

Requirements Engineering / Software Engineering: **Spezifikationen** (in Softwareentwicklungsprojekten)

Eine Übersicht
Für Softwareentwickler,
Requirements Engineers und Projektmitarbeiter
Stand: 12/2016

Sie finden diese und weitere
Präsentationen unter (→ Klick):
<https://www.peterjohann-consulting.de/praesentationen/>

Alle Rechte vorbehalten. Reproduktion zum nicht-kommerziellen Gebrauch mit Quellenangabe gestattet. Reproduktion – auch auszugsweise – zum kommerziellen Gebrauch sowie der Gebrauch für Vortragszwecke sind nur mit schriftlicher Bewilligung des Verfassers gestattet.

Zusammengestellt von H. Peterjohann
Zur Verteilung an Interessierte
Version 1.50 vom 20.12.2016
44 Seiten



Diese Ausarbeitung ist eine (eigenständige) Anleitung für Softwareentwickler, um ohne großen theoretischen „Ballast“ recht schnell zu einer einfachen, aber dennoch gebrauchsfertigen Spezifikation zu gelangen.

Daher wird hier ein einfaches Projektszenario vorausgesetzt, bei welchem dem Leser die Inhalte, der Projektumfang und der technische Rahmen seines Projekts bereits bekannt sind. Der Leser ist ein Softwareentwickler mit Vorkenntnissen, dem die gängigen Methoden und Modelle des Software Engineerings geläufig sind.

Bitte beachten Sie:

Als Ergänzung zu dem Thema „Spezifikationserstellung“ kann die Präsentation „**Requirements Engineering: Eine Einführung**“ herangezogen werden, die ebenfalls frei auf der Website verfügbar ist: <https://www.peterjohann-consulting.de/pdf/peco-re-einfuehrung.pdf>).



Folgende Inhalte werden in dieser Ausarbeitung behandelt und sollten Ihnen nach dem Durcharbeiten bekannt sein:

- Sie wissen, was Spezifikationen sind
- Sie kennen die Vorgehensweisen bei der Erstellung von Spezifikationen (in Softwareentwicklungsprojekten)
- Sie können die wesentlichen Elemente einer Spezifikation benennen

Zielgruppe: Softwareentwickler, Requirements Engineers und Projektmitarbeiter

Voraussetzungen: Erste Erfahrungen mit Requirements in Projekten

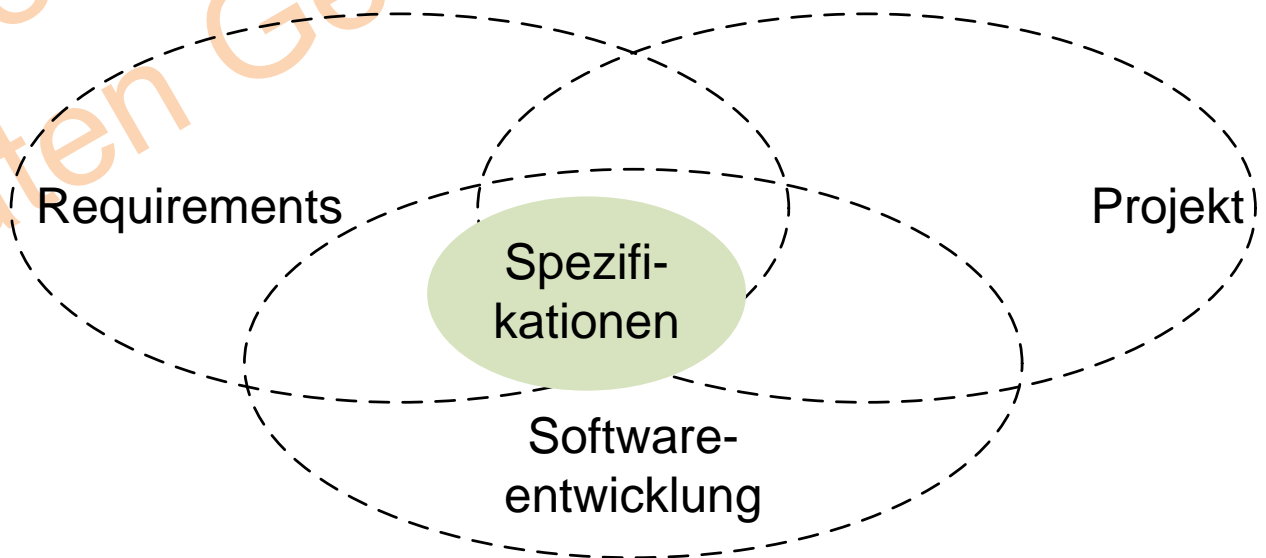
Schwierigkeitsgrad: Gering bis mittel



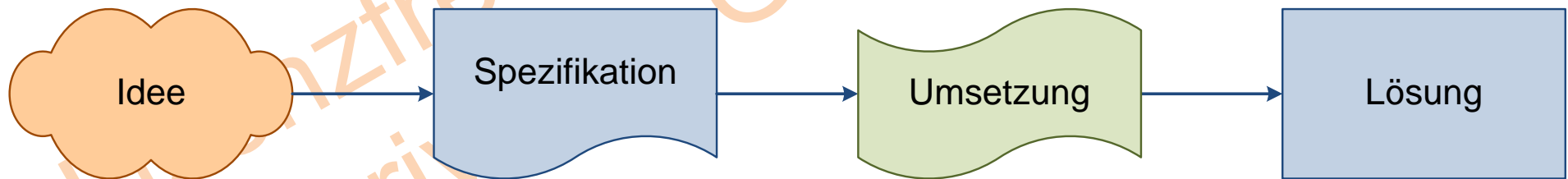
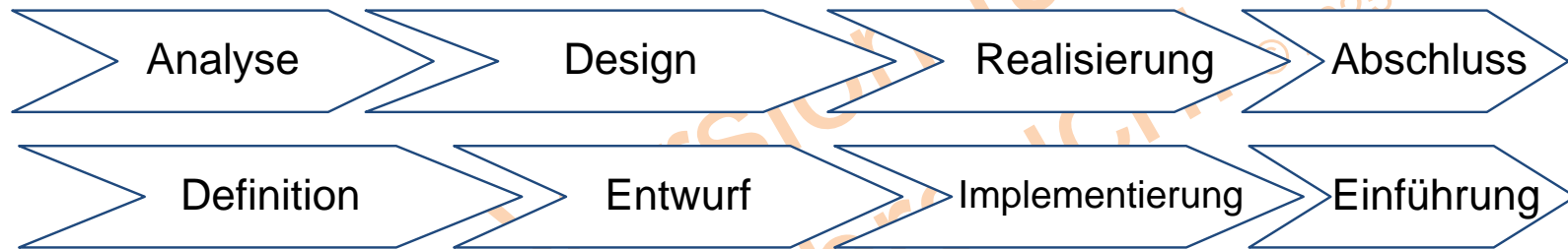
- Einleitung
- Die Spezifikation im SW-Erstellungsprozess
- Zum Begriff der Spezifikation
- Motivation (Grundsätzliches, Das Wasserfallmodell, Gesamtaufwand und Kosten)
- Problemkreise (Muss spezifiziert werden?, Wer spezifiziert?)
- Der Fokus der Feinspezifikation
- Die Feinspezifikation als zentrales Element
- Die Anforderungsentwicklung
- Die Spezifikation und Änderungen
- Wortwahl
- Von der Spezifikation zur Abnahme
- Struktur der Anforderungsspezifikation (IEEE 830-1998)
- Die Gliederung einer Spezifikation (eigenes Modell)
- Der Start
- Was gehört hinein?
- Merkgeregeln (Richtige Begriffe & Zeitbezug, Elemente, Organisation)
- Der Publikationsprozess
- Die Status der Spezifikation
- Aufgaben und Verantwortlichkeiten des Projektmanagers
- Fragen
- Spezifikationen und Agilität
- Checkliste: Ist die Spezifikation formal korrekt?
- Anhang (Literatur, Weblinks, Meine Dienstleistungen – Das kann ich für Sie tun, Kontakt zum Autor)

Die Spezifikation ist ein zentrales Element bei der Produktentwicklung, insbesondere bei der Erstellung von Software-Produkten. In ihr wird festgehalten, was entwickelt werden soll – eine gute Spezifikation hilft, dass richtige Produkt richtig zu entwickeln.

Diese Ausarbeitung beschäftigt sich damit, wie eine Spezifikation aussehen könnte (formaler Aufbau) und wie sie in einen organisatorischen Ablauf(prozess) innerhalb von Projekten eingebettet werden sollte.



Die „Spezifikation“ unterstützt den SW-Erstellungsprozess von der Aufnahme der Idee oder des Problems bis zur Einführung der Lösung!





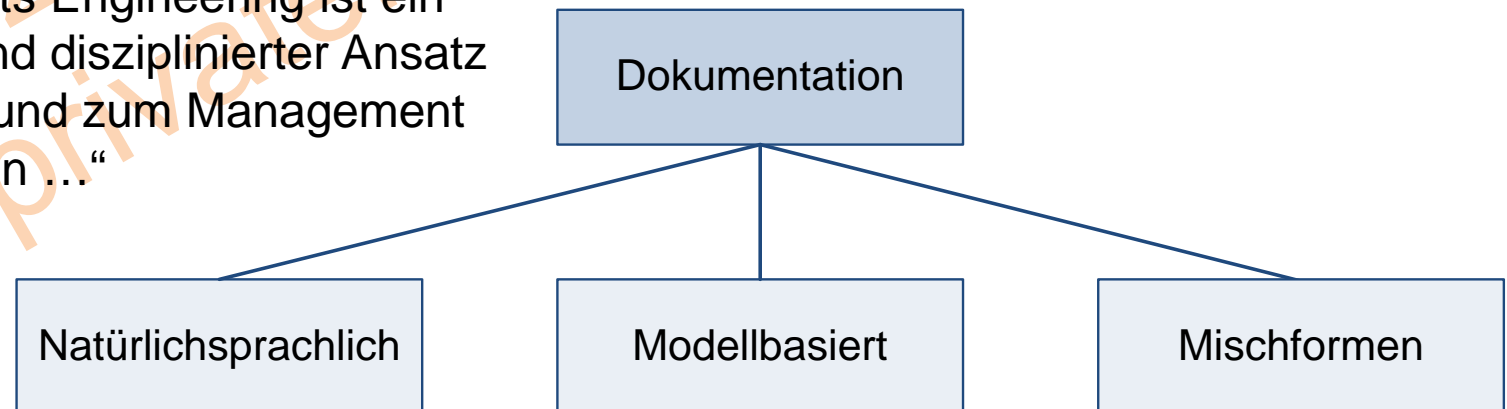
Beim International Requirements Engineering Board – IREB /IREB15/ – wird die Spezifikation als Anforderungsspezifikation bezeichnet und folgendermaßen definiert:

„Eine **Anforderungsspezifikation** ist eine systematisch dargestellte Sammlung von Anforderungen (typischerweise für ein System oder eine Komponente), die vorgegebenen Kriterien genügt.“

Anforderungsspezifikationen können dabei in natürlicher Sprache oder durch konzeptuelle Modelle (oder als Mischform) verfasst werden.

Weiter führt das IREB /IREB15/ aus:

„Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen ...“





Der Begriff „Spezifikation“ ist nicht eindeutig festgelegt – es finden sich auch folgende Begriffe: Lastenheft, Pflichtenheft, Software Requirements Specification, Systemspezifikation, Anforderungsspezifikation, Lösungsspezifikation, Produktspezifikation, ...

In der Spezifikation werden alle (zu berücksichtigenden) **Anforderungen** untergebracht.

- **Anforderungsspezifikation:**
Beschreibt, **was** von der Lösung erwartet wird. Wird auch **Lastenheft** genannt
- **Systemspezifikation:**
Beschreibt, **wie** die Lösung ausschauen wird. Wird auch als **Pflichtenheft** bezeichnet

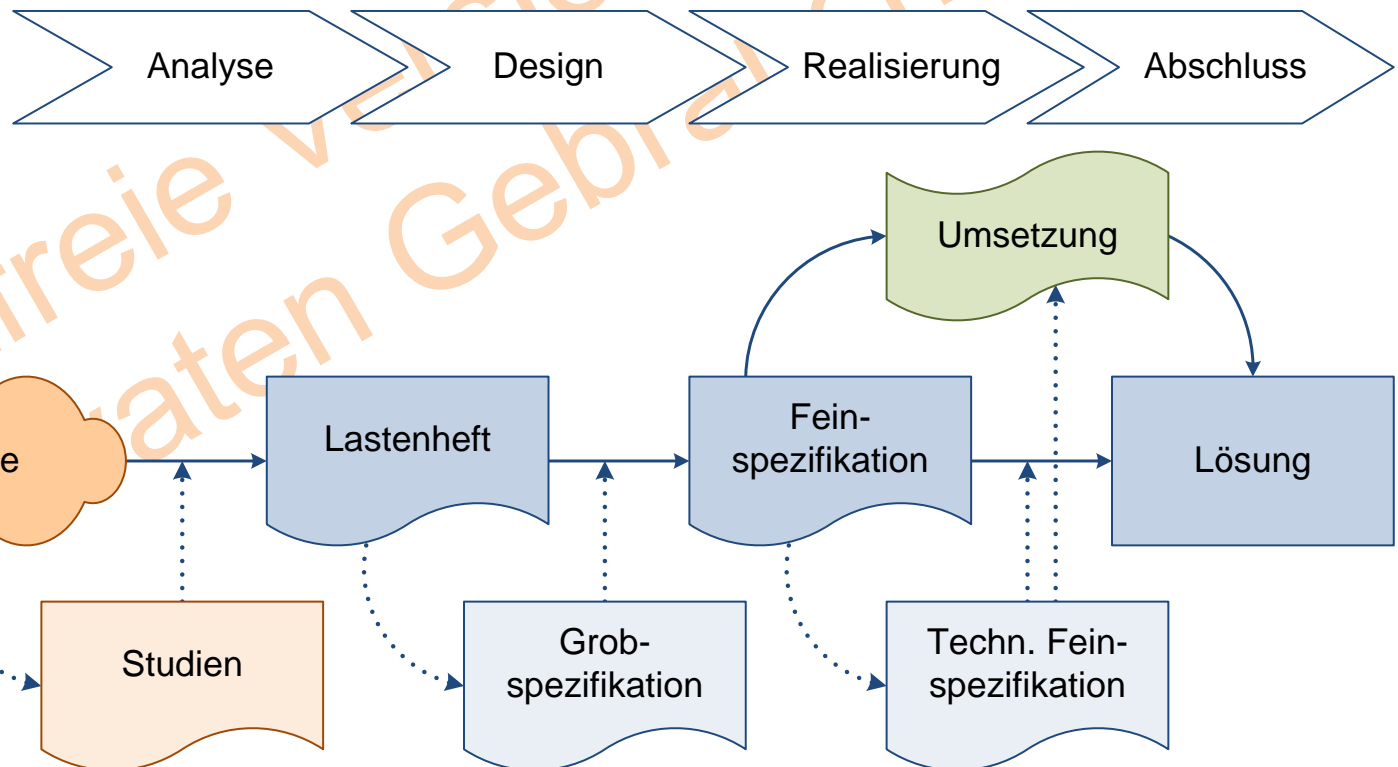
Zum Thema **Lastenheft und Pflichtenheft** gibt es eine eigenständige Präsentation des Autors, die ebenfalls auf der Website unter https://www.peterjohann-consulting.de/_pdf/peco-pm-lastenheft-und-pflichtenheft.pdf frei herunterladbar ist.



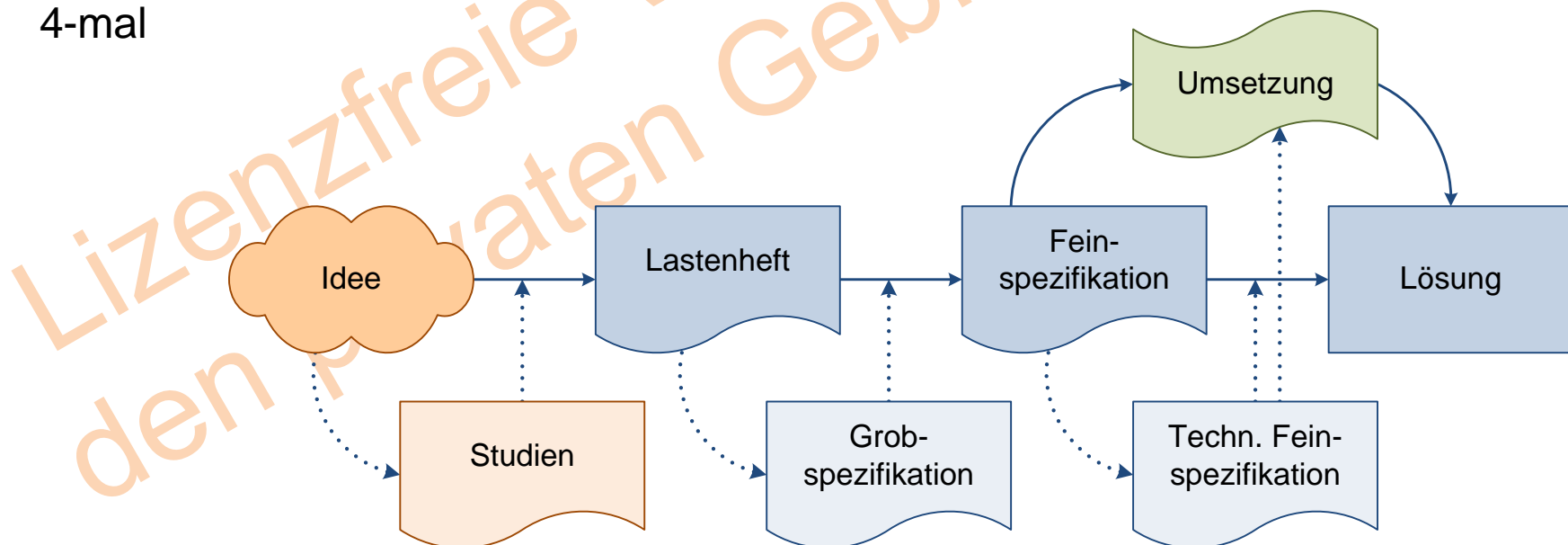
	Definition	Unterkategorie	Definition	
Lastenheft (Anforderungs- spezifikation)	<ul style="list-style-type: none"> Beschreibt, was zu tun ist (was von der Lösung erwartet wird) Hat rechtliche Relevanz (Vertragsgrundlage) DIN 69901-5:2009: Vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines (Projekt-)Auftrags. 		<p>Ist hier der Hauptfokus</p>	
Pflichtenheft (System- spezifikation, Lösungs- spezifikation, Produktspezifikation) (weniger gebräuchlich: Sollkonzept, Fachfeinkonzept, fachliche Spezifikation)	<ul style="list-style-type: none"> Beschreibt, wie die Lösung ausschauen wird (was von der Lösung erwartet wird) Basiert auf dem Lastenheft DIN 69901-5:2009: Vom Auftragnehmer erarbeitete Realisierungsvorgaben auf der Basis des vom Auftraggeber vorgegebenen Lastenheftes 	Grobspezifikation		Liefert eine Ausarbeitung des Lastenhefts, verzichtet aber auf Details wie die Benennung einzelner Fenster-Elemente
		Feinspezifikation		Benennt alle Details der Lösung (nicht unbedingt technisch), ist aber für den Key User wie auch für den Entwickler lesbar und verstehbar
		Technische Feinspezifikation		Benennt alle Details der Lösung, so dass ggf. das Dokument nicht mehr allgemein verständlich ist

Unser Ansatz hier:

- **Es wird nur die Erstellung der Feinspezifikation betrachtet**
- Das Lastenheft ist nur als „mündliche Vereinbarung“ vorhanden (Inhouse-Projekt)
- Die Spezifikation treibt den Softwareerstellungsprozess

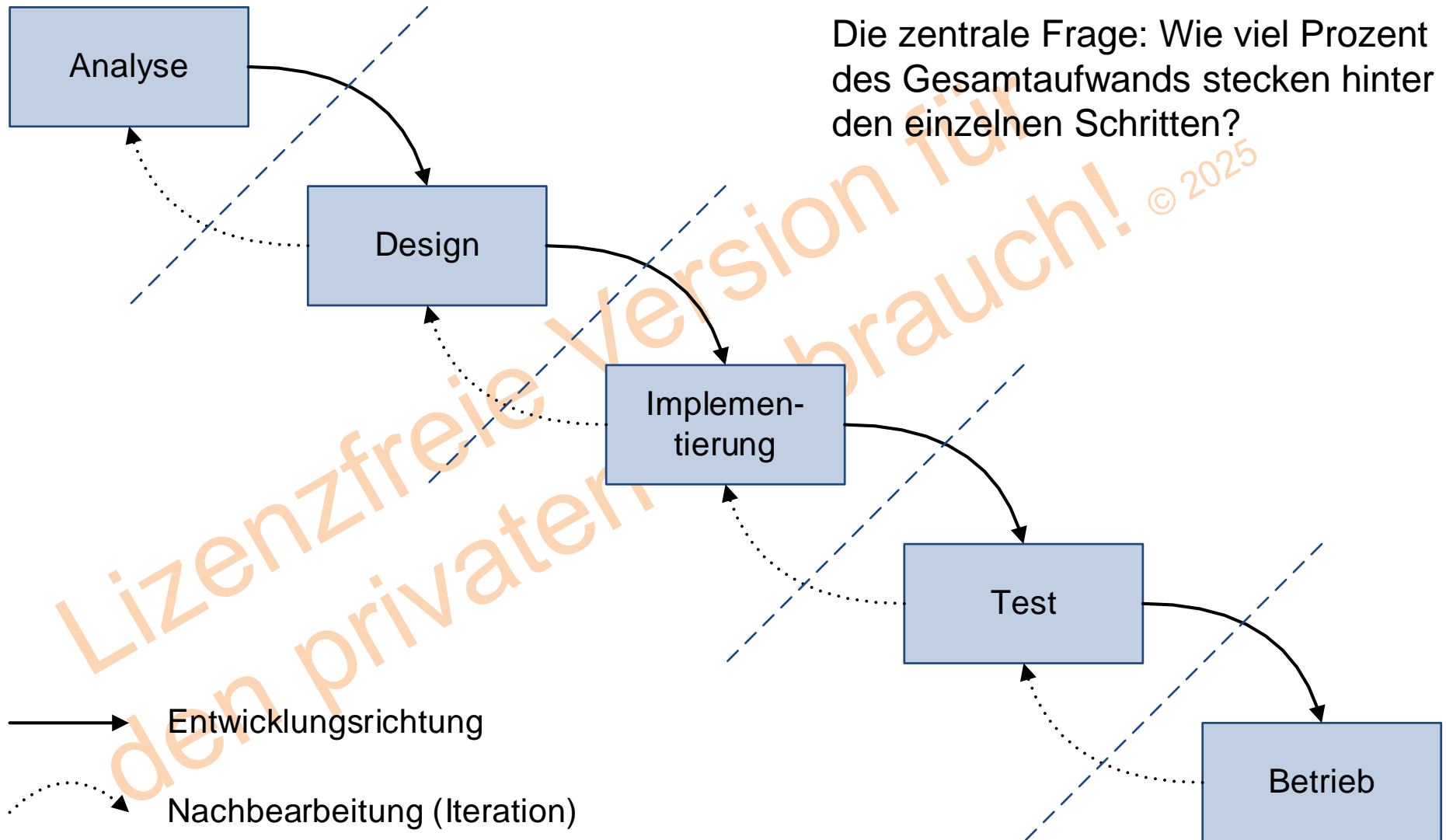


- Bei der Analyse entsteht ein Lastenheft, welches nur einmal erstellt wird (und damit statisch ist); das Lastenheft hat rechtliche Relevanz
- Beim Design entstehen: die Grob-, die Fein- und die Technische Feinspezifikation, wobei die Unterscheidung zwischen Grob-, Fein-, und Technischer Feinspezifikation in der Praxis kaum noch vorgenommen wird; stattdessen werden häufig nur die Feinspezifikationen für die (Teil-)Applikationen erstellt
- Typische Größenordnungen: Die Feinspezifikation ist etwa 2- bis 3-mal so umfangreich wie die Grobspezifikation, die Technische Feinspezifikation etwa 3- bis 4-mal





- Egal, welche Vorgehensweise (welches Vorgehensmodell) man bei der Softwareerstellung beherzigt: Spezifikationen müssen (fast) immer erstellt werden
- Bei Großprojekten: etwa 10 % der Zeit fallen auf die Codierung, aber 30 % auf die Erstellung der Anforderungsdokumente (Analyse und Design), der Rest auf die Wartung und den Betrieb
- Im Wesentlichen werden in den Spezifikationen die „Ws“ beschrieben (Wer, Wann, Was, Womit, ...)
- Die Spezifikation dient zur Abstimmung der Entwicklungsarbeit zwischen Entwickler und Benutzer und wird i.A. durch den Requirements Engineer erstellt; die Spezifikation kann zudem als Vertragsgrundlage (Ausschreibung), Kommunikationsgrundlage, Grundlage für die Systemintegration, Wartung sowie Pflege, Grundlage für Architektur, ... dienen
- Disziplinen hierzu: Requirements Engineering (RE) /Ebert14, Rupp14/, Systemanalyse und -design /Balzert08, Balzert09/





- Typische Größenordnung: 60 bis 80 % der Kosten einer Software (im gesamten Lebenszyklus) entstehen nach der ersten Produktivsetzung, d.h. im Betrieb und bei Nachbesserungen
- Um den Gesamtaufwand im Rahmen zu halten, müssen nachträgliche oder zusätzliche Änderungen (Iterationen) vermieden werden
- Dies geht nur, indem die Anforderungen möglichst früh (zu Beginn) so gut wie möglich erfasst werden
- „Lieber mehr Aufwand in die Spezifikation stecken und dann anschließend geringere Betriebs(nachfolge)kosten tragen“



Immer:

- Bei Projekten mit externen Kunden muss immer ein Lastenheft erstellt werden
- Bei Projekten mit rechtlichen oder starken sicherheitstechnischen Randbedingungen (Medizin, Luftfahrt, Automotive) muss ebenso wie bei Großprojekten komplett (bis in das Detail) spezifiziert werden
- Bei „schwierigen Randbedingungen“ sollte spezifiziert werden

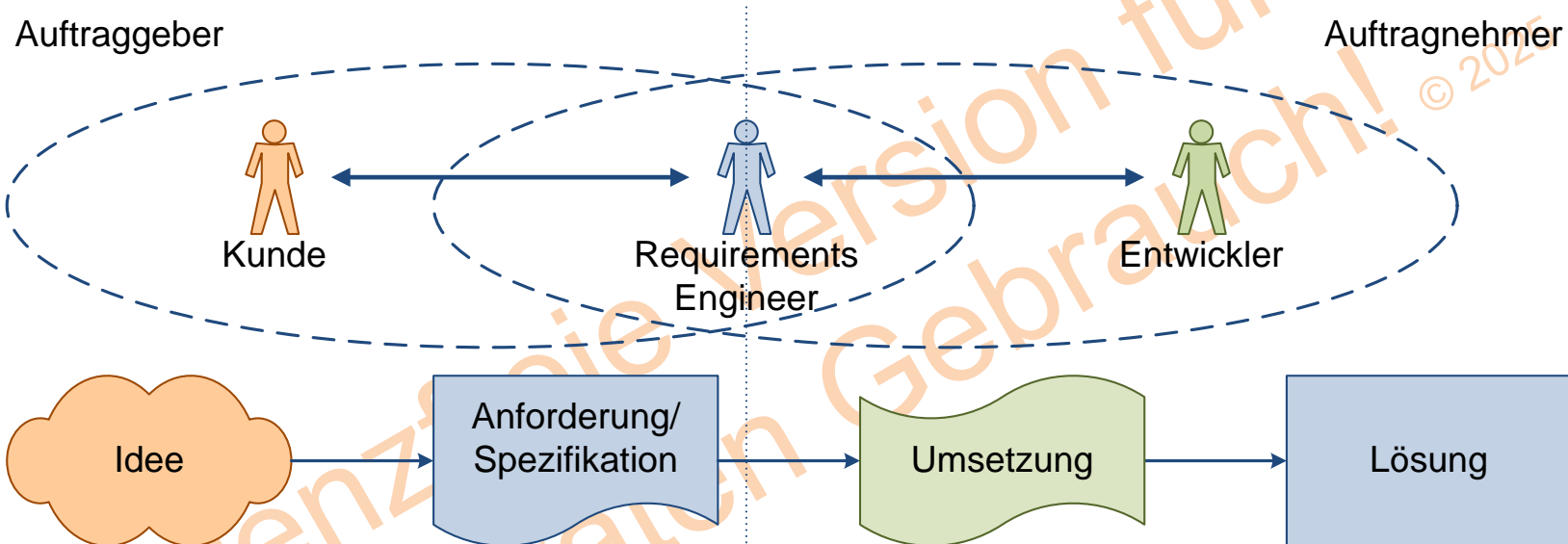
Nicht unbedingt:

- Bei Kleinprojekten mit internen Kunden, wenn die Inhalte bekannt sind
- Bei One-Man-Shows

Regel:

- Wenn das Risiko des Nichtspezifizierens getragen werden kann, so muss nicht (unbedingt in vollem Umfang) spezifiziert werden

Die Spezifikation könnte durch den Kunde oder durch die Entwickler erstellt werden (Problemkreis: Auftraggeber oder Auftragnehmer?).

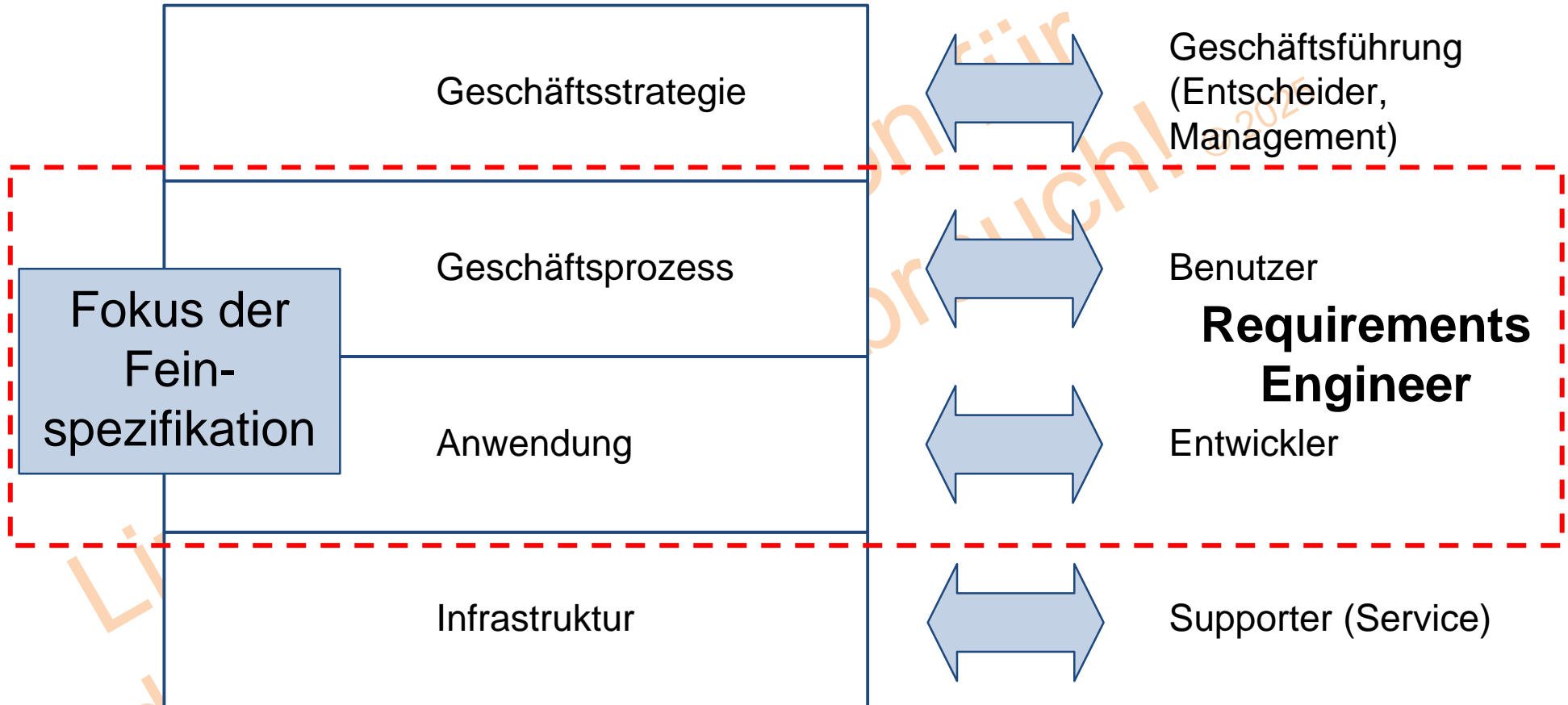


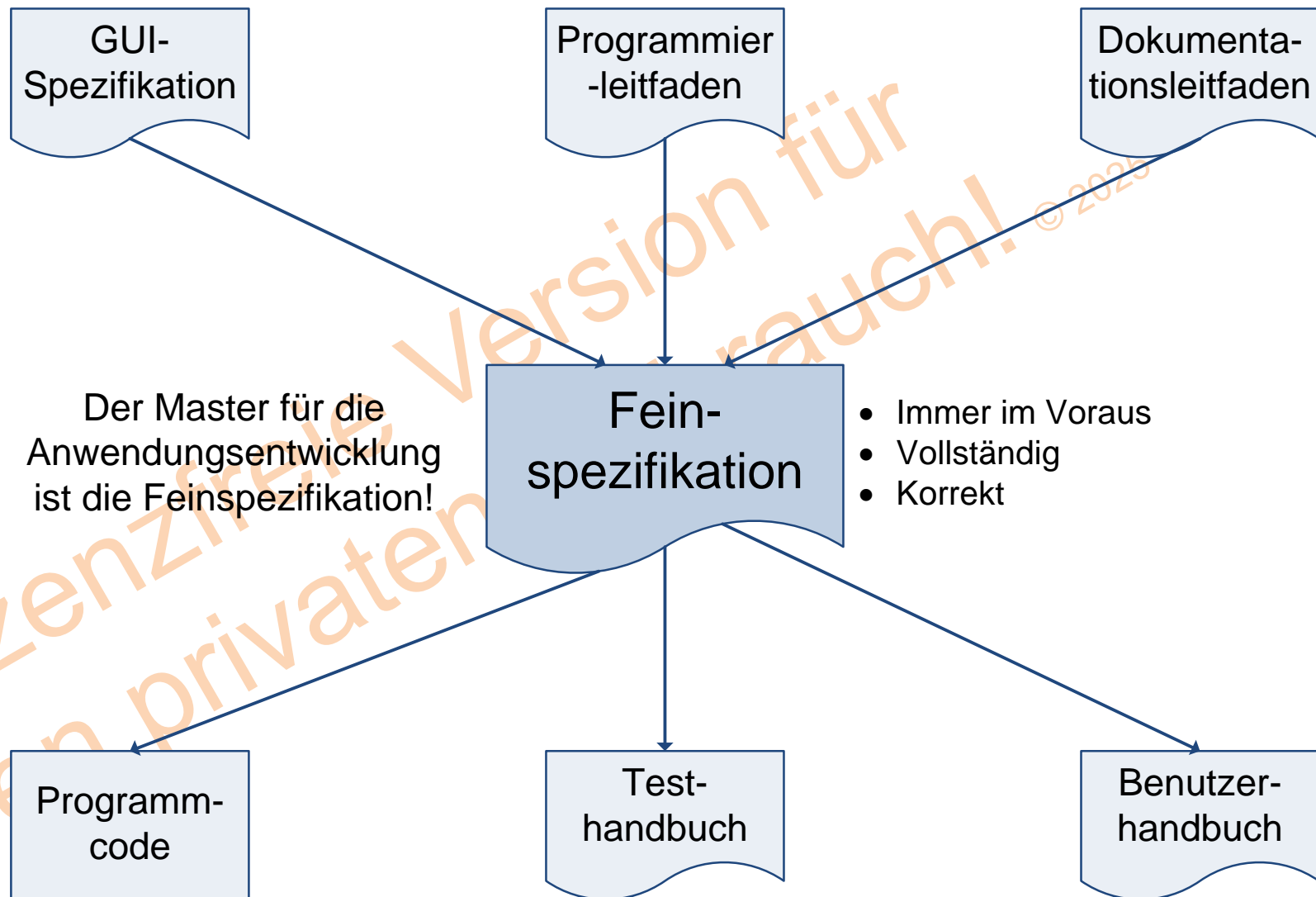
- weiß, was er will
- kann ausdrücken, was er will
- ist Repräsentant aller Kunden und Benutzer

- versteht die Sprache des Kunden und des Entwicklers
- versteht den Problem- und den Lösungsbereich
- ist Mittler zwischen Kunde und Entwickler
- hat genügend Zeit, die Anforderungen zusammenzutragen

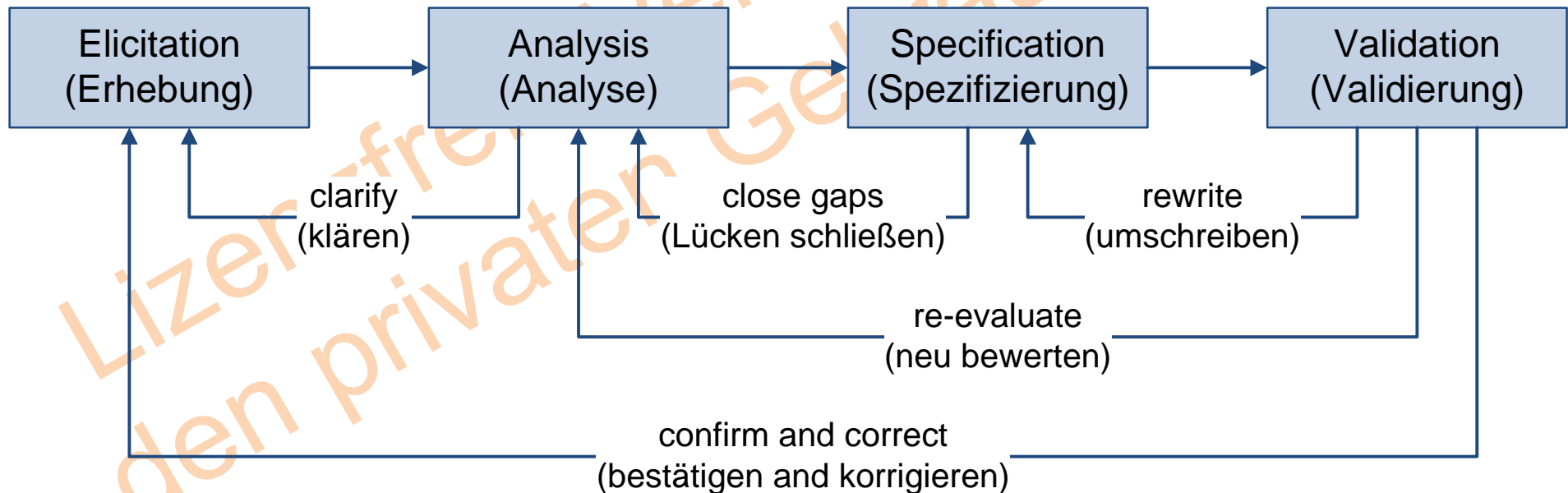
- liest die Spezifikationen (und versteht sie)
- setzt die Spezifikation um

Hier: Der Requirements Engineer ist Mitglied des Entwicklungsteams!





- Anforderungen müssen „nach und nach“ entwickelt werden; dies setzt neben den technischen Fähigkeiten auch fachliche, organisatorische und kommunikative Fähigkeiten beim Ersteller voraus
- Die Erstellung (alternativ auch: Entwicklung) der Spezifikation ist ein iterativer, inkrementeller Prozess

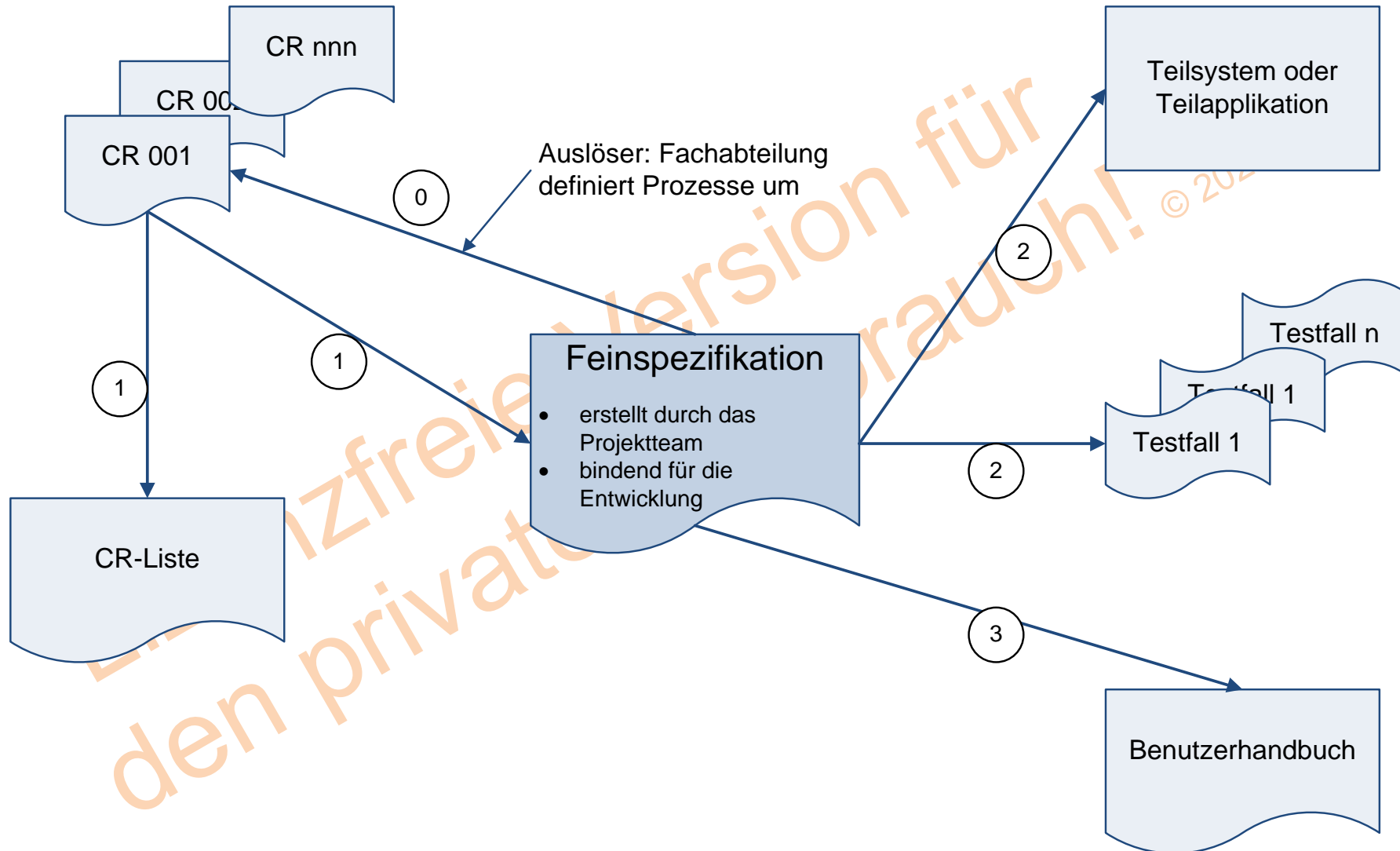


/Wieggers13/



- Änderungen treten leider auf; diese müssen in einem definierten Prozess in das Software-System zur Umsetzung gelangen
- Zunächst werden Änderungsanforderungen (Change Requests) gestellt und dann in den Erstellungsprozess eingebracht
- Katastrophal: Änderungen direkt in die Softwareentwicklung einfließen lassen
- Schlecht: Änderungen unmittelbar in die Spezifikation und dann in die Entwicklung einfließen lassen
- Gut: Änderungen über einen Änderungsprozess (Änderungs- oder CR-Liste) in die Spezifikation und dann in die Entwicklung einfließen lassen

**Der Änderungsprozess ist sehr komplex und wird hier nur kurz gestreift!
Unsere Annahme: Die Änderungen kommen (nach und nach) aus einer zentralen Quelle, z.B. der Änderungs- oder Anforderungsliste.**





Bei der Wortwahl in der Spezifikation ist zu beachten:

- Den Istzustand zu einem bestimmten Zeitpunkt beschreiben
- Muss unabhängig vom Lesezeitpunkt gültig sein, d.h. insbesondere, dass nicht vorausgesetzt werden darf, dass ein Alt-System oder Alt-Abläufe bekannt sind
- Immer „Ist“-Formulierungen verwenden (kein „könnte“ und kein „sollte“)
- Zielpublikum beachten
- Wichtig: Die Spezifikationen müssen klar, vollständig, testbar, korrekt und eindeutig sowie umsetzbar (verstehbar) sein!

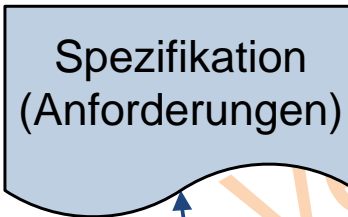


Verbindlichkeit	Schlüsselwort	Keyword
Pflicht	muss	shall
Wunsch	sollte	should
Absicht	wird	will
Kommentar	-	-

/Rupp14/

Alle Anforderungen müssen testbar sein, damit sie abgenommen werden können.

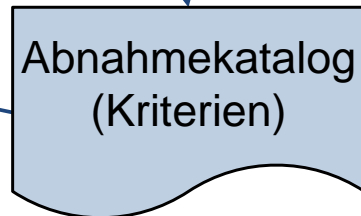
Auftraggeber
(Kunde)



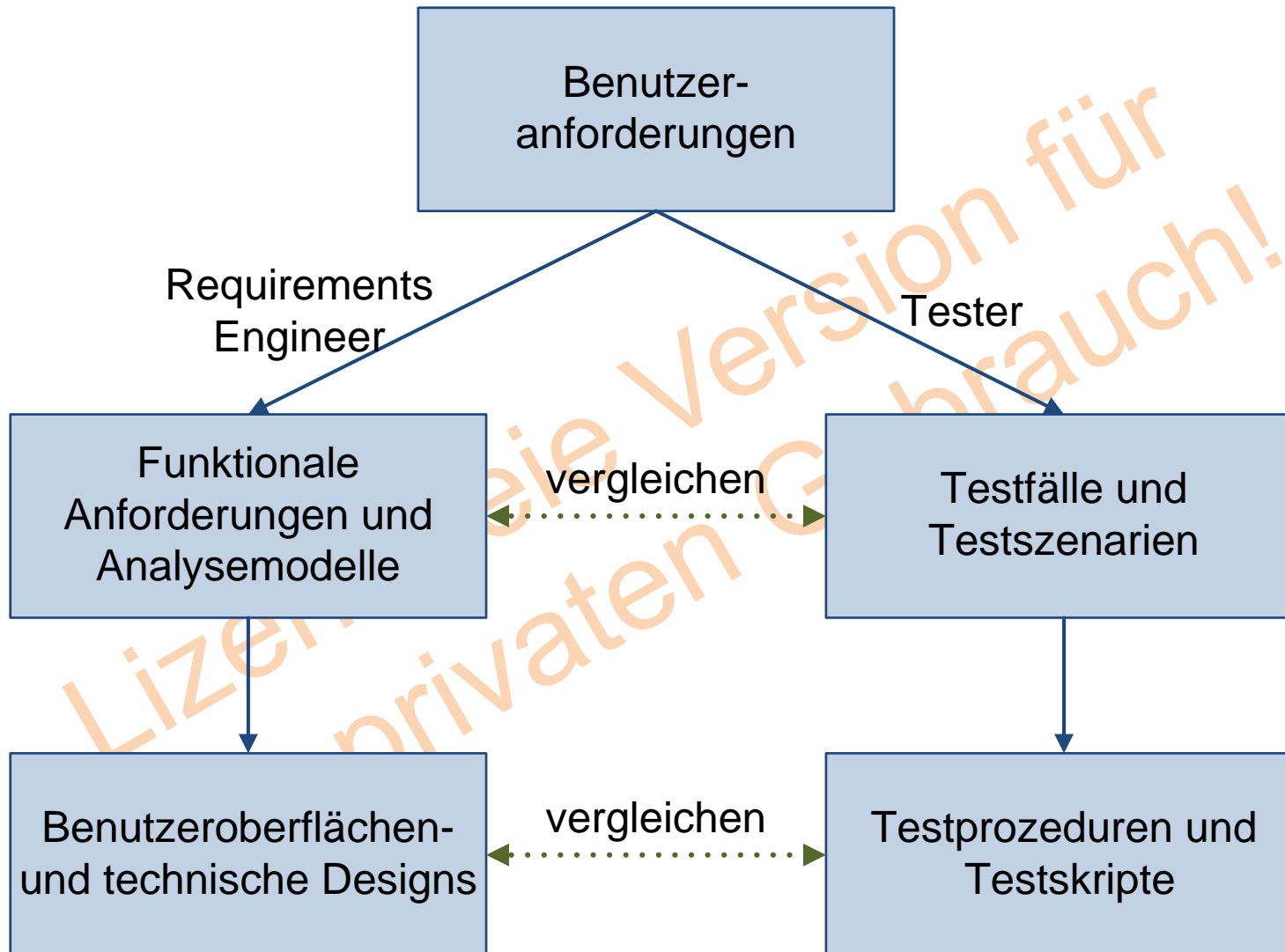
Erstellen von
Abnahmekriterien

Änderungen

Fragen &
Antworten



Auftragnehmer
(Entwickler)



nach /Wieggers05/



Die Anforderungsspezifikation sollte gemäß der Norm IEEE 830-1998 folgendermaßen aufgebaut sein:

1. Einführung
2. Glossar
3. Spezifikation der Kundenanforderungen
4. Systemarchitektur
5. Spezifikation der Systemanforderungen
6. Systemmodelle
7. Evolution des Systems
8. Anhänge
9. Index



Deckblatt, (Vorwort,) Inhaltsverzeichnis, Abbildungsverzeichnis,
Tabellenverzeichnis

1. Einleitung mit Zielgruppe, Sprech- und Schreibweisen, Aufbau, Top-10-Glossar
2. Analyse der Prozesse / Systeme (grundsätzlich); Top-5-Prozesse
3. Beschreibung der neuen Prozesse
4. Beschreibung der Bedienoberfläche
- ...
- X-3. Die Schnittstellen
- X-2. Das Objekt- und das Datenmodell (soweit relevant)
- X-1. Berichte (auch wenn keine vorhanden sind)
- X. Anhang mit Literatur, Historie (der Applikation), Glossar



So sollte mit der Spezifikationserstellung begonnen werden:

- Verwenden der Spezifikations-Schablone
- Ausfüllen der Grundinformationen (Dauer: etwa 10 Minuten)
- Aufbau der Gliederung (mit einem erfahrenen Partner)
- Statische Teile einbringen (Literatur, Historie, Mengen- und Wachstumsgerüst, Zielgruppe, Benutzer und Berechtigungen, Kontexte)
- Menü- und Fensterliste einbringen
- Screenshots der Fenster einbringen und dokumentieren



Mengen- und Wachstumsgerüst, Liste aller Menüpunkte, Liste aller Fenster, Historie, Zielgruppe, Verwendete Schreib- und Sprechweisen, Literatur, Top-3-5-10-Benutzer, Benutzerrollen und -berechtigungen, Begriffe, Prozesse, Aufbau des Dokuments, Kontext-Verbindung zu anderen Applikationen, Einbettung des Systems (in das übergeordnete Gesamtsystem), Listung aller verwendeten Meldungsfenster, Liste aller Berichte und Schnittstellen, Glossar, Literaturverzeichnis, zugelassene Suchbegriffe, Liste der verwendeten Schlüsseltabellen

... und was nicht: komplette Code-Segmente, das Datenmodell in der Komplett-Darstellung, Elemente des Projektplans



- Ist ein Begriff bereits eingeführt, so sollten Alternativen nicht mehr (in Klammernausdrücken) verwendet werden
- Nur Begriffe verwenden, die man selber kennt
- Nicht zu viele Begriffe auf einmal einführen
- Eine Spezifikation sollte entweder ...
 - a) zeitlos oder
 - b) zeitpunktsbezogen (in der Zukunft) sein.

Lizenzfreie Version für
den privaten Gebrauch! © 2025



- Erlaubte Notationen: UML, BPMN, eEPK, ERM, firmenspezifische Nomenklatur, ...
- Sinnvoll auch: Use Cases, Screenshots der Oberfläche, Blockdiagramme
- Einbinden von Screenshots: über ein definiertes Format, z.B. JPG oder PNG (mit Tools wie Greenshot und IrfanView zu erstellen)
- Einbinden von technischen Zeichnungen: nur EMF (Windows Enhanced Metafile) verwenden (über Export-Funktion)
- Das Datenmodell nur in vereinfachter Darstellung einbringen



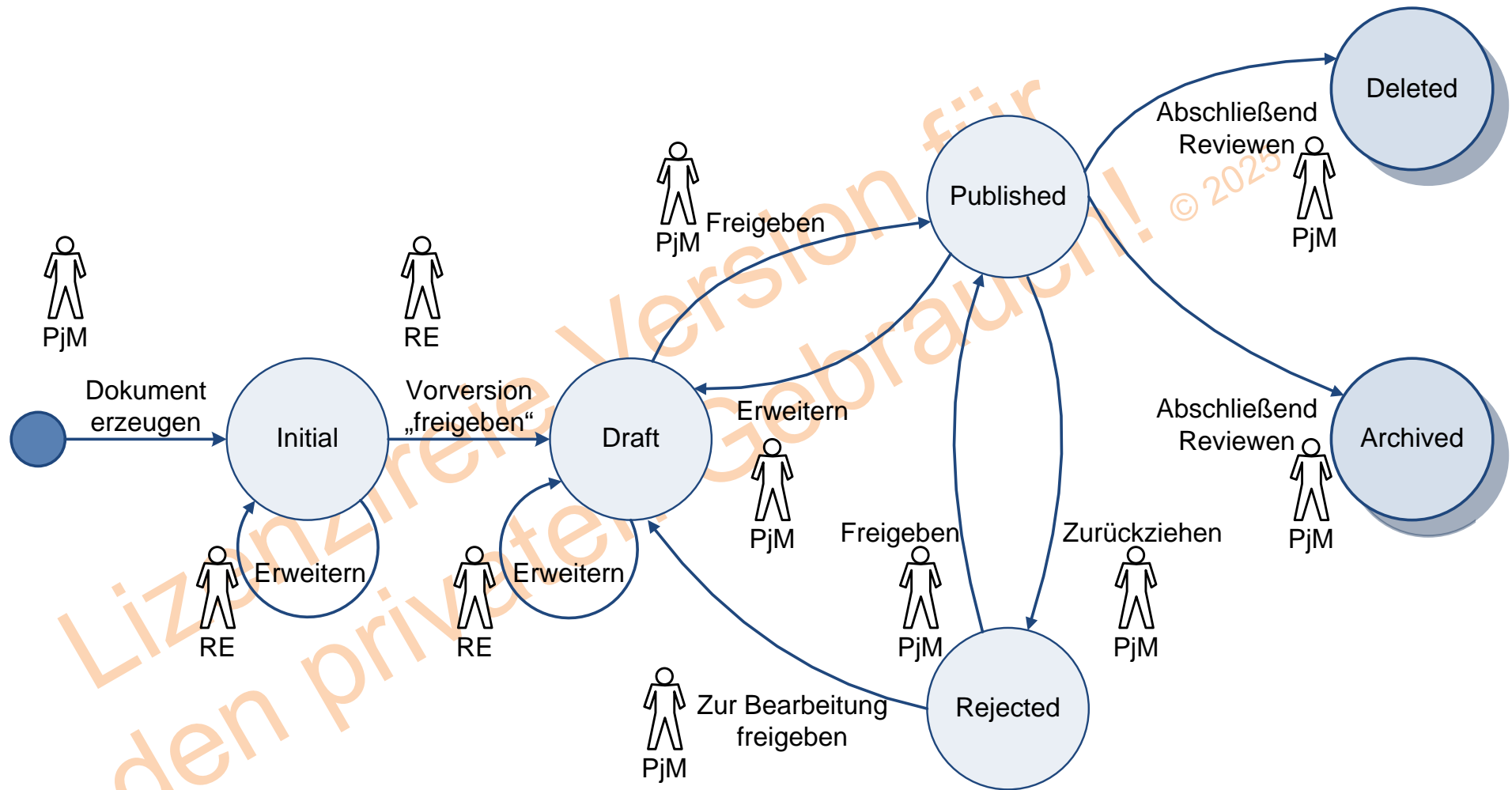
- Jeder Teilprojektmanager (Entwickler) ist für seine (Teil-) Applikation verantwortlich (inkl. der Spezifikation)
- Alle „neuen“ Spezifikationen müssen beim Projektmanager angemeldet werden -> der Projektmanager hält die Spezifikationen zusammen
- Pro (Teil-)Applikation nur eine Spezifikation
- Die Aufwandsschätzung zur Umsetzung erfolgt separat (in einem eigenen Schätzprozess)

Zur **Aufwandsschätzung** gibt es eine eigenständige Präsentation des Autors, die ebenfalls auf der Website unter https://www.peterjohann-consulting.de/_pdf/peco-pm-schaetzen.pdf frei herunterladbar ist.



Ein einfacher Publikationsprozess für eine Spezifikation könnte folgendermaßen aufgebaut sein:

- Freigeben (und Verteiler bestimmen)
- In das pdf-Format konvertieren; eine Version hiervon als pdf-Datei mit Version und Datum im entsprechenden Verzeichnis ablegen
z.B. EP1-Benutzerhandbuch.doc -> EP1-Benutzerhandbuch-V1.00-161220.pdf
- In die entsprechende Publikationsliste eintragen und Projektmitarbeiter sowie den Projektmanager informieren (automatisiert oder persönlich)





Über Nummern können einzelne Status indirekt codiert werden:

- 0.XX (unter Version 0.50): Initial – nur für Projekt-Interne („Ideensammlung“)
- 0.XX (> 0.50 bis 0.99) Draft („Entwurf“) – intern und an ausgewählte Benutzer und Entwickler
- 1.00: Published („Freigegeben“)
- > 1.00: entweder
 - a) Rejected („Zurückgezogen“)
 - b) Draft („in Überarbeitung“)
 - c) erneut Published
- Nach Abschluss des Projekts: entweder
 - a) Deleted (wenn sie keine Relevanz mehr hat – dürfte bei Spezifikationen nicht auftreten) oder
 - b) Archived (in nicht mehr veränderbarer Form mit abschließender Versionsnummer)



Folgende Aufgaben sollte der Projektmanager bei dem Management der Spezifikationen wahrnehmen:

- Zusammen- / Gegenhalten aller Spezifikationen (z.B. über Namensvorgaben oder Listen)
- Erstellung des Projekt- und Release-Plans der Spezifikationen in Abhängigkeit der Produktentwicklung
- Koordinierung der Qualitätssicherung (QS) der Spezifikationen
- Koordinierung der QS der Applikationen (mit den erstellten Testhandbüchern)
- Generieren / Erstellen der Benutzerhandbücher
- Koordinierung des Reviews aller Spezifikationen (zu einem definierten Zeitpunkt)

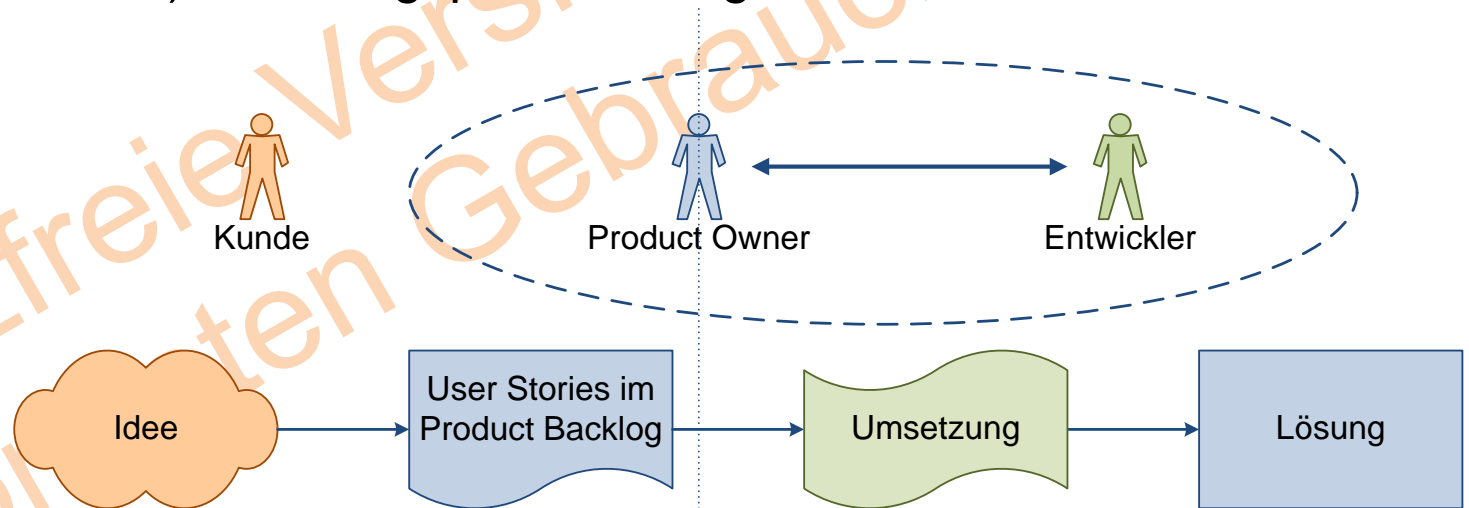


1. Was ist eine Spezifikation?
2. Wer ist für die Spezifikationen im Projekt verantwortlich?
3. Welche Arten von Spezifikation kennen Sie?
4. Wer schätzt den Aufwand zur Umsetzung der Anforderungen?
Wann findet die Schätzung statt?

Lizenzfreie Verwendung
den privaten Gebrauch! © 2025

In dem agilen Kontext gibt keine „klassischen“ Spezifikationen, die in einem Prozess erstellt werden müssten. Stattdessen werden Anforderungen typischerweise über User Stories erfasst und dann umgesetzt.

Sind aber bereits Spezifikationen vorhanden, so können diese über User Stories in den (Software-)Erstellungsprozess eingebracht werden.



Zum **Agilen Requirements Engineering** gibt es eine eigenständige Präsentation des Autors, die ebenfalls auf der Website unter https://www.peterjohann-consulting.de/_pdf/peco-agile-agiles-re.pdf frei herunterladbar ist.

Checkliste: Ist die Spezifikation formal korrekt?



Frage	Ja	Nein	Offen	Maßnahmen
Ist das Deckblatt korrekt ausgefüllt? Produkt- und Projektangaben, Autor, Version, Datum, Freigabeinfo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sind die Angaben in den Kopfzeilen korrekt? Firmenlogo, Titel, Version, Format, Rechtschreibung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sind die Angaben in den Fußzeilen korrekt? Dateiname, Datum, Seitenzahlangaben	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ist die Änderungshistorie korrekt gepflegt?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ist die Rechtschreibung korrekt?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ist das Dokument vollständig bezüglich der Beschreibungen und Bilder?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ist für alle Anforderungen die Identifikation (ID) eindeutig vergeben?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sind für alle Anforderungen der Name, die Beschreibung, die Version und der Status angegeben?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

nach /Grande14/



- Literatur
- Weblinks
- Meine Dienstleistungen – Das kann ich für Sie tun
- Kontakt zum Autor

Anhang

Seite
40–44



- /Balzert08/ Helmut Balzert: Lehrbuch der Softwaretechnik: Softwaremanagement, Spektrum Akademischer Verlag, Heidelberg 2. Auflage 2008, ISBN 978-3-8274-1161-7
- /Balzert09/ Helmut Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, Spektrum Akademischer Verlag, Heidelberg 3. Auflage 2009, ISBN 978-3-8274-1705-3
- /Ebert14/ Christof Ebert: Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren, analysieren und verwalten, dpunkt, Heidelberg 5. Auflage 2014, ISBN 978-3-86490-139-3
- /Grande14/ Marcus Grande: 100 Minuten für Anforderungsmanagement: Kompaktes Wissen nicht nur für Projektleiter und Entwickler, Springer Vieweg, Wiesbaden 2. Auflage 2014, ISBN 978-3-658-06434-1
- /IREB15/ Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt, Heidelberg 4. Auflage 2015, ISBN 978-3-86490-283-3
- /Rupp14/ Chris Rupp: Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil, Hanser, München 6. Auflage 2014, ISBN 978-3-446-43893-4
- /Schienmann01/ Bruno Schienmann: Anforderungsmanagement: Prozesse – Techniken – Werkzeuge, Addison-Wesley, München 2001, ISBN 978-3-8273-1787-2
- /Wiegers13/ Karl E. Wiegers, Joy Beatty: Software Requirements, Microsoft Press, Redmond, Washington 3rd Edition 2013, ISBN 978-0-7356-7966-5



/Wiki-d/ Deutsche Wikipedia: <https://de.wikipedia.org>; eingesehen am 20.12.2016

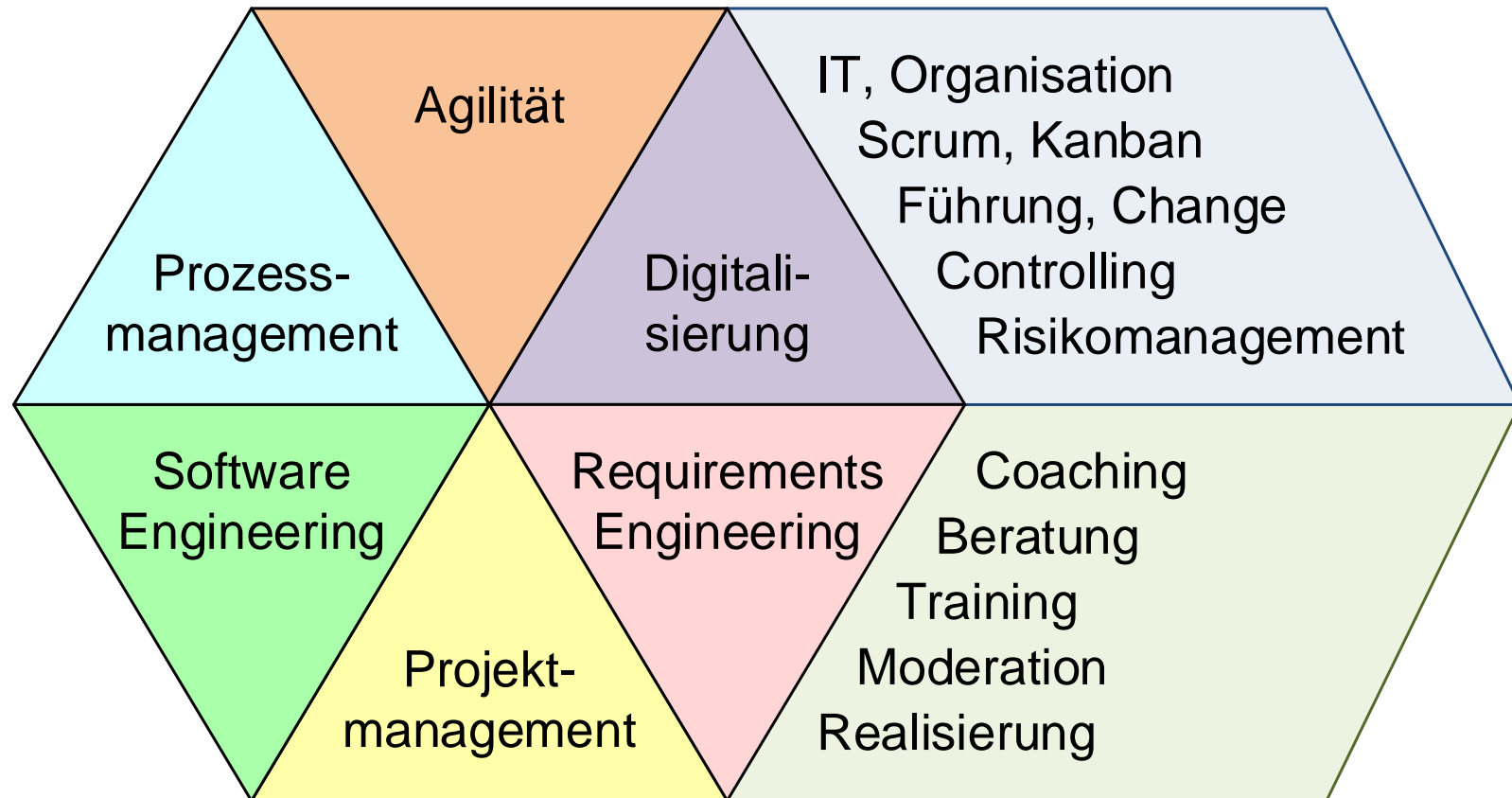
/#Wiki-Spezifikation/ Spezifikation in der deutschen Wikipedia:
<https://de.wikipedia.org/wiki/Spezifikation>; eingesehen am 20.12.2016

Legende:

// Verweis auf Website generell

/*/ Verweis auf eine Website, die als Buch-Ergänzung dient

/#/ Verweis auf einzelnes Thema auf einer Website



Sie benötigen noch weitere Informationen?
Kontaktieren Sie mich!

Peterjohann Consulting

Dipl.-Inform.

Horst Peterjohann

PMP, PMI-PBA, CPRE, CTFL, PSM I, ITILv2

Kattenvenner Straße 24

49549 Ladbergen



Telefon: 0 54 85 / 830 17 29

Mobil: 0 162 / 977 47 65

E-Mail: kontakt@peterjohann-consulting.de

Website: <https://www.peterjohann-consulting.de>