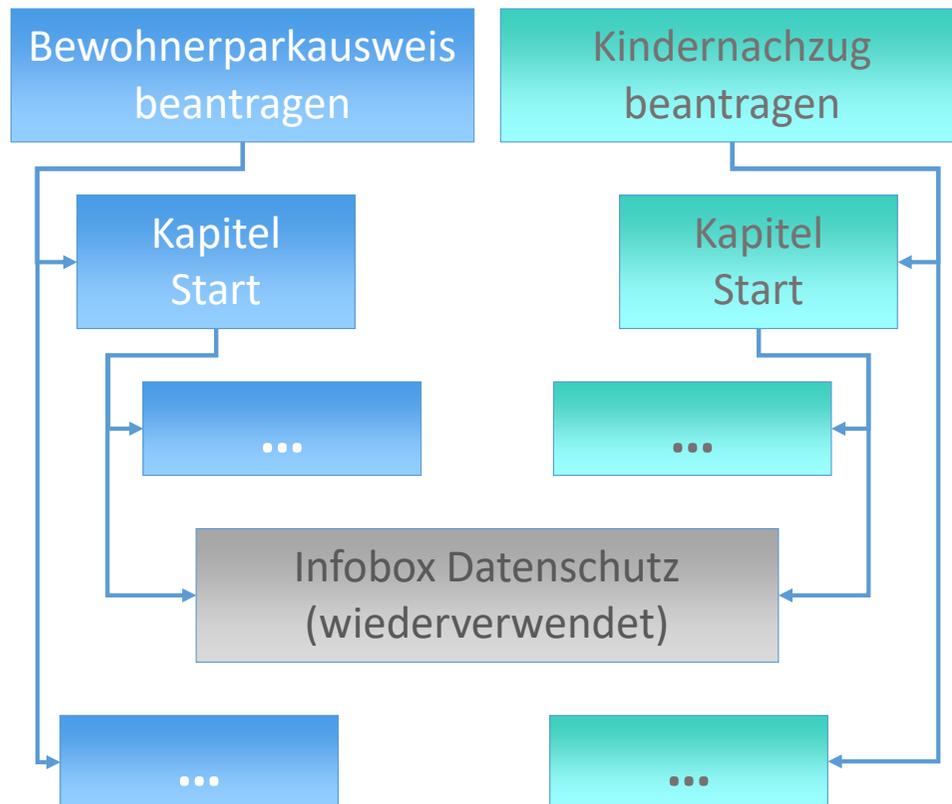
Fachliches Datenmodell		Datentyp	Typ	Max. Länge	Zeichenart	Sonderzeichen	Validierung
▼	NatürlichePerson	Familienname	Text	120	U, S	-	Eingabe <>' ' und keine Zahl (1234567890)
	Is a Person (from Personen)						
	a: +Vorname : Vorname						
	a: +Familienname Familienname						
	a: +Titel : Titel [0..1]						

Fachliche Datentypen wie Familienname, Geldbetrag, Postleitzahl, IBAN usw. spezifizieren

- Welche Werte für Attribute des betreffenden Datentyps erlaubt sind
- Welche Operationen man mit Werten eines Datentyps durchführen darf, z.B.
 - Mit Geldbeträgen darf man rechnen und erhält wieder Geldbeträge
 - Berechnungen mit Postleitzahlen sind fachlich sinnlos und daher verboten



Bausteine kann man extrahieren und in den Antrags-Baukasten aufnehmen zur Nachnutzung

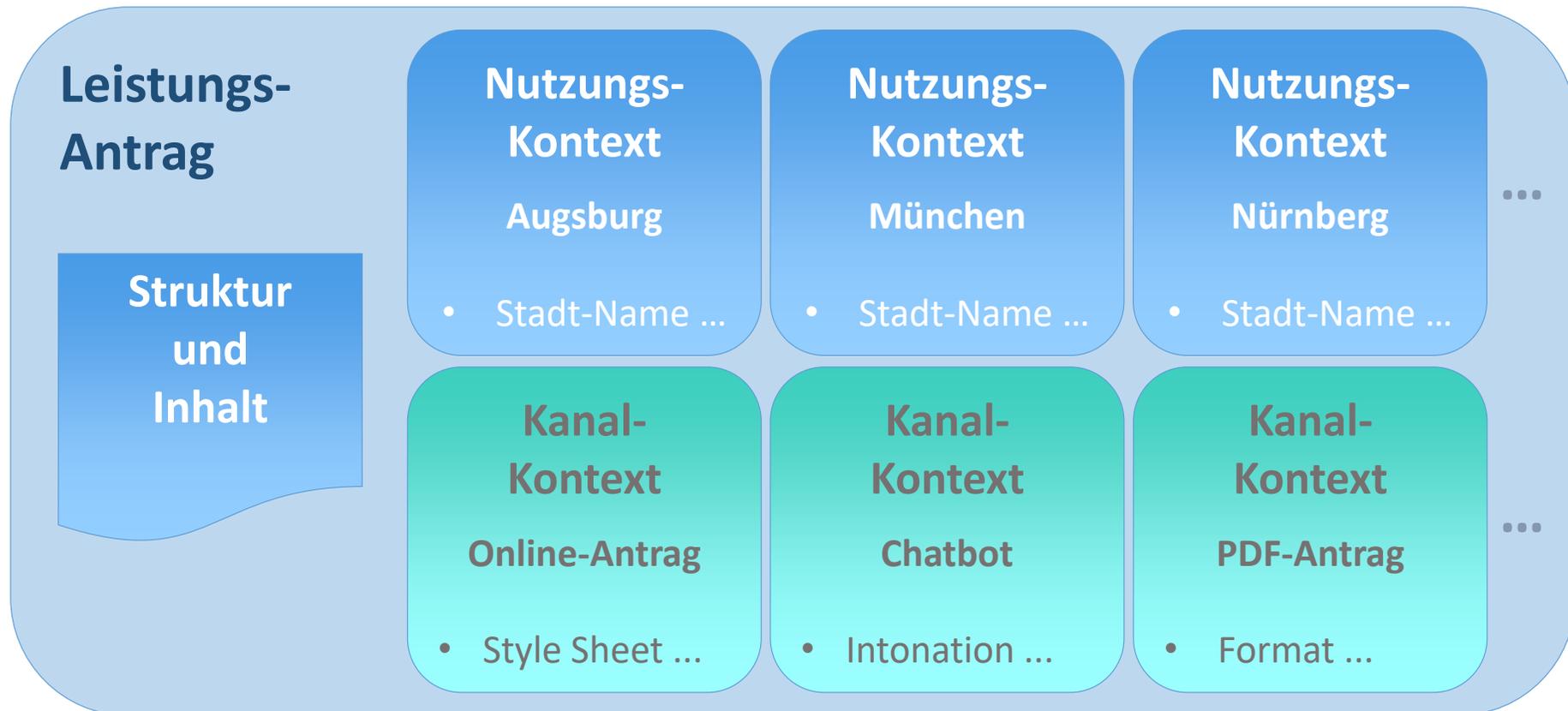
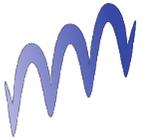


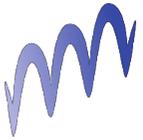
- Ein neuer Baustein entsteht im Zuge einer Antrags-Spezifikation
 - Beispiel: Datenschutzerklärung für Bewohnerparkausweis
- Ein neuer Antrag (z.B. Kindernachzug) braucht fachlich die selbe Datenschutzerklärung
 - Der Baustein wird aus der initialen Antrags-Spezifikation extrahiert
 - In beide Spezifikationen fügt man eine Referenz auf den Baustein ein

Eine Antrags-Spezifikation wird durch Kontexte parametrisiert

Antrags-Spezifikation

- Fachliches Datenmodell
- Struktur und Inhalt
- Kontexte
- Regeln
- Aktionen





Kontexte fördern die Nachnutzung

Leistungs-Antrag

Bewohnerparkausweis beantragen

„Sie möchten einen Bewohnerparkausweis ... beantragen? ... Weitere Informationen finden Sie unter <LinkPlatzhalter Parkraummanagement>.“

Nutzungs-Kontext: München

LinkPlatzhalter: Parkraummanagement

- Text: Parkraummanagement
- URL:
<https://www.muenchen.de/rathaus/Stadtverwaltung/Kreisverwaltungreferat/Verkehr/Parkraummanagement.html>

• Nutzungs-Kontexte

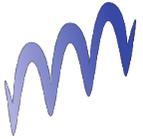
- Platzhalter festlegen
 - Z.B. Email-Adressen, Links auf weiterführende Informationen, usw.
- Bausteinen-Sichtbarkeit festlegen
 - Z.B. Felder ausblenden
- Feldlängen kürzen bis zur Anpassung des betreffenden Fachverfahrens*

• Kanal-Kontexte

- Legen Kanal-spezifische Merkmale eines Antrags fest
- Z.B. bei Online-Anträgen ein Style Sheet für die Web-Darstellung usw.

* Z.B.: Änderung Melderegister-Gesetz 2019 – Straße/Hausnr 62 Zeichen statt 50

Fachliche Regeln dienen zur Plausibilisierung von Eingaben und zur Steuerung des Ablaufs



Antrags-Spezifikation

- Fachliches Datenmodell
- Struktur und Inhalt
- Kontexte
- **Regeln**
- Aktionen

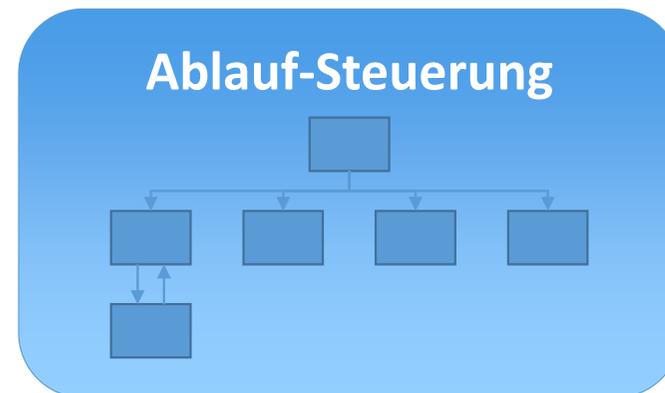


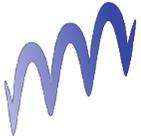
Eingabe-Plausibilisierung

Bauart des Fahrzeugs: *

Bitte ausfüllen.

- Ich möchte einen PKW oder ein Motorrad in den Parkausweis eintragen.
- Ich möchte ein aufgrund der Bauart vorrangig zur gewerblichen Nutzung vorgesehenes Fahrzeug in den Parkausweis eintragen.





Viele fachliche Regeln kann man deklarativ mit Entscheidungs-Tabellen nach DMN¹ spezifizieren

ParkausweisErfordertFührerschein		
U	HatFührerschein	Folgeaktion
1	Ja	Keine Aktion
2	Nein	AntragBeenden Fehlermeldung: „Wenn Sie keinen Führerschein besitzen und trotzdem einen Bewohnerparkausweis beantragen möchten, wenden Sie sich bitte per E-Mail an <LinkPlatzhalter EmailParkausweise> “

Geeignet für z.B.

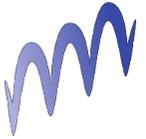
- Steuerung von Benutzer-Interaktionen beim Beantragen von Leistungen (siehe oben)
- Plausibilisierung von Benutzer-Eingaben

¹ Decision Model and Notation: Standard für maschinenlesbare Regeln

Nicht geeignet für z.B.

- Spezifikation von algorithmischen Regeln, z.B. zur Kerosin-Umlage auf Flugstrecken
- Spezifikation von Syntax-Regeln²

² Viele Syntax-Regeln kann man am einfachsten mit Regulären Ausdrücken spezifizieren



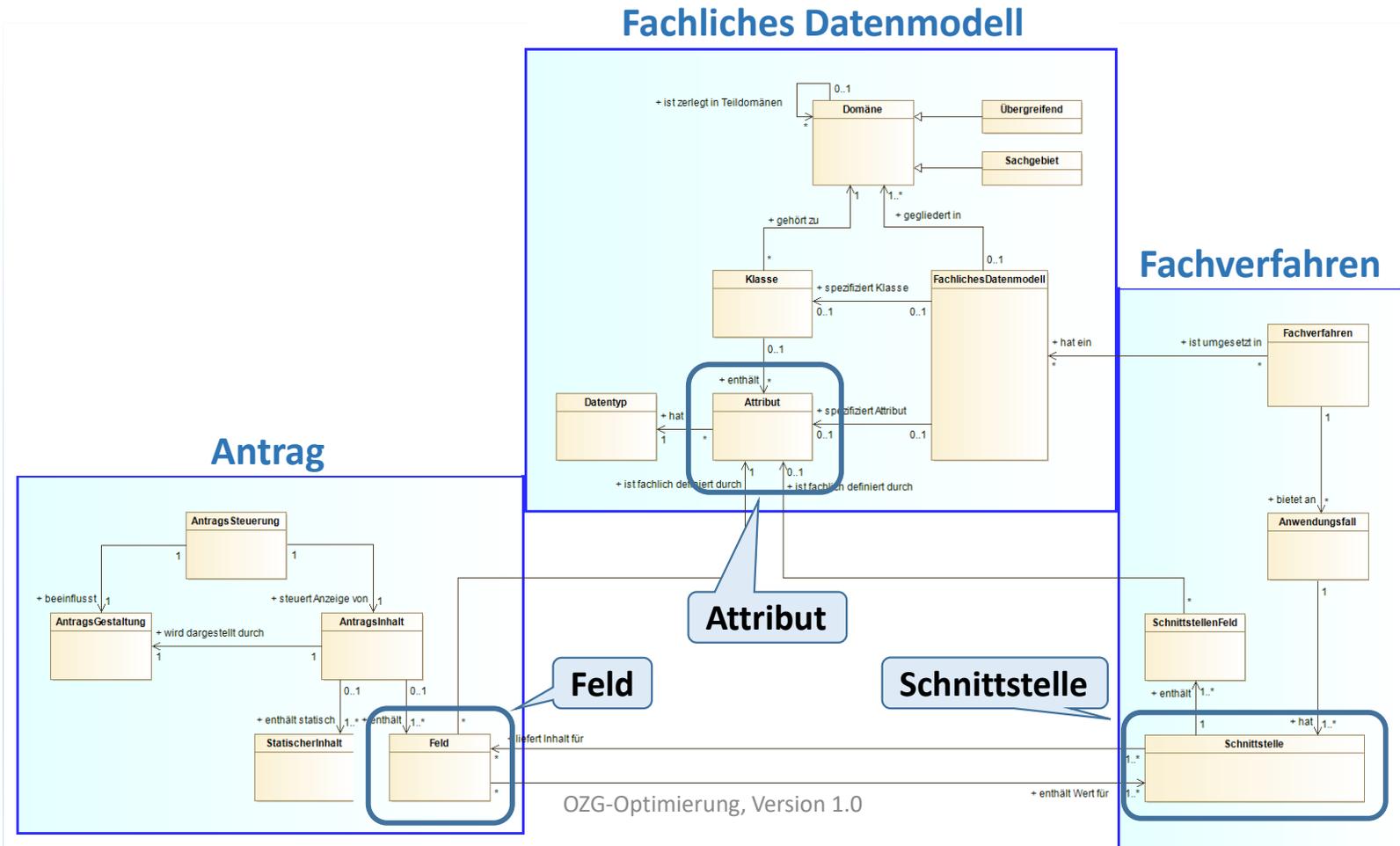
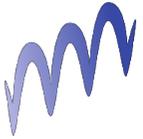
Beim Beantragen von Leistungen können Aktionen ausgelöst werden

Antrags-Spezifikation

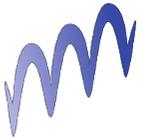
- Fachliches Datenmodell
- Struktur und Inhalt
- Kontexte
- Regeln
- **Aktionen**

- **Dialog-Aktionen**
 - Dienen zur Steuerung des Dialogs
 - Beispiele
 - AntragBeenden (siehe vorige Folie)
 - Ausblenden unnötiger Bausteine
- **Fach-Aktionen**
 - Lösen Anwendungsfälle in Fachanwendungen aus
 - Beispiele
 - Kindernachzug beantragen
 - Bewohnerparkausweis beantragen

Eine Fach-Aktion beliefert eine Fachverfahrens-Schnittstelle mit Feldern aus dem Leistungs-Antrag



Einen Leistungs-Antrag kann man in großen Teilen maschinenlesbar spezifizieren

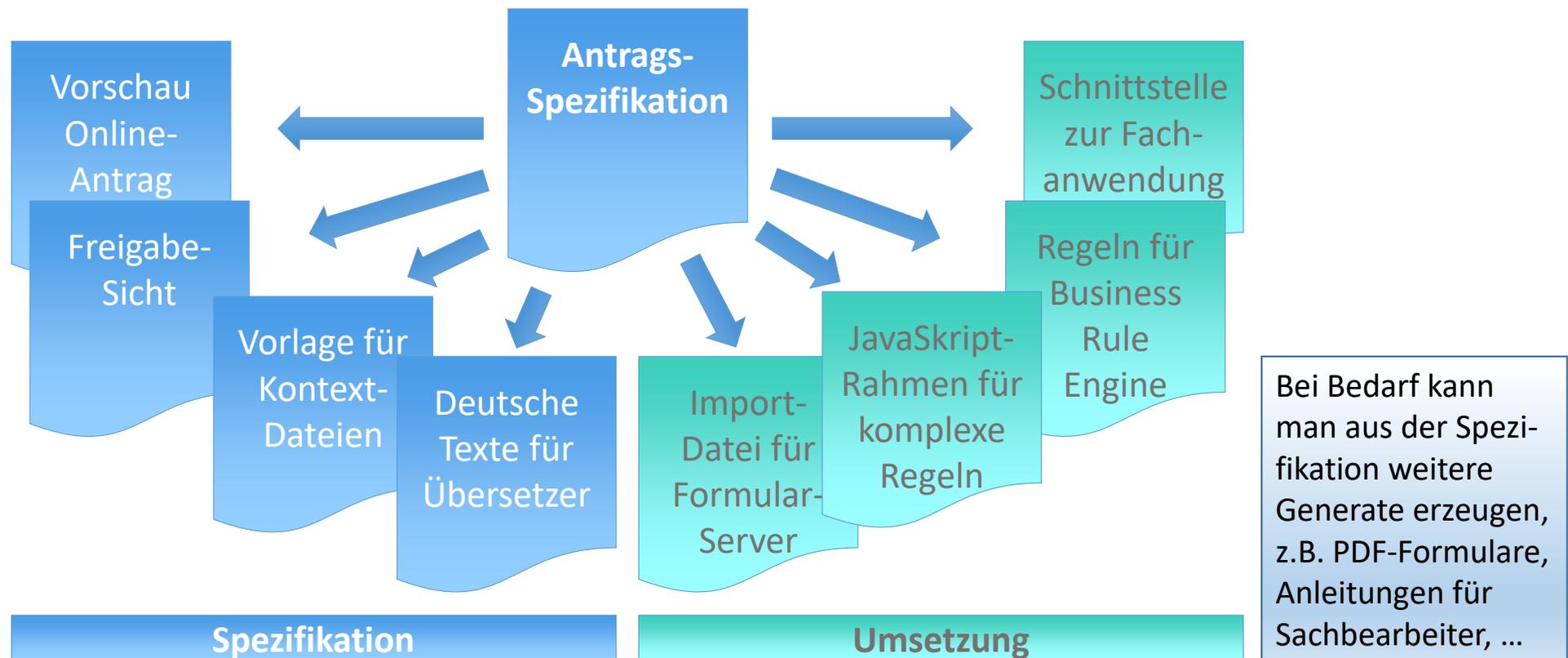
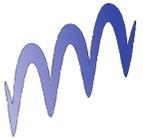


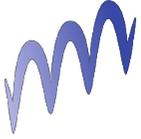
Antrags-Spezifikation

- Fachliches Datenmodell
- Struktur und Inhalt
- Kontexte
- Regeln
- Aktionen

Antrags-Spezifikation	Standard
Fachliches Datenmodell	UML
Struktur und Inhalt	XML / XSD
Kontexte	XML / XSD
Regeln	DMN, RegEx
Aktionen	UML / XÖV

Maschinenlesbare Antrags-Spezifikationen kann man relativ einfach transformieren





Aus der Antrags-Spezifikation kann man leicht eine Online-Vorschau generieren

* Vergrößerung siehe nächste Folie

Antrags-Spezifikation (XML)*

XML Notepad - C:\Users\am\Documents\Beruf\LHM\Antrags-Spezifikation\Beispiele\bewohnerparkausweis_beartragen.xml

Tree View: XSL Output

- antrag
 - kapitel
 - uberschrift
 - name
 - istischer
 - erleitung
 - feld
 - name
 - beschreibung
 - istpflichtig
 - istfeldart
 - definiertdurch
 - feld
 - name
 - beschreibung
 - istpflichtig
 - istfeldart
 - definiertdurch
 - feld
 - name
 - beschreibung
 - istpflichtig
 - istfeldart
 - definiertdurch
 - regel
 - name
 - beschreibung
 - feldgruppe
 - name
 - beschreibung
 - feld
 - name
 - istpflichtig
 - istfeldart
 - definiertdurch
 - platzhalter
 - regel
 - name
 - beschreibung
 - feld
 - name
 - istpflichtig
 - istfeldart
 - definiertdurch
 - platzhalter
 - feld
 - name
 - istpflichtig
 - istfeldart
 - definiertdurch
 - platzhalter
 - feld
 - name
 - istpflichtig
 - istfeldart
 - definiertdurch
 - platzhalter

Bewohnerparkausweis beantragen

Landeshauptstadt München

Quick-Check

* Pflichtfelder

Um sicher zu stellen, dass das vorliegende Online-Formular Ihre Bedürfnisse deckt, bitten wir Sie folgende Fragen zu beantworten.

Besitzen Sie einen gültigen Führerschein? *

Ja
 Nein

Haben Sie bereits einen gültigen Parkausweis? *

Ja
 Nein

Verfügen Sie über einen anderen Stellplatz auf Privatflächen oder in einer Garage im Parklizenzbereich oder in einem angrenzenden Gebiet? *

Ja
 Nein

Zwecks einer etwaigen Überprüfung Ihrer Angaben benötigen wir die Daten Ihres Wohnung Vermieters / Ihrer Wohnungvermieterin. Wenn sich die Wohnung in Ihrem Eigentum befindet, geben Sie bitte Ihre Daten ein.

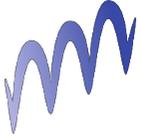
Name: *

Straße und Hausnummer: *

Postleitzahl: *

Wohnort: *

Online-Vorschau (HTML/CSS)



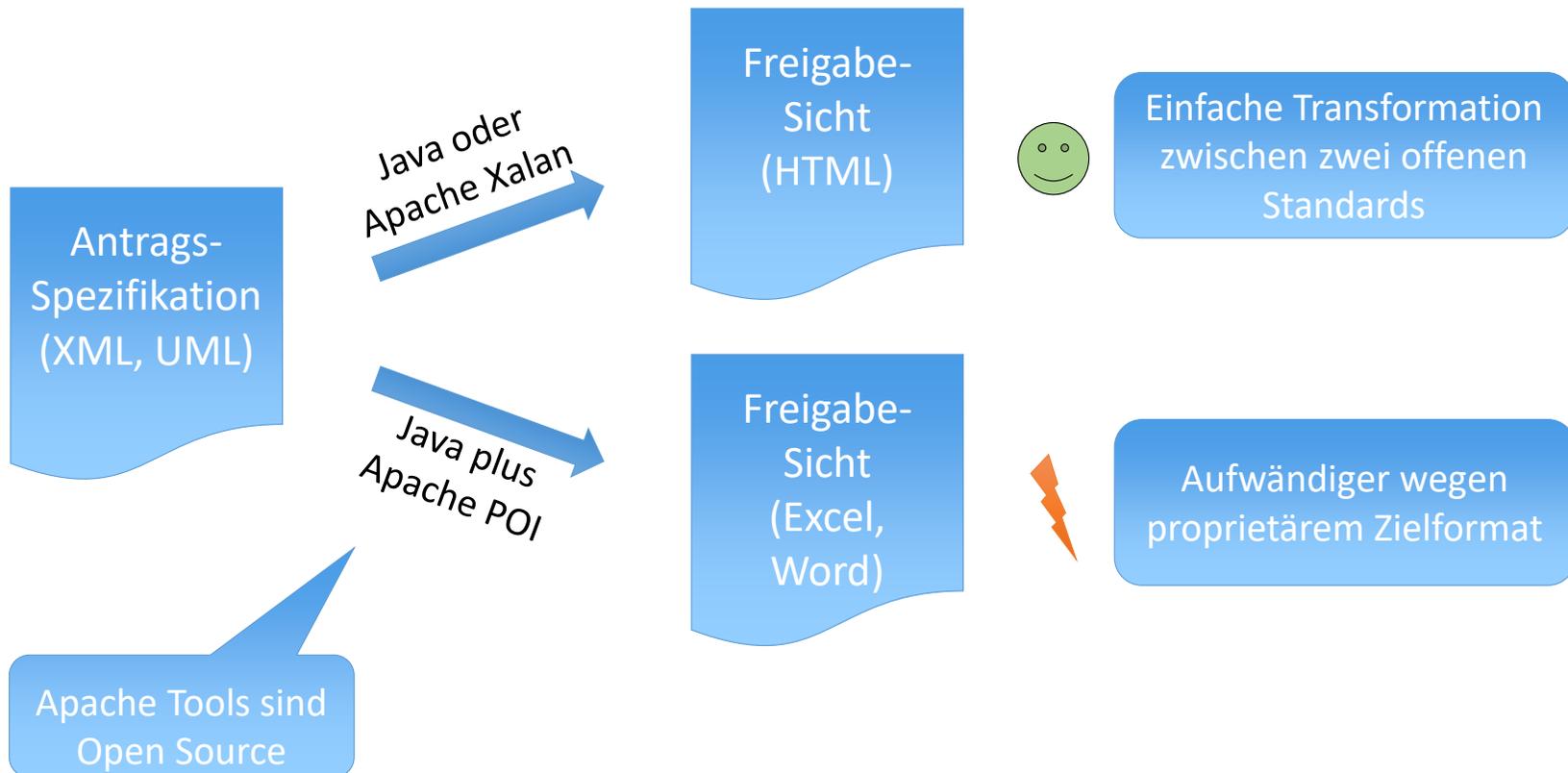
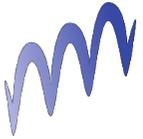
Eine XML-Antrags-Spezifikation kann man komfortabel ohne technisches Wissen editieren

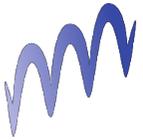
XML Notepad* ist Open Source

The screenshot shows the XML Notepad application with a tree view on the left and a user interface on the right. The tree view shows a hierarchy of elements: `xml` (root), `antrag` (child), `kapitel` (child of `antrag`), and `feld` (child of `kapitel`). The `feld` elements are further categorized into `regel` and `feld` sub-elements. The user interface on the right shows a form with various input fields and labels, such as "Wohnungseigentümer", "Name", "Straße und Hausnummer", "HausnummerAngegeben", "Postleitzahl", and "Wohnort". A blue watermark "Beispiel ohne Anspruch auf Vollständigkeit" is overlaid on the image.

* Man kann einen beliebigen XML-Editor für die Spezifikation von Leistungs-Anträgen verwenden

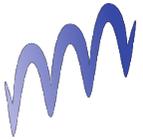
Eine Antrags-Spezifikation kann man sehr einfach zu einer redundanten Freigabe-Sicht expandieren



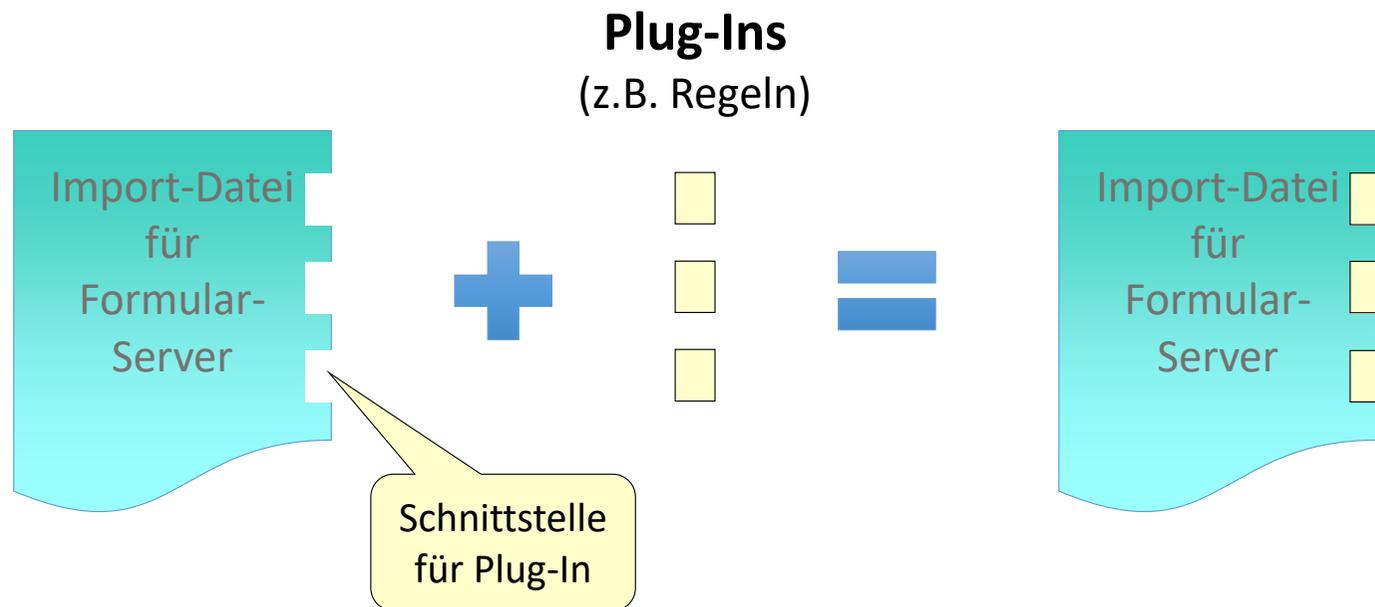


Eine maschinen-lesbare Spezifikation bietet die Chance, Produkt-Abhängigkeiten zu reduzieren





Dank Plug-Ins können die Ergebnisse der Transformation unverändert bleiben



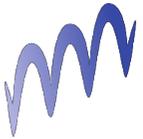
Vorausgesetzt, der Transformator gilt nach Tests als korrekt: Plug-Ins verändern die Import-Datei nicht. Daher muss nach Plug-In-Integration die Import-Datei selbst nicht getestet werden, nur ihre Integration mit Plug-Ins.



Überblick

- Zusammenfassung
- Analyse des aktuellen Stands
- Optimierung
 - Methodische Grundlagen
 - Maßnahmen
- **Nächste Schritte**
- Kontakt

Ein schlanker Prozess zur effizienten Optimierung integriert Best Practices und Nutzer-Sichten



Quick-Check

- Risiken ermitteln durch Abgleich mit bewährten Praktiken
- Optimierung-Potential identifizieren

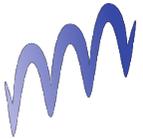
Priorisierung

- Optimierung-Potential mit Stakeholdern abstimmen
- Optimierung-Potential mit Stakeholdern priorisieren

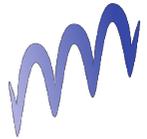
Umsetzung

- Optimierungs-Maßnahmen aus Potential ableiten
- Umsetzung der Maßnahmen starten (PDCA-Zyklen)

Ein Quick-Check anhand bewährter Praktiken ermittelt Risiken und Optimierungs-Potential

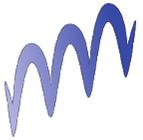


Eine Liste potentieller Optimierungs- Maßnahmen erleichtert die Priorisierung

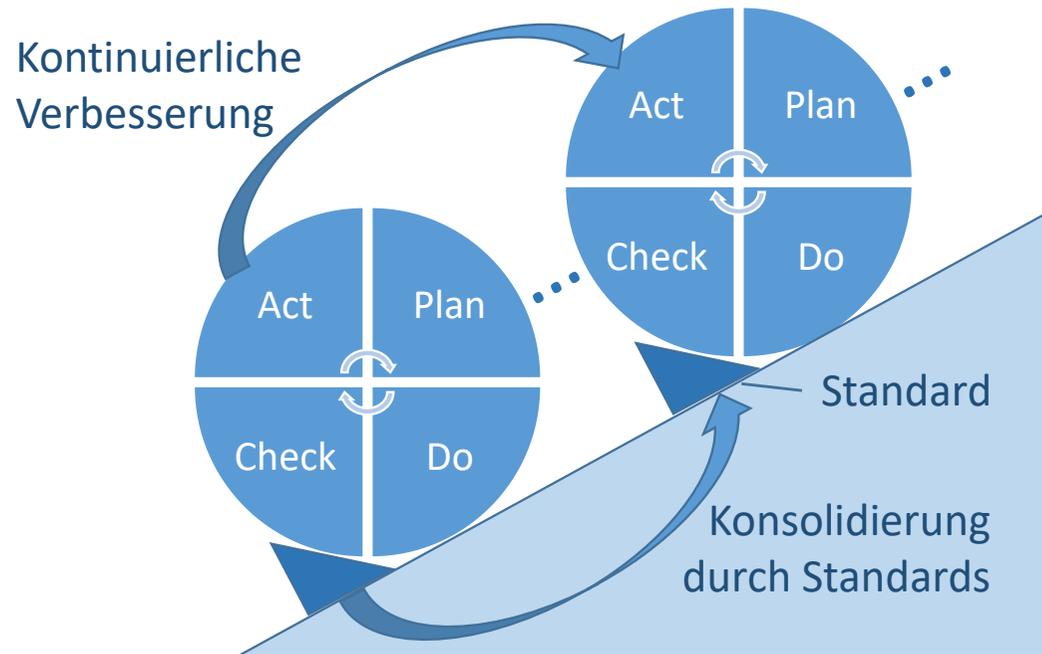


#	Maßnahmen	Nutzen	Abh.	N ¹	K ²
1	UML-Datenmodell	Fachwissen dokumentiert => MDD			
2	XML-Antrags-Spezifikation	Produkt-Unabhängigkeit	1		
3	XML-Kontexte	Entkopplung von Nutzungs- und Kanal-Spezifika	1, 2		
4	DMN-Regeln	Regeln als ausführbare Spezifikation	1		
5	UML/XÖV-Aktionen	Standards fördern Werkzeug-Einsatz	1		
6	Vorschau-Generierung	Iterative Spezifikation mit kurzen Feedback-Zyklen	1, 2		
7	Formular-Server-Import	Weniger Tests notwendig; No-Code-Wartung durch Fachbereich ³	1, 2, 3		

¹ Nutzen ² Kosten: wenige Personentage (PT) wenige PW wenige PM ³ am besten kombiniert mit CI/CD



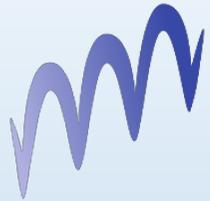
Die Umsetzung nutzt freie Kapazitäten für effiziente PDCA-Zyklen



Freie Kapazitäten sind ein Nebeneffekt von Pull-Prinzip und WiP-Limits – und essentiell für die Gesamt-Effizienz. Man nutzt sie zur Optimierung per PDCA.



Kontakt



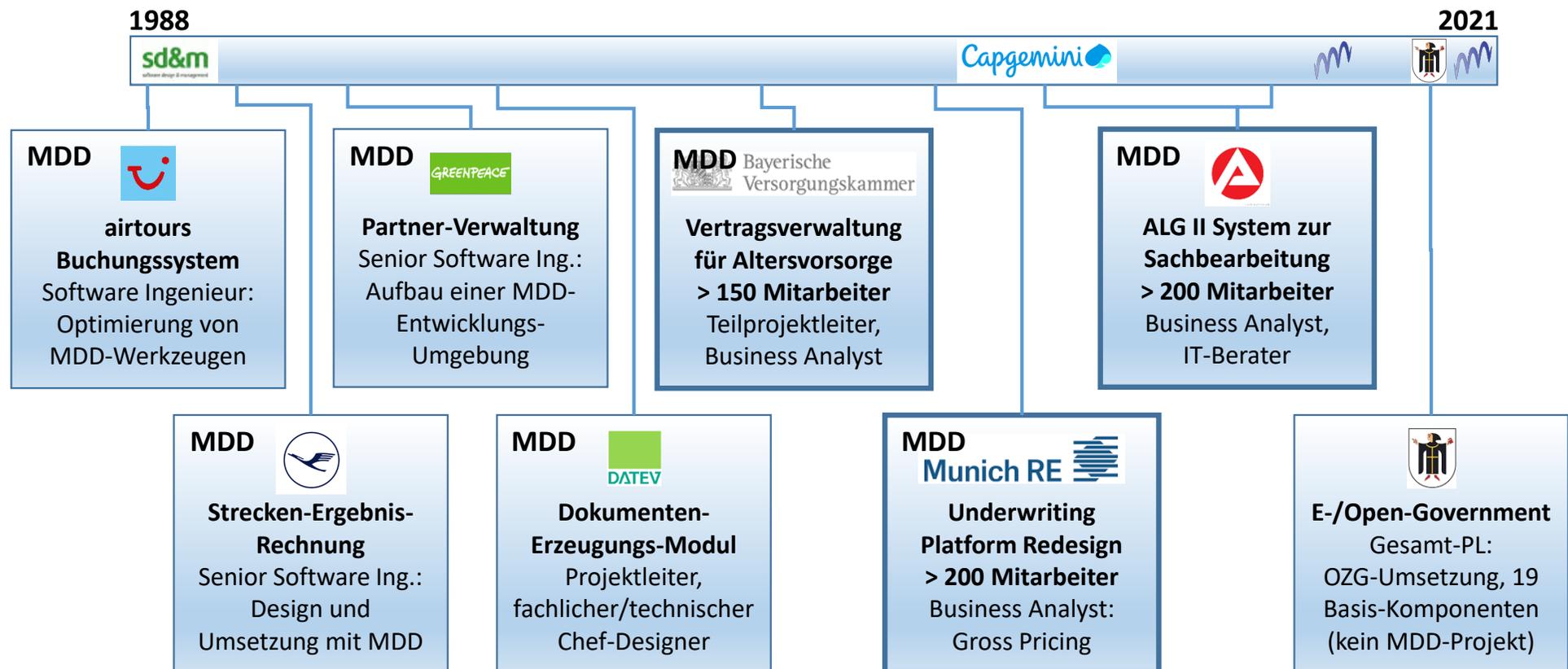
Andreas Mieth

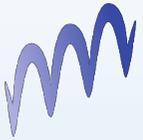
Freiberuflicher IT-Berater
für Projektmanagement
und Business Analyse



Telefon: [0160 9860 2234](tel:016098602234)
Email: andreas.mieth@mailbox.org
Homepage: www.amieth.de

Erfahrungen mit OZG, Öffentlichem Dienst, Projektleitung, Business Analyse, MDD (Auszug)





Zitate aus Zeugnissen

Herr Mieth verfügt über ein hervorragendes und auch in Randbereichen sehr tiefgehendes Fachwissen, welches er in unser Unternehmen stets in höchst gewinnbringender Weise einbringt.

Herr Mieth ist flexibel einsetzbar, er verfügt über große fachliche Kompetenz und hat neue, komplexe Themengebiete eigenständig erarbeitet und anderen zugänglich gemacht. Die Qualität seiner Arbeitsergebnisse ist beispielhaft. Herr Mieth ist extrem belastbar und zeigt sehr hohen Arbeitseinsatz, dabei bleibt er stets ausgeglichen und freundlich. Er verfügt über eine hohe, allgemeine Lösungskompetenz und eine ausgeprägte analytische Begabung, die er mehrfach in fachliche Modelle und Arbeitsorganisationen umgesetzt hat.

Er arbeitet stets zügig, umsichtig, sorgfältig und genau. Herr Mieth überzeugt stets durch seine außerordentliche Verlässlichkeit.

Aufgrund seiner sehr guten Auffassungsgabe ist er jederzeit in der Lage, auch schwierige Situationen sofort zutreffend zu erfassen und schnell sehr gute Lösungen zu finden. Herr Mieth zeigt jederzeit hohe Eigeninitiative und identifiziert sich immer voll mit seinen Aufgaben und unserem Unternehmen, wobei er auch durch seine sehr große Einsatzfreude überzeugt. Auch in Situationen mit großem Arbeitsaufkommen erweist er sich immer als in hohem Maße belastbar.

Er wird wegen seines freundlichen und ausgeglichenen Wesens allseits sehr geschätzt. Er ist immer hilfsbereit, zuvorkommend und stellt, falls erforderlich, auch persönliche Interessen zurück. Sein Verhalten zu Vorgesetzten, Kolleginnen und Kollegen sowie Kundinnen und Kunden ist ausnahmslos einwandfrei.