

OZG-Umsetzung

Wie kann eine Organisation
ihr Vorgehen schnell optimieren?
(Kurzfassung)

Andreas Mieth, 14.02.2021

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

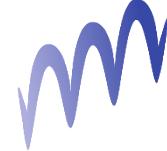


- Warum?
 - Die OZG-Umsetzung ist ein Großprojekt: viele Ergebnisse, Beteiligte, Versionen, Jahre
 - Ein Ad-hoc-Vorgehen¹ ist riskant, vor allem im Bezug auf Skalierung und Nachnutzung
- Wie? – Methodische Grundlagen
 - Die Optimierung basiert auf nutzer-zentriertem Design und Software Engineering
- 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 volatillem Umfeld, Kosten-Nutzen-Optimierung und kontinuierliche Verbesserung
 - Durch Model Driven Development (MDD) fördert sie Skalierung und Nachnutzung

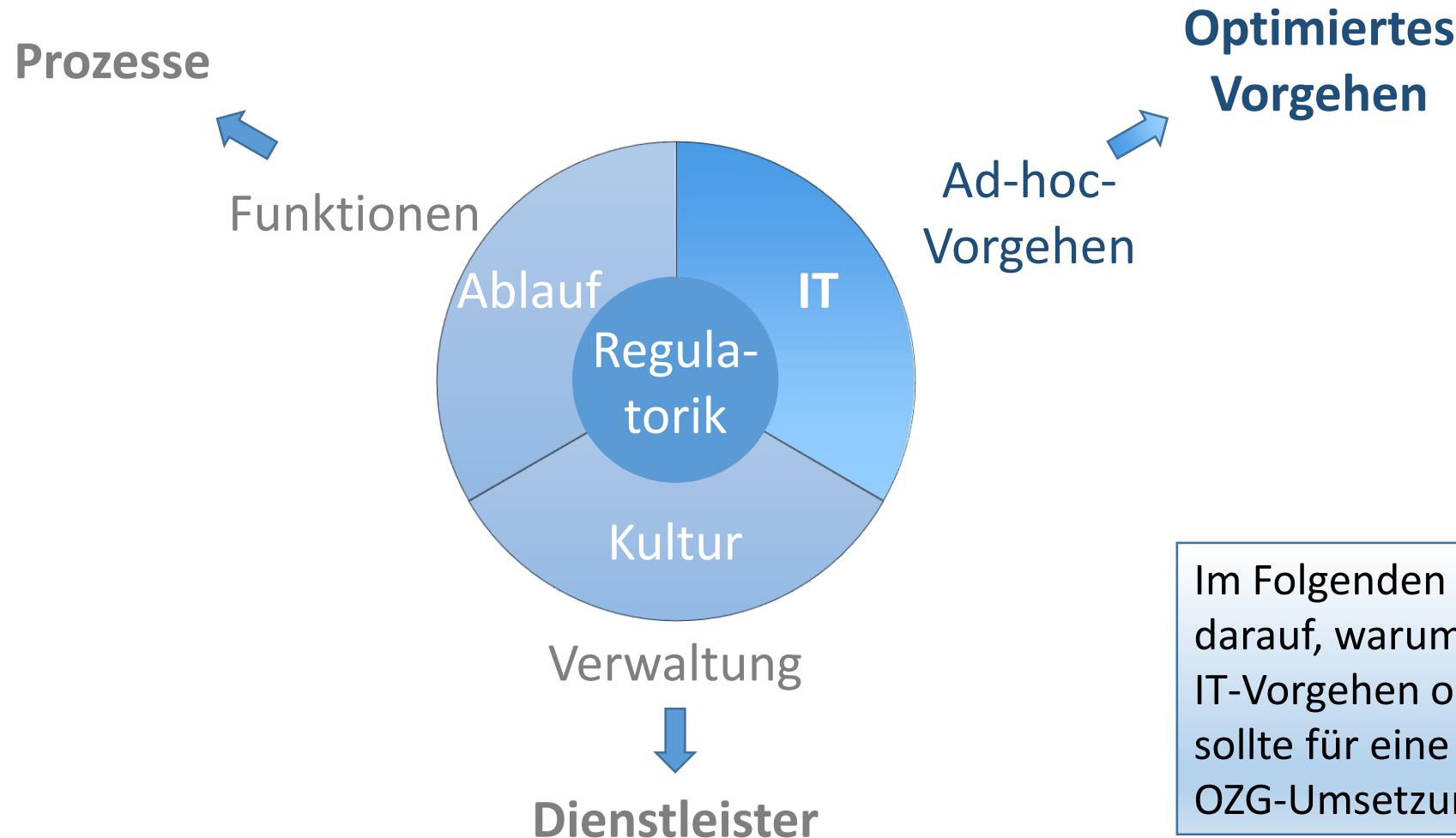
¹ Gewohntes Vorgehen ungeachtet der Größe und Komplexität des Vorhabens

Bund
oder
Land
1845 LeiKa-
Leistungen

Land
oder
Kommune
4522 LeiKa-
Leistungen

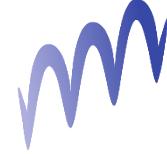


Das OZG erfordert einen komplexen Wandel



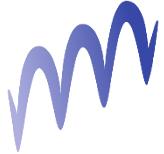
Im Folgenden liegt der Fokus darauf, warum und wie das IT-Vorgehen optimiert werden sollte für eine erfolgreiche OZG-Umsetzung

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

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



Anbieter-zentriert: Such-Maschine



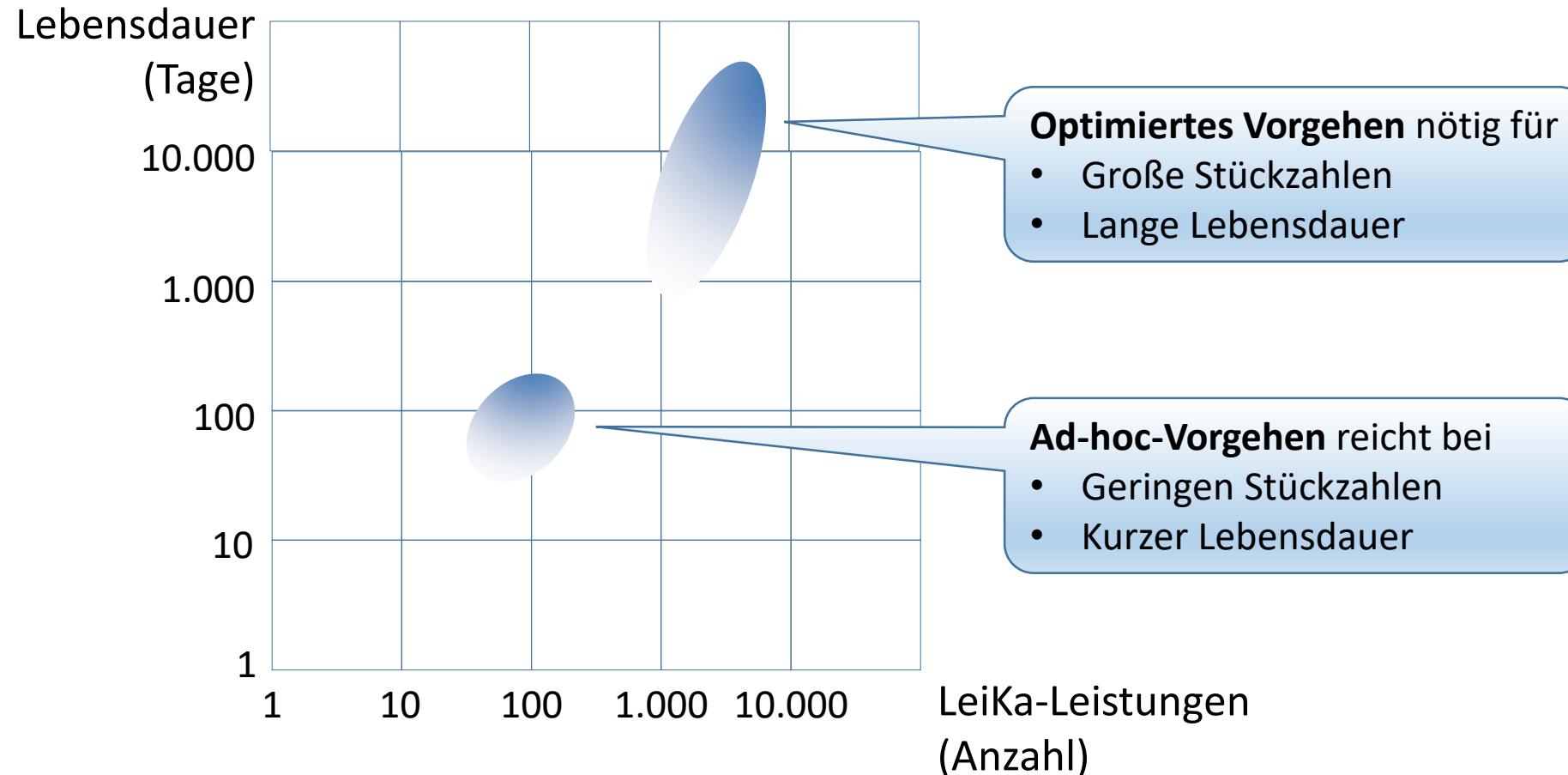
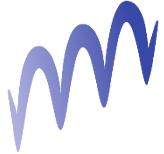
Nutzer-zentriert: Finde-Maschine



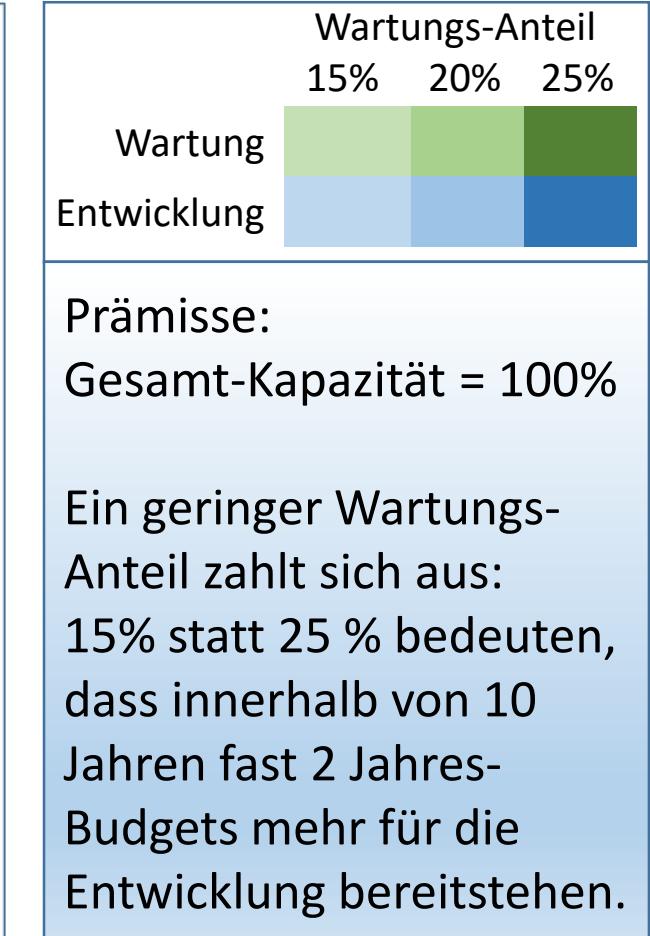
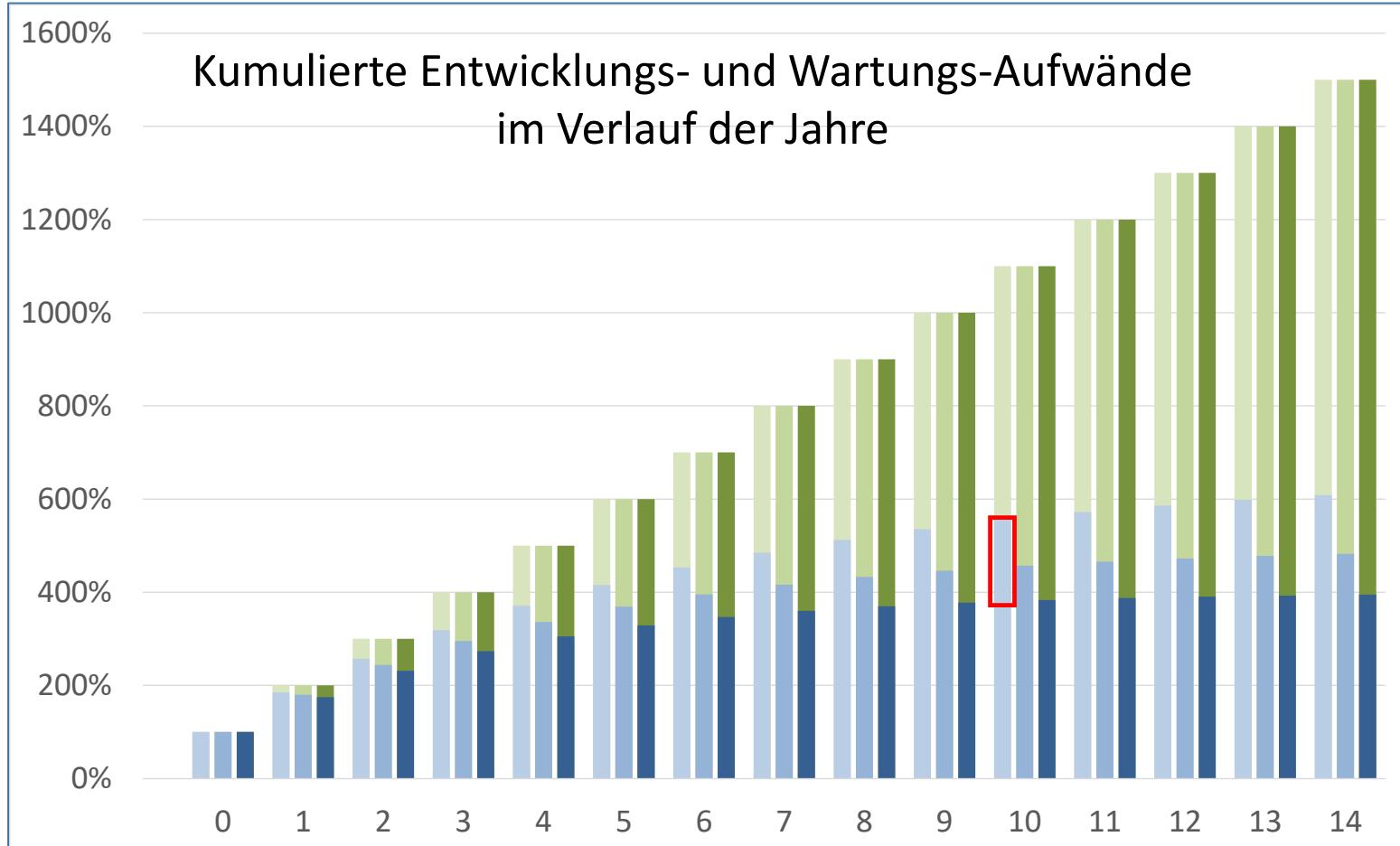
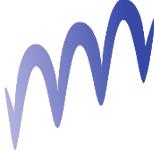
OZG-Pendant:
Verwaltungs-zentrierte Vorgänge und Formulare

OZG-Pendant:
Digitale Leistungen entworfen für die Nutzer

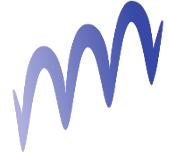
Die OZG-Umsetzung soll um mehrere Größenordnungen wachsen



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



Die Nachnutzung kann auf verschiedenen Ebenen stattfinden



Nachnutzung „im Großen“

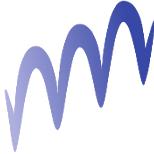
- Modell A
 - Einer für Alle/Viele
- Modell B
 - Nachnutzbare Software dezentral betrieben
- Modell C
 - FIM-basierte Eigenentwicklung

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

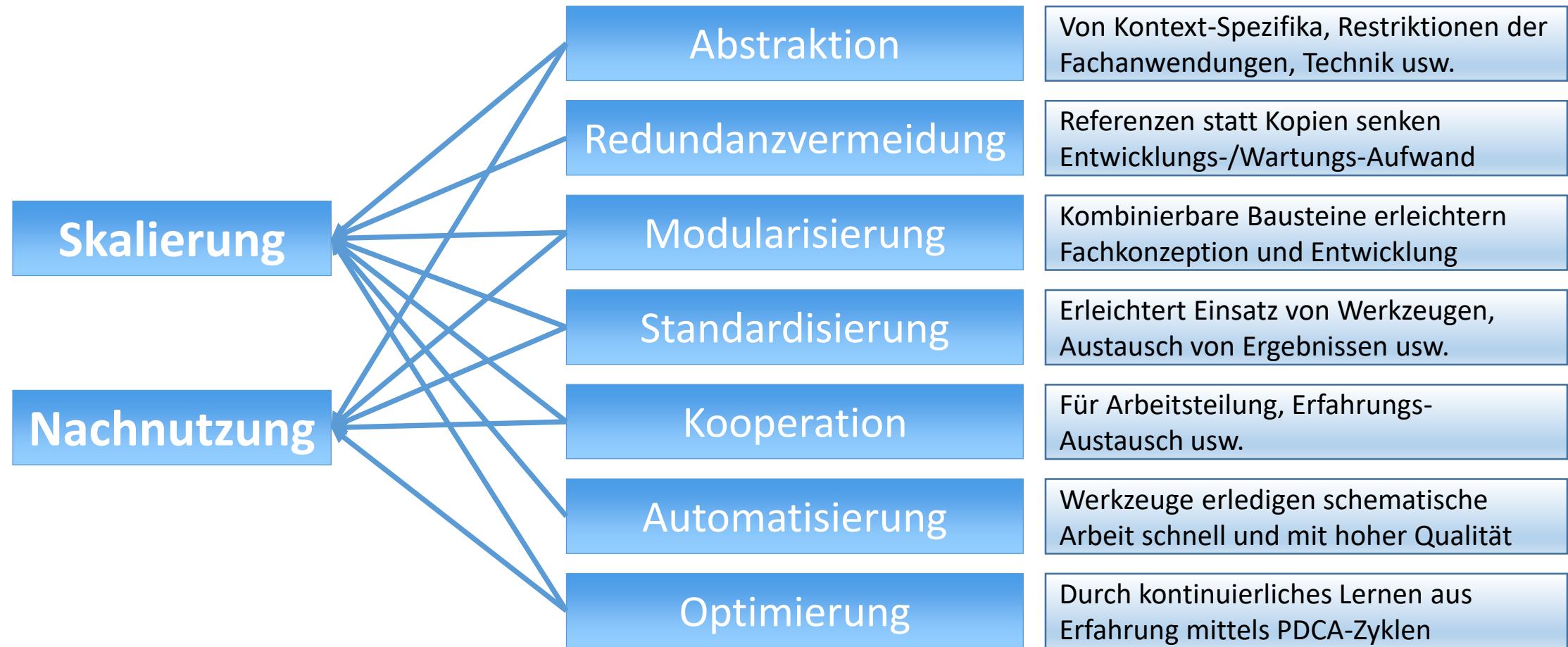
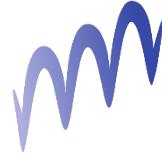
* Don't Repeat Yourself; siehe „The Pragmatic Programmer“, 2. Auflage, 2020.

Nutzer-zentriertes Design verfügt über zahlreiche bewährte Praktiken



Jacques Carelman: Catalogue d'objets introuvables; Paris 1969

Pragmatisches Software Engineering förderst Skalierung und Nachnutzung



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

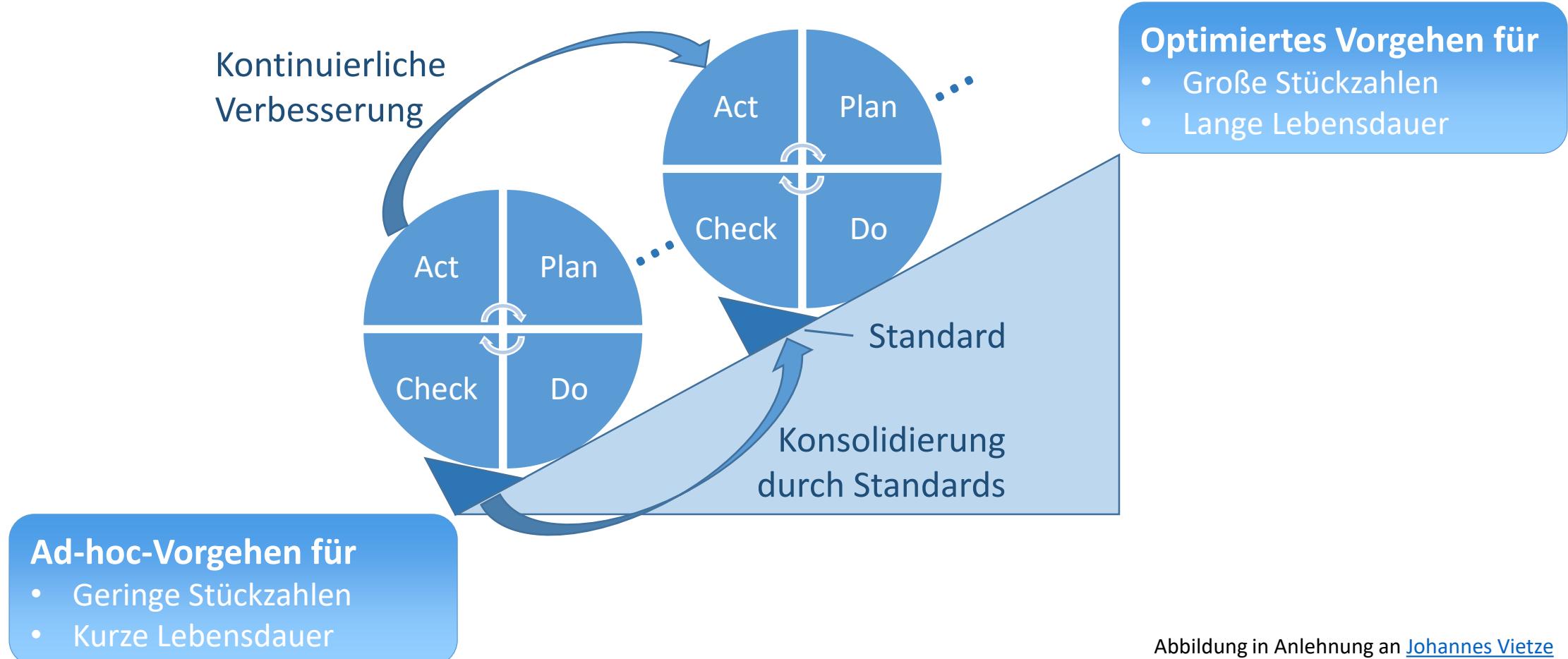
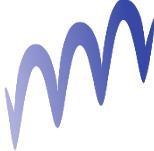
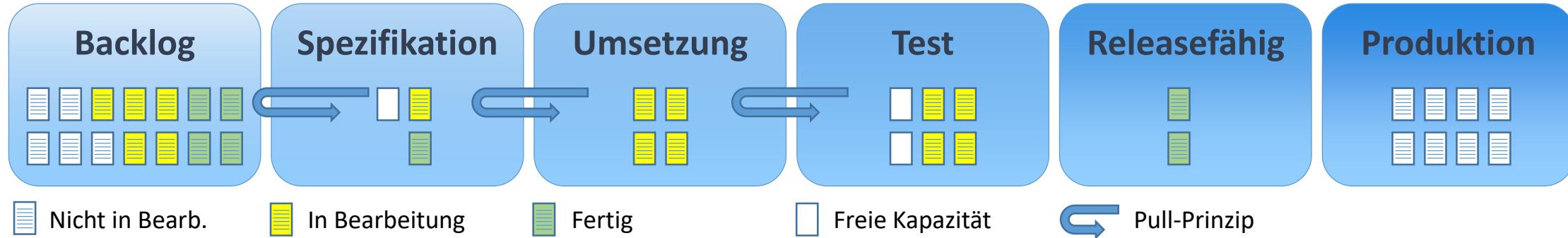


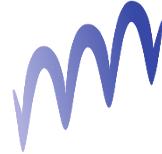
Abbildung in Anlehnung an [Johannes Vietze](#)

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 volatilum 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

Einen Leistungs-Antrag kann man in großen Teilen maschinenlesbar spezifizieren



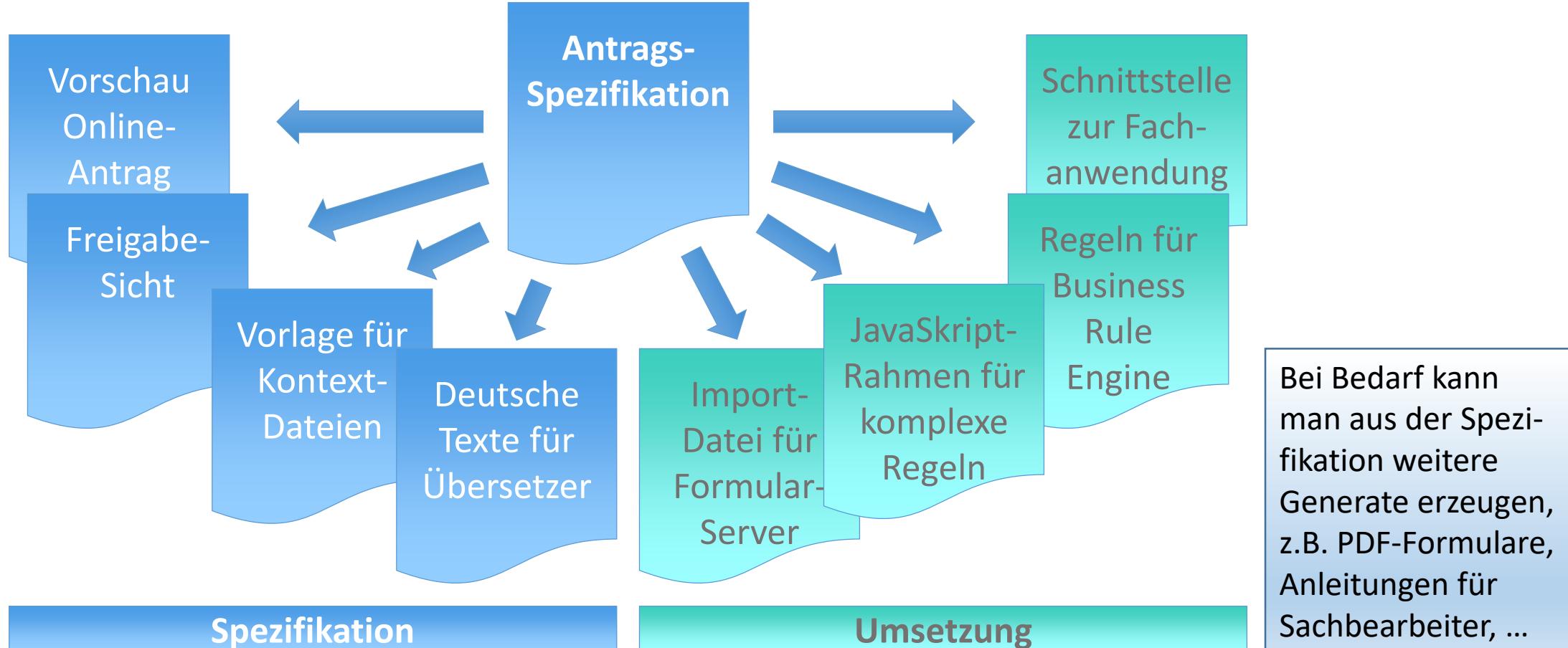
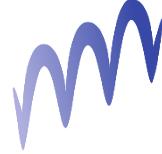
Antrags-Spezifikation

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

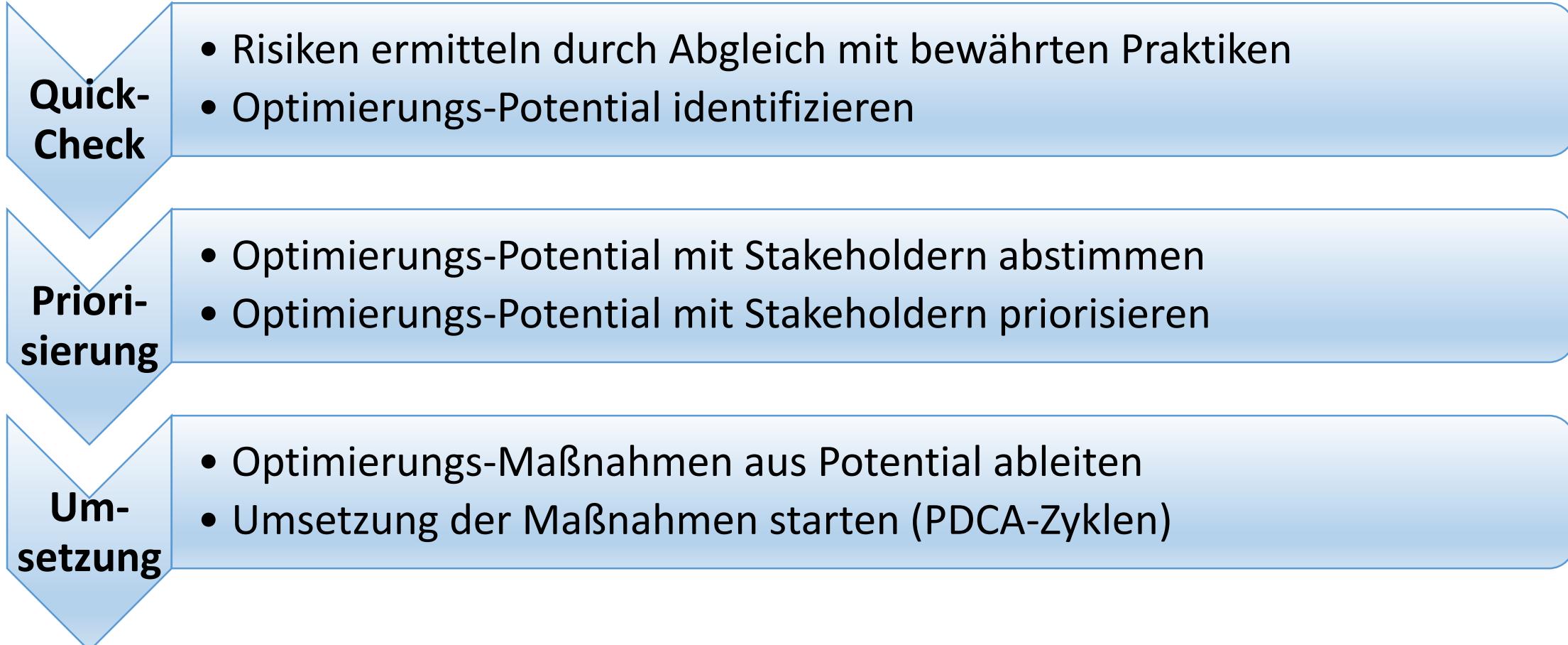
Antrags-Spezifikation	Standard
Fachliches Datenmodell	UML
Struktur und Inhalt	XML / XSD
Kontexte	XML / XSD
Regeln	DMN, RegEx
Aktionen	UML / XÖV

* Zur Parametrisierung je Organisation (Nutzungs-Kontext) bzw. je Kanal (Kanal-Kontext, z.B. Formular-Server, Chatbot, PDF-Generierung usw.)

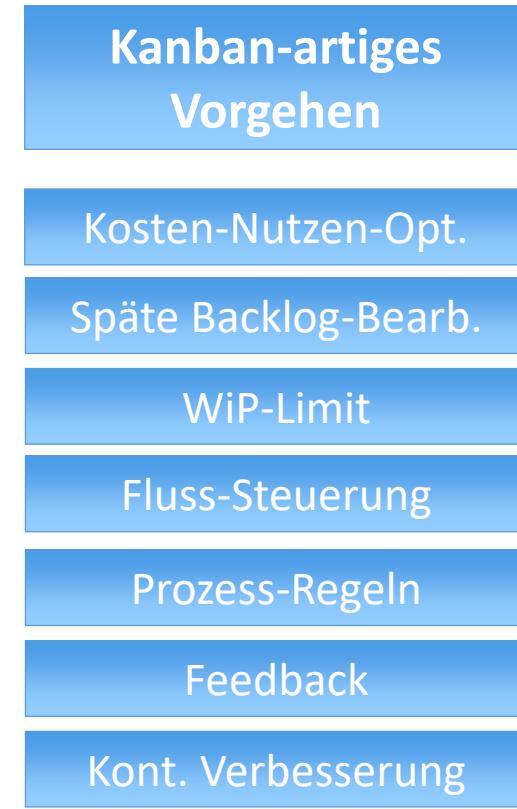
Maschinenlesbare Antrags-Spezifikationen kann man relativ einfach transformieren



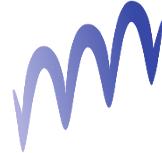
Ein schlanker Prozess zur effizienten Optimierung integriert Best Practices und Nutzer-Sichten



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



Eine Liste potentieller Optimierungs-Maßnahmen erleichtert die Priorisierung



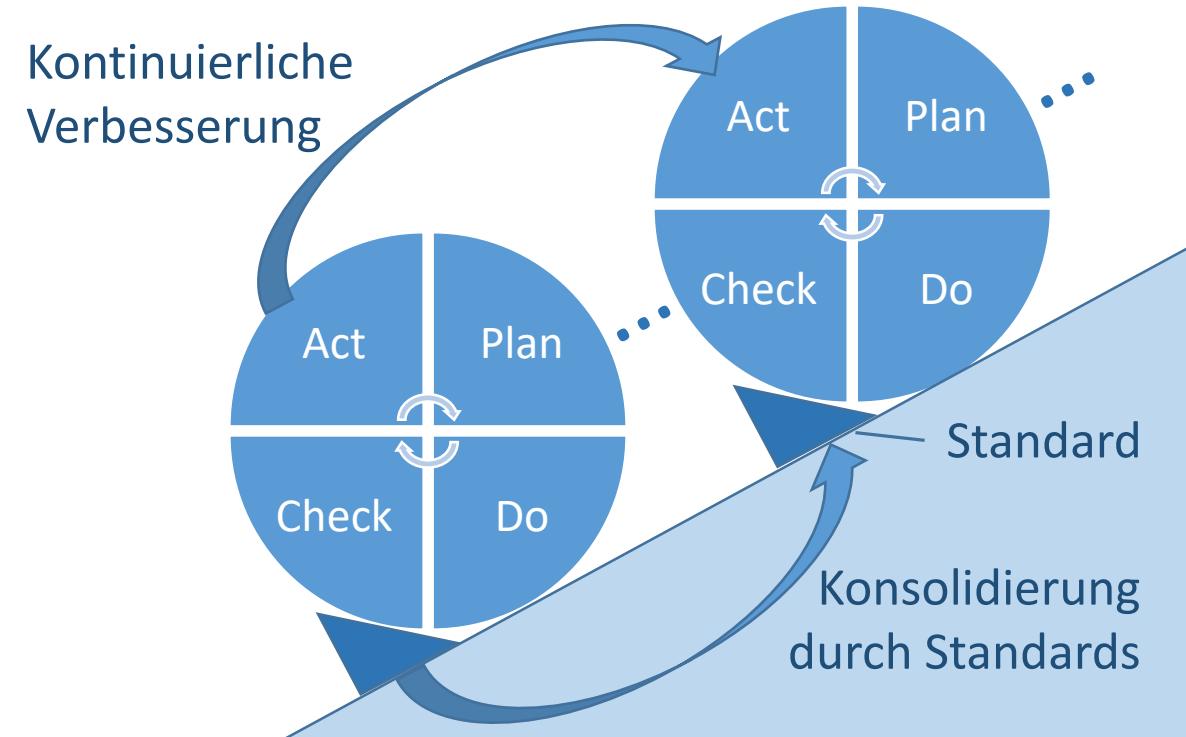
#	Maßnahmen	Nutzen	Abh.	N ¹	K ²
1	UML-Datenmodell	Fachwissen dokumentiert => MDD			
2	XML-Antrags-Spezifikation	Produkt-Unabhängigkeit	1		
3	XML-Kontexte	Entkopplung von Nutzungs- und Kanal-Spezifika	1, 2		
4	DMN-Regeln	Regeln als ausführbare Spezifikation	1		
5	UML/XÖV-Aktionen	Standards fördern Werkzeug-Einsatz	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

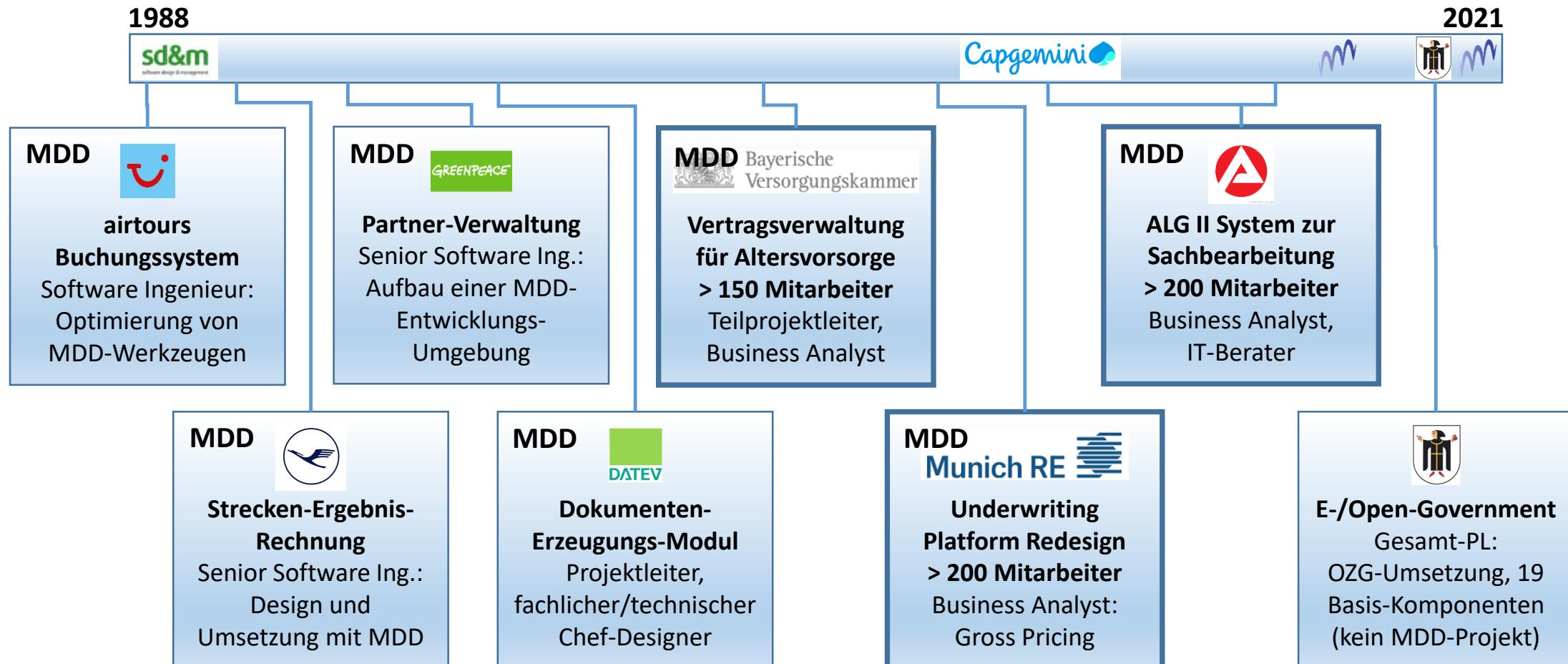
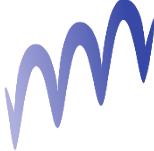


Telefon: [0160 9860 2234](tel:016098602234)

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.