

Requirements Engineering (und Business Analysis)

Eine Einführung (RE-Basispräsentation)
Für Requirements Engineers, Business Analysten,
Product Owner, Projektmanager, Projektmitarbeiter,
System-Ingenieure und Softwareentwickler
Stand: 03/2017

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 0.96 vom 02.03.2017
248 Seiten



Version	Datum	Autor	Beschreibung	Seiten	Verteiler
0.91	03.10.2012	HP	Erste Version nach IREB-Standard	140	Website
0.92	04.11.2013	HP	Überprüfung und Ausbau	200	Website
0.94	10.04.2015	HP	Aktualisierung und Überprüfung der Literatur	216	Website
0.96	02.03.2017	HP	Ausbau (mit Update wegen BABOK V3 und PMI-PBA)	248	Website

Änderungsvorhersage:

Version	Datum	Autor	Beschreibung
1.00	Feb 2021	HP	Weitere Überarbeitung und Ausbau mit Erscheinen des „PMI Guide to Business Analysis“ /PMG-BA17/ und des Lehrplans 3.0.x zum CPRE des IREB sowie der Neuerscheinung des Buchs zum Lehrplan (Nachfolge /IREB15/)



Das Ermitteln, Dokumentieren, Prüfen, Abstimmen und Verwalten von Anforderungen gehört zu den zeitaufwendigsten Aufgaben bei der Entwicklung von (neuen) Produkten oder Dienstleistungen und wird zusammenfassend als **Requirements Engineering** (oder abgekürzt **RE**) bezeichnet.

Was im Einzelnen getan werden muss, um hinreichend gute Anforderungen zu erhalten, wird in der Literatur aus verschiedenen Sichten beschrieben.

Generell hat sich das RE in letzten Jahren etabliert und die in der Literatur zu findenden unterschiedlichen Sprach- und Verständnisweisen stark reduziert.

In dieser Präsentation wird ein neutraler Überblick der wesentlichen Aspekte des Requirements Engineerings (und der Business Analysis) geliefert.

Zuordnung:

Nach dem PMI (Project Management Institute) /PBG12, PBG12-d/ werden Requirements in dem Wissensgebiet (*Knowledge Area*) „*Scope Management*“ behandelt, allerdings dort anders beschrieben als in der RE-Fachwelt üblich. **Diese Ausarbeitung orientiert sich an den Vorgaben des International Requirements Engineering Boards (IREB /IREB15/), das seit 2007 Personen-Zertifizierungen für das RE anbietet.**



- Diese Präsentation ist „für sich“ eigenständig, kann also ohne Sekundärquellen nachvollzogen werden
- Die hier gewählte Darstellung ist zwischen „reiner Präsentation“ und „ausformuliertem Text“ angesiedelt. Daher kann es sein, dass einige Angaben nicht unmittelbar verständlich sind
- Es wird hier eine Übersicht des Requirements Engineerings (REs) und der Business Analysis (BA) geliefert und damit kein Lehrbuch ersetzt
- Es werden bevorzugt deutsche Fachbegriffe verwendet, jedoch werden englische Bezeichnungen dort, wo sie massiv Einzug gehalten haben, auch zusätzlich in kursiver Schrift (*Font*) notiert
- In dieser Präsentation sind Übungen, Fragen, Tipps und Checklisten enthalten, die der Verständnisvertiefung dienen
- Werkzeuge und Methoden (*Tools & Techniques*) werden nur kurz vorgestellt
- Auf Literatur (Bücher) wird im Text in der Form „/Buch17/“ verwiesen; die Literaturliste findet sich im Anhang. Weblinks sind ebenfalls im Anhang enthalten: Sie werden im Text in ähnlicher Form wie Bücher zitiert



Auch wenn diese Präsentation ohne zusätzliche Hilfe verständlich sein sollte – der formale Aufbau wird beschrieben in den ...

Erläuterungen

Sie möchten, dass ich Sie in Ihrem beruflichen Umfeld unterstütze?
Es gibt hierzu eine Übersicht meiner ...

Dienstleistungen

Sie möchten diese Präsentation in Ihrem beruflichen Umfeld einsetzen?
Informationen dazu gibt es hier:

Lizenzierung

Durch Klicken der blauen Schaltflächen gelangen Sie zu weiteren Informationen auf meiner Website.

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

E-Mail: kontakt@peterjohann-consulting.de

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



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

- Sie kennen die Aufgaben des Requirements Engineerings
- Sie können abschätzen, mit welchem Aufwand Sie Requirements Engineering betreiben müssen
- Sie kennen die Hauptaktivitäten des Requirements Engineerings
- Die wichtigsten Werkzeuge und Methoden (*Tools and Techniques*) des Requirements Engineerings sind Ihnen geläufig

Zielgruppe: Requirements Engineers, Business Analysten, Product Owner, Projektmanager, Projektmitarbeiter, System-Ingenieure und Softwareentwickler

Voraussetzungen: Erste Erfahrungen mit Anforderungen

Schwierigkeitsgrad: Gering bis mittel



Im **Teil I** dieser Ausarbeitung werden die Grundlagen des Requirements Engineerings vorgestellt. Hierzu gehören Definitionen und grundlegende Beschreibungen.

Der **Teil II** beschäftigt sich mit dem zunächst Kerngebiet des REs, der Anforderungsentwicklung. Auf der Basis der vom IREB (International Requirements Engineering Board) vorgegebenen vier Hauptaktivitäten werden einige Werkzeuge und Methoden zur Entwicklung vorgestellt. Abgeschlossen wird der zweite Teil mit der Beschreibung der Anforderungsverwaltung.

Teil III beschreibt neben dem Einsatz von RE-Tools auch einige Aspekte bei der Umsetzung von RE in der Praxis. Das Agile Requirements Engineering (ARE) wird ebenfalls (kurz) beschrieben.

Im Anhang (**Teil IV**) finden sich neben einer Literaturliste und Weblinks auch Normen und Standards, die im Requirements Engineering relevant sind.

Teile
I-II

Gliederung

Seite
10–150**Teil I: Grundlagen**

1. Einleitung und Grundlagen 11–31
2. Kategorien und Prozesse 32–44
3. Fragen und Übungen zum Teil I 45–50

Teil II: Der Requirements-Engineering-Prozess nach IREB

4. Anforderungen ermitteln 52–73
5. Anforderungen dokumentieren 74–115
6. Anforderungen prüfen und abstimmen 116–130
7. Anforderungen verwalten 131–146
8. Fragen und Übungen zum Teil II 147–150



Teile
III-IV

Gliederung

Seite
151–248

Teil III: Requirements Engineering in der Praxis

- | | | |
|-----|---|---------|
| 9. | Tools für das Requirements Engineering | 152–159 |
| 10. | Agiles Requirements Engineering | 160–174 |
| 11. | Die Einführung von Requirements Engineering | 175–179 |
| 12. | Weitere Praxis-Aspekte | 180–186 |
| 13. | Fragen und Übungen zum Teil III | 187–188 |

Teil IV: Anhang

- | | | |
|----|---|---------|
| A. | Literatur, Weblinks, Sprüche und Glossar | 190–220 |
| B. | Normen, Standards und Zertifikate | 221–236 |
| C. | Reifegradmodelle und techn. Dokumentation | 237–244 |
| D. | Weitere Präsentationen, Kontakt zum Autor | 245–248 |



Teil I

Seite
10–50

Teil I

- Kapitel 1 Einleitung und Grundlagen
- Kapitel 2 Kategorien und Prozesse
- Kapitel 3 Fragen und Übungen zum Teil I



Teil I

- Anforderungen: Grundsätzliches, Definitionen
- Anforderungen und Informationen
- Anforderungen, Ziele und Visionen
- Definition: Requirements nach IEEE 610.12-1990
- Was ist Requirements Engineering? Definitionen
- Requirements Engineering im Kontext (eines Software-Projekts)
- Die drei Dimensionen des Requirements Engineerings
- Die Stakeholder (Beschreibung, Bedeutung, Unterteilung)
- Der Requirements Engineer (Beschreibung, Die sieben notwendigen Fähigkeiten)
- Requirements Engineering und Business Analysis
- Motivation für Requirements Engineering
- Wenn auf Requirements Engineering verzichtet wird

Kapitel 1

Seite
11–31



Anforderungen ...

- sind das deutsche Wort für „Requirements“.
- werden typischerweise „möglichst früh“ (im Projekt) ermittelt.
- müssen für alle Beteiligten (eines Projekts) zugänglich („zentral abgelegt“) sein.

Anforderungen dienen als Grundlage für ...

- Kommunikation.
- Ausschreibung und Vertragsgestaltung.
- Systemintegration, Wartung und Pflege.
- Systemarchitektur.

Anforderung:
Die Präsentation
muss mit pdf-
Readern gelesen
werden können.

**Anforderung** (engl. *Requirement*):

„Beschaffenheit, Fähigkeit oder Leistung, die ein Produkt, Prozess oder die am Prozess beteiligte Person erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere formell vorgegebene Dokumente zu erfüllen.“ (DIN 69901-5:2009-01)

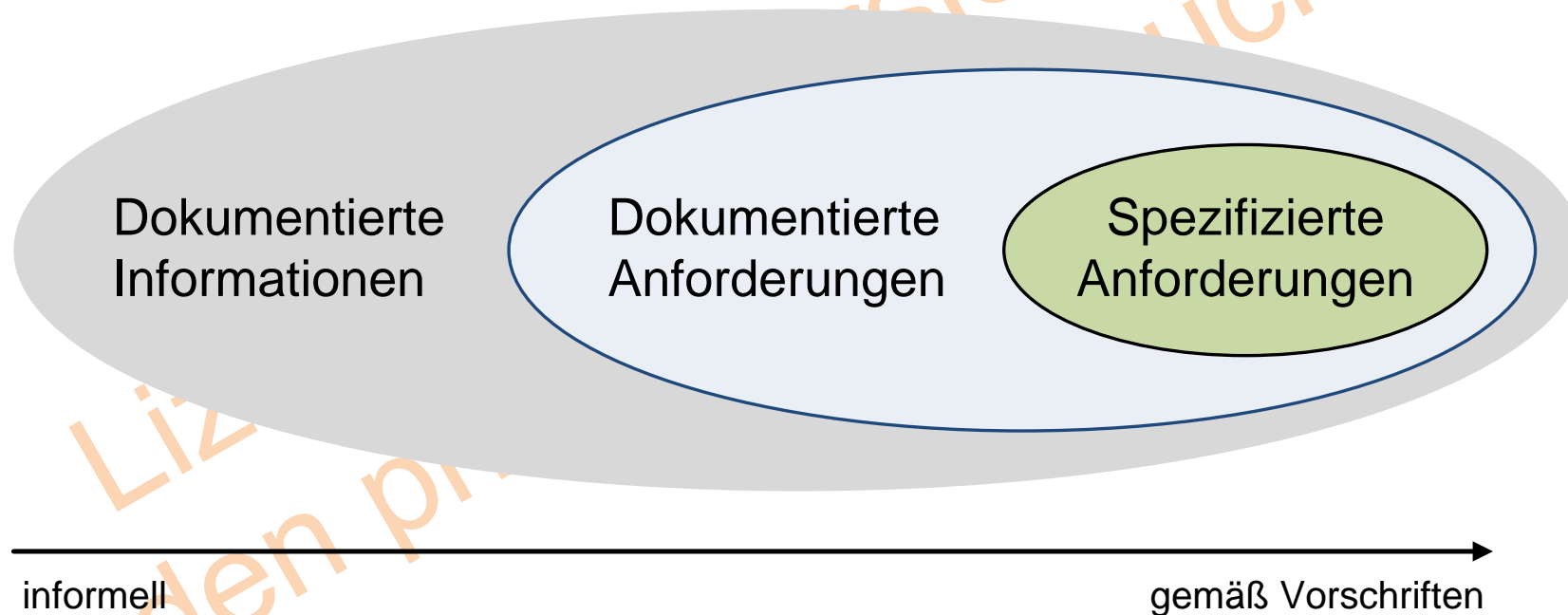
Anforderung im IEEE-Standard 830-1998:

„Eine Anforderung ist eine Bedingung oder eine Fähigkeit, die ein Benutzer benötigt, um ein Problem zu lösen oder um sein Ziel zu erreichen.“

Anforderungen/Requirements sind:

- „Something that the product must do, or a property that the product must have, and that is needed or wanted by the stakeholders.“ /Robertson12/
- „Requirements are a specification of what should be implemented. They are descriptions of how the systems should behave, or of a system property or attribute. They may be a constraint on the development process of the system.“ /Wiegers13/
- „Needs or conditions to be satisfied on behalf of users and suppliers.“ /Got02/

Anforderungen sind immer dokumentierte Informationen. Wenn die Informationen stärker aufbereitet und verdichtet werden (nach gewissen Vorschriften) entstehen daraus spezifizierte Anforderungen.



/Pohl08/



Anforderungen und Ziele sind im Requirements Engineering nicht immer scharf voneinander abgetrennt.

Ebert /Ebert14/ schreibt hierzu:

„**Anforderungen konkretisieren Ziele** und definieren die Basis, auf der eine Lösung zum Kundennutzen entwickelt wird. Ziele und Anforderungen sind nicht absolut wahr!“

Achtung:

Im Projektmanagement werden häufig Ziele den Anforderungen gleichgesetzt. Im Requirements Engineering wird aber unterschieden.

Zu den **Zielen und deren Ermittlung in Projekten** gibt es eine eigenständige Präsentation des Autors, die ebenfalls auf der Website unter https://www.peterjohann-consulting.de/_pdf/peco-pm-ziele.pdf frei herunterladbar ist.



A requirement is /BBG09/ (based on IEEE 610.12-1990):

1. A condition or capability needed by a stakeholder to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a solution or solution component to satisfy a contract, standard, specification, or other formally imposed documents.
3. A documented representation of a condition or capability as in (1) or (2).

Eine Anforderung ist nach IREB (/IREB15/ – Übersetzung der IEEE 610.12-1990):

1. Eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die ein System oder ein Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).



Das IREB (International Requirements Engineering Board) definiert das Requirements Engineering wie folgt /IREB15/:

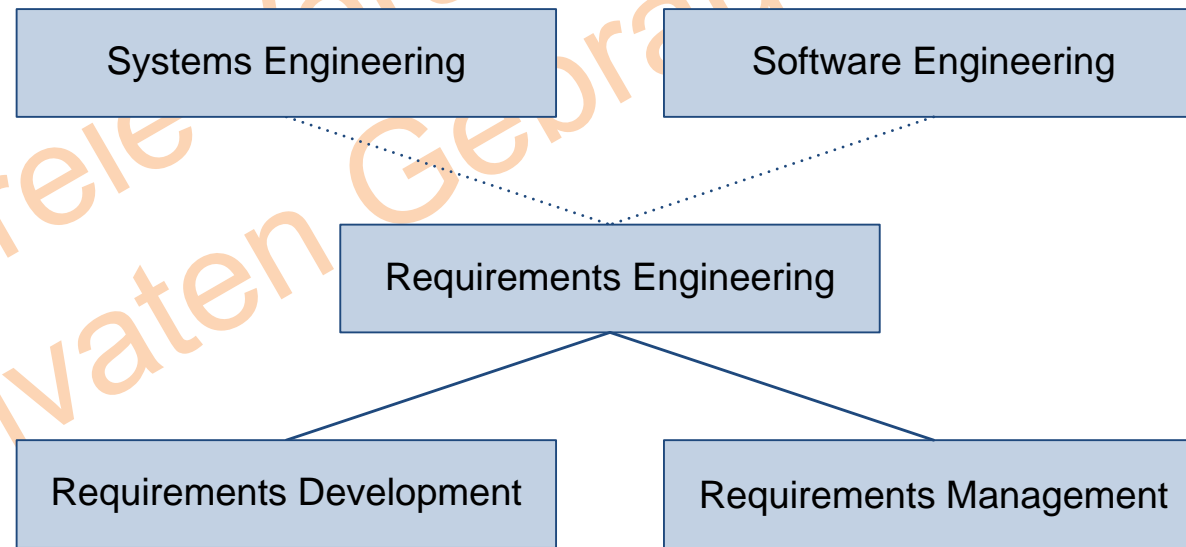
„Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
2. Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht Wünschen und Bedürfnissen der Stakeholder entspricht.“



Wieggers /Wieggers13/ definiert:

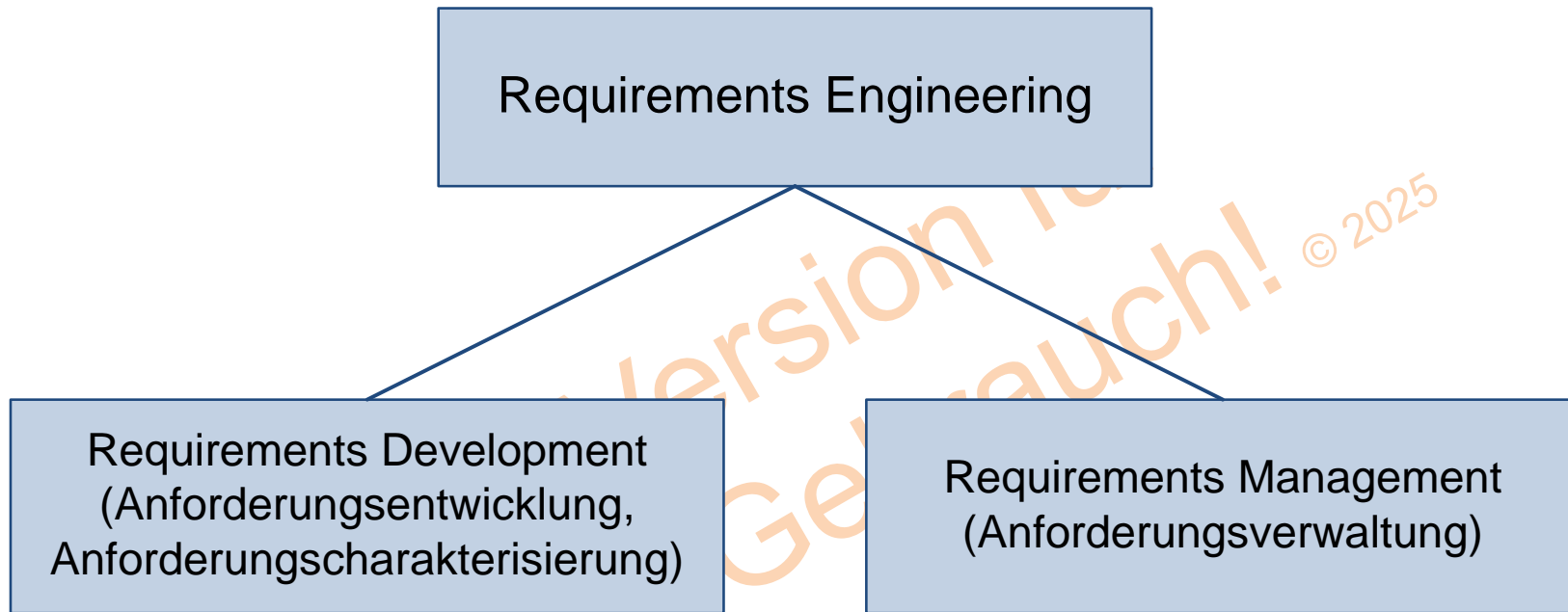
„RE is the subdiscipline of systems engineering and software engineering that encompasses all project activities associated with understanding a product's necessary capabilities and attributes. Includes both requirements development and requirements management.”



..... ist Teil von (is subdiscipline)
— beinhaltet (includes)



- „Das Ziel von RE ist es, qualitativ gute – nicht perfekte – Anforderungen zu generieren, die es erlauben, das Projekt mit einem akzeptablen Risiko zu beginnen.“ /Ebert14/
- RE ist der systematische Ansatz, Anforderungen des Systems zu erheben, zu organisieren und zu dokumentieren und einen Prozess, der eine Übereinkunft zwischen dem Kunden und dem Projektteam liefert, zu etablieren
- Ziel des REs ist es, möglichst früh alle (relevanten) Anforderungen zu identifizieren, zu erfassen und zu beschreiben
- „Der Zweck des REs ist es, ein Einverständnis zwischen dem Kunden und dem (Software-)Projekt (also dem Lieferanten) über jene Anforderungen zu erreichen, die durch das Softwareprojekte abgedeckt werden.“ /Ebert12/
- „RE ist die Disziplin, die Bedürfnisse auf Lösungen abbildet.“ /Ebert12/



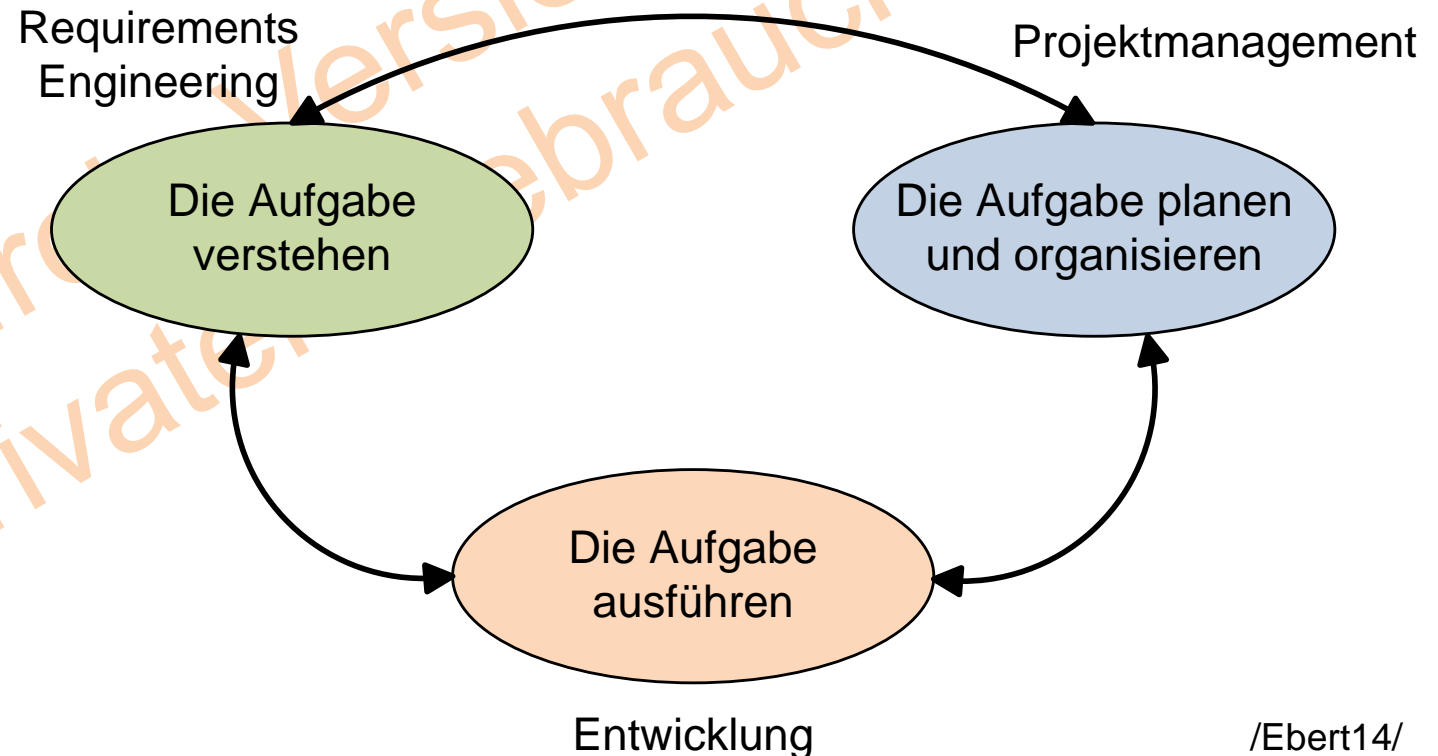
auch: Requirements Analysis

Bitte beachten Sie:

- Requirements Engineering (RE) wird vielfach dem Begriff Requirements Management (RM) gleichgesetzt; hier ist das RE dem RM übergeordnet
- Die Schreibweise Requirements-Engineering (mit Bindestrich) wird hier nicht verwendet



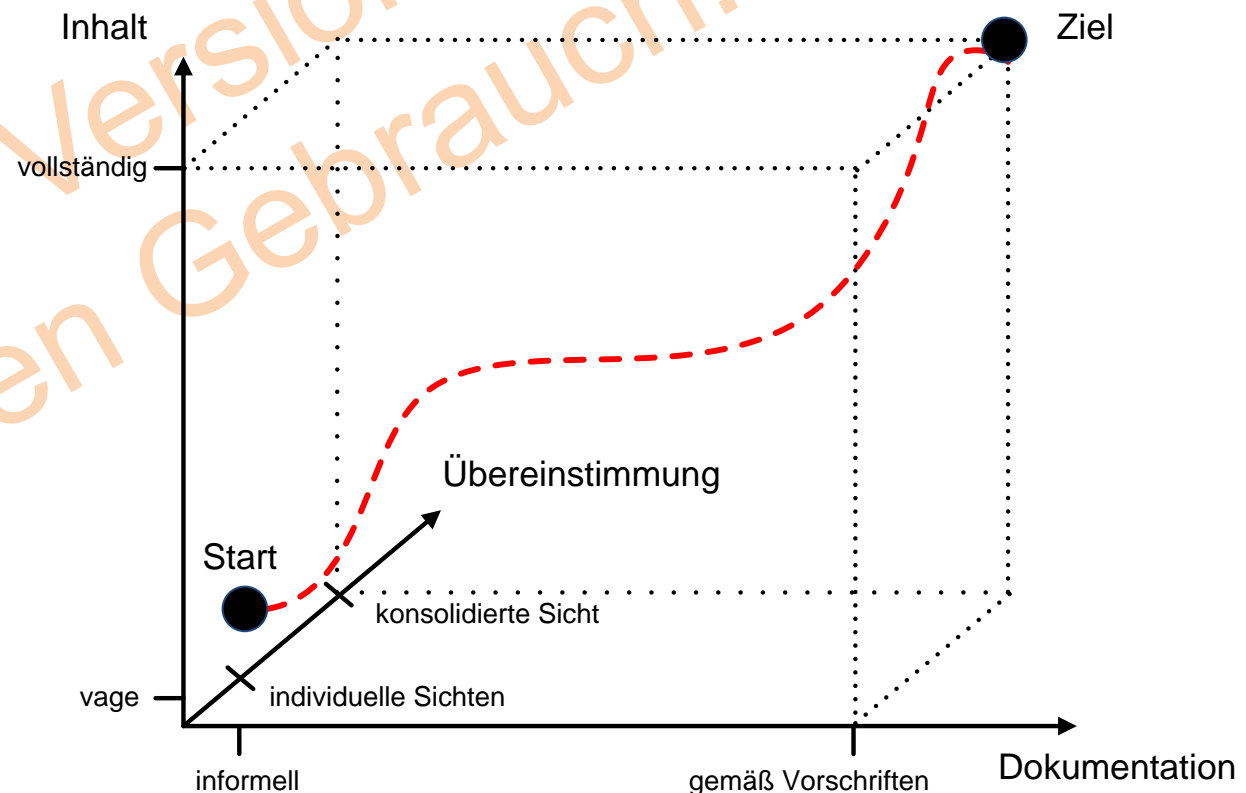
Das Requirements Engineering beschäftigt sich damit, relevante Aufgaben zu erfassen und zu beschreiben. Die Umsetzung dieser Aufgaben erfolgt durch ein „Entwicklungs-Team“, welches wiederum – genau wie das „Requirements-Team“ – durch das Projektmanagement gesteuert wird. Alle drei Disziplinen greifen ineinander.





Beim (klassischen) Requirements Engineering wird versucht, den Inhalt (des zu beschreibenden Gegenstands oder Systems) vollständig zu erfassen, dabei die Sichten der unterschiedlichen Stakeholder zu vereinheitlichen und eine nachvollziehbare Dokumentation zu erstellen.

Der „Weg des RE-Wissens“ ist hier als rot gestrichelte Linie eingezeichnet.



/Pohl08/



Das IREB (International Requirements Engineering Board) definiert Stakeholder folgendermaßen /IREB15/:

„Ein Stakeholder ist eine Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen hat.

Beispiele für Stakeholder sind Nutzer des Systems, Betreiber des Systems, Entwickler, Architekten, Auftraggeber und Tester.“

Stakeholder können sein:

Der Fachanwender oder die Fachabteilung, der Kunde, das Management, der Vertrieb, das Marketing, der Projektmanager / -leiter, der Projektmitarbeiter, die Schnittstellen (!), ...

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

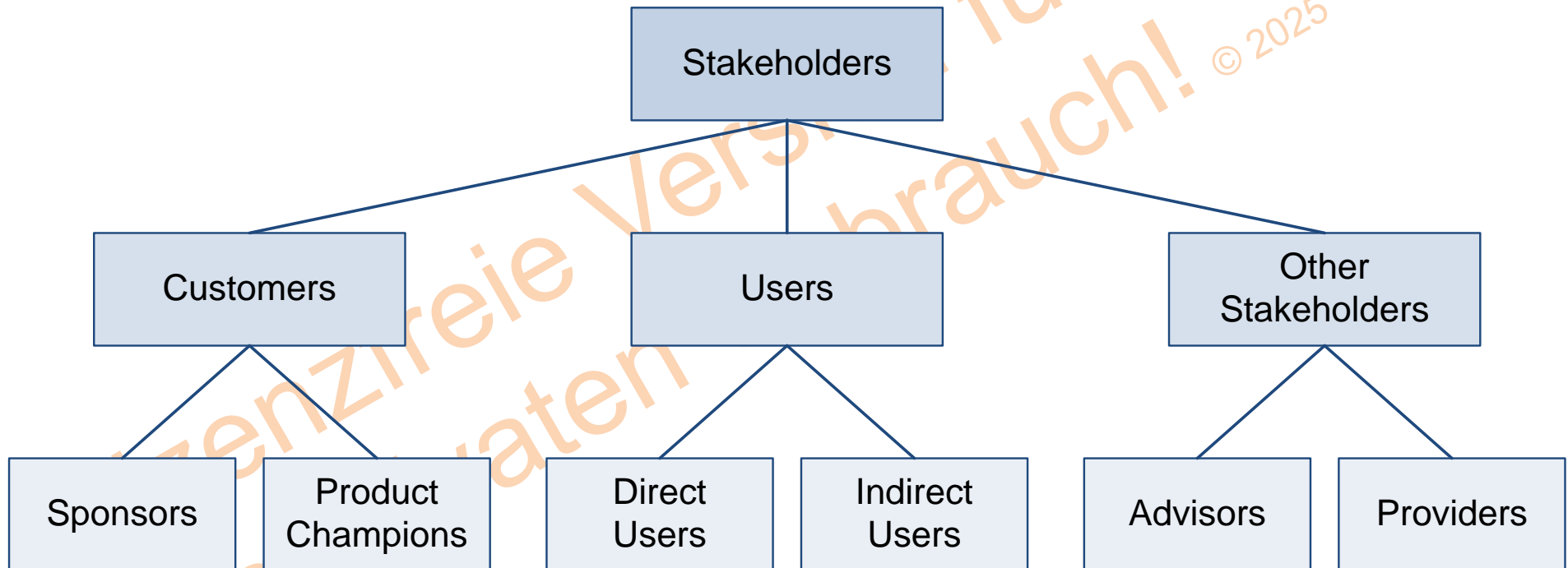


Stakeholder werden im Requirements Engineering (in einem Projekt) immer besonders betrachtet, da sie das Projekt stützen müssen und die Anforderungen kennen. Daher gibt es eine eigenständige Disziplin für die Behandlung der Stakeholder – das Stakeholdermanagement.

Typische Werkzeuge und Methoden für das Stakeholdermanagement sind:

- Stakeholderlisten
- Kraftfeldanalysen
- Wirkungsanalysen
- (Kommunikationsmatrizen)
- (Risikomatrizen)

Gottesdiener /Got05/ unterteilt die Stakeholder in die drei Kategorien „Kunden“, „Anwender“ und „Andere“, die wiederum Unterkategorien haben.



Je nach Bedeutung und Art des Einflusses werden dann die Stakeholder im Stakeholdermanagement behandelt.



„Der Requirements Engineer als Projektrolle steht häufig im Mittelpunkt des Geschehens. Er pflegt in der Regel als Einziger direkten Kontakt zu allen Stakeholdern und hat die Chance und Verantwortung, sich ausreichend in das Fachgebiet der Stakeholder einzuarbeiten sowie die Sprache in den jeweiligen Fachgebieten zu erlernen und zu verstehen.“ /IREB15/

Der Requirements Engineer ist derjenige, der das zu erstellende Produkt, System oder die Dienstleistung beschreibt.

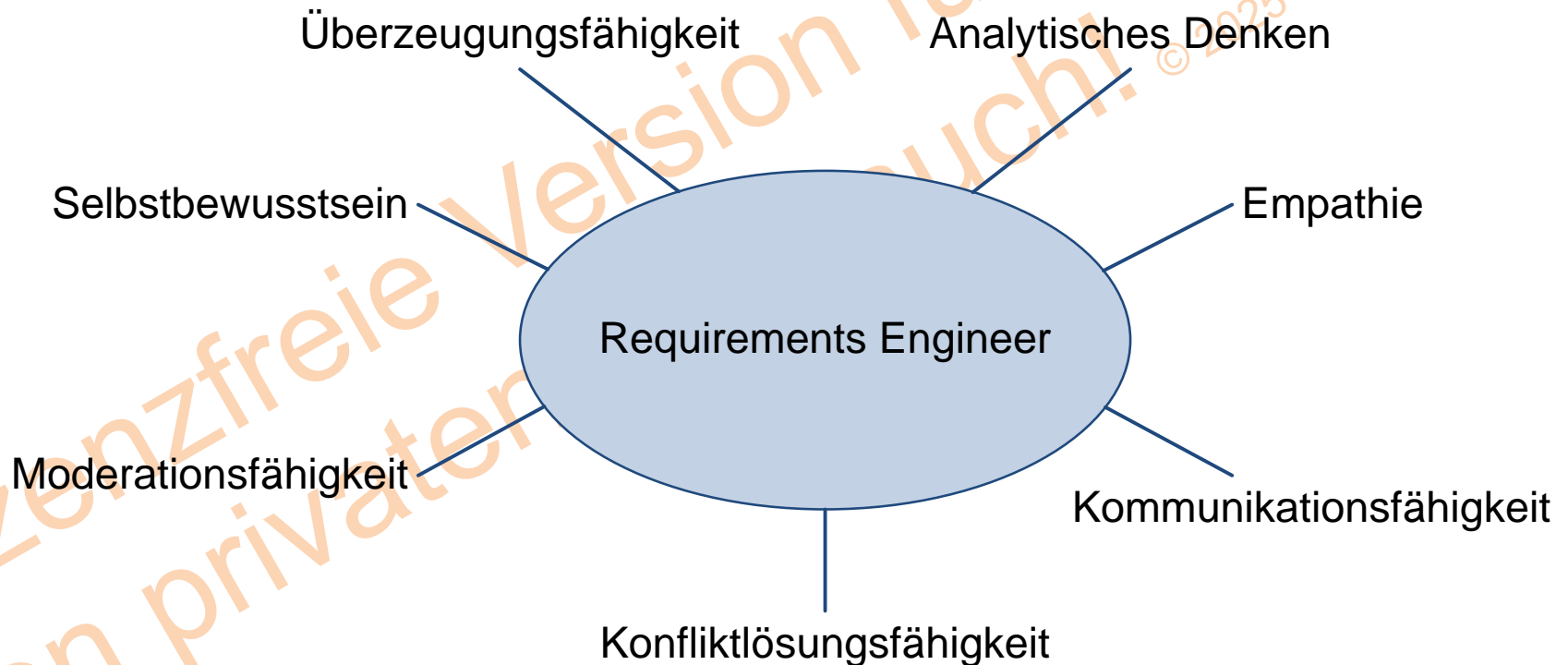
Andere Bezeichnungen:

Requirements-Ingenieur, Business Analyst, Anforderungsingenieur, Systemanalytiker, System Analyst, Requirements Analyst, ...

„Der Requirements-Ingenieur ist das Bindeglied zwischen Kunden, Benutzer, Marketing/Vertrieb, Produktmanagement und der Entwicklung. Er ist für die Ermittlung und adäquate Dokumentation der Kundenbedürfnisse und der daraus abgeleiteten Markt-, Produkt- und Kundenanforderungen zuständig.“ /Ebert12/



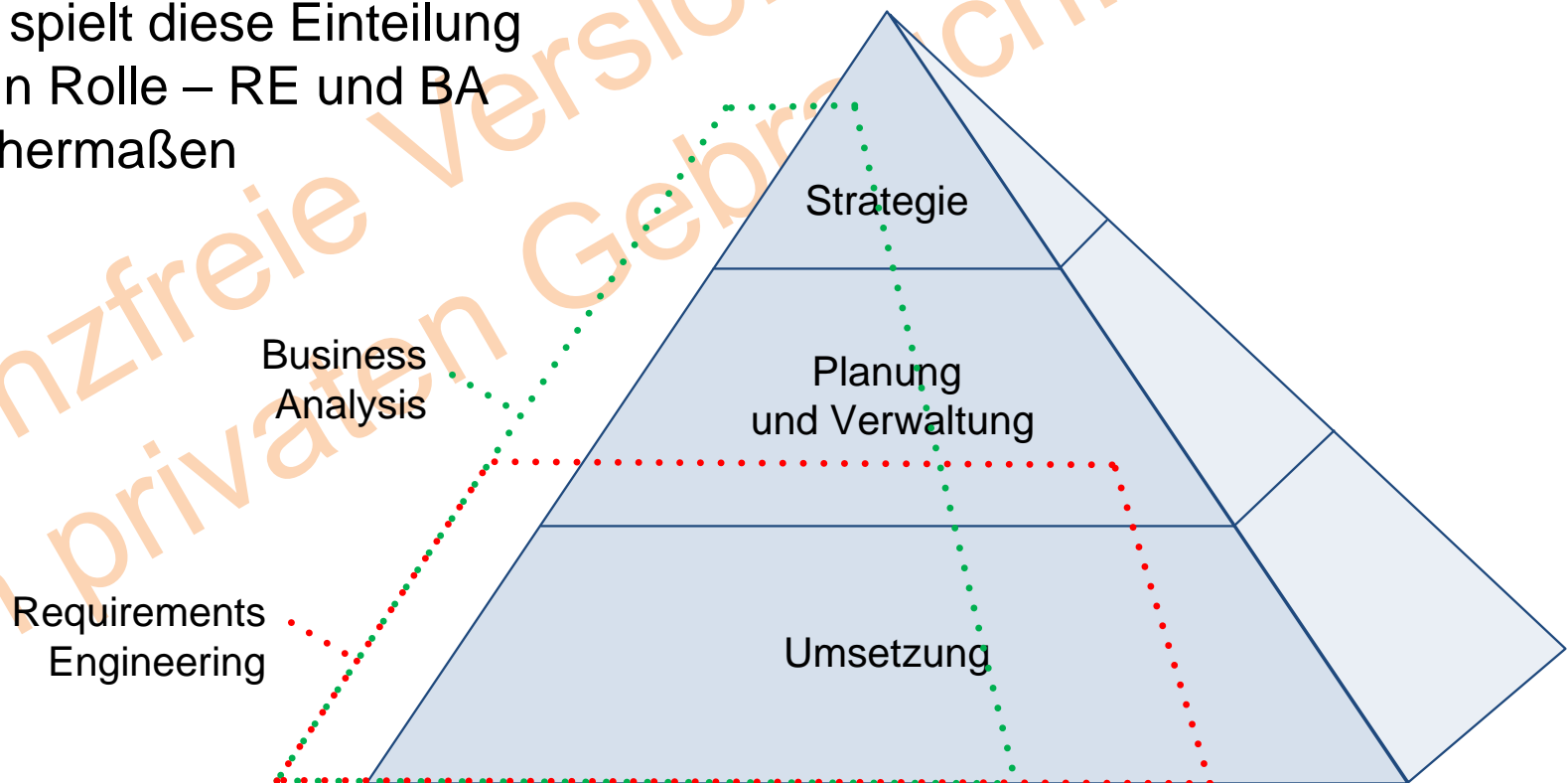
Das IREB benennt sieben Fähigkeiten, die ein Requirements Engineer neben dem RE-Methodenwissen besitzen sollte /IREB15/.



Diese Fähigkeiten decken sich stark mit denen, die von einem Projektmanager (Projektleiter) gefordert werden.

Requirements Engineering (RE) und Business Analysis (BA) haben einen leicht unterschiedlichen Fokus: Während das RE den Schwerpunkt mehr auf die technischen Umsetzung der Anforderungen legt, betrachtet die BA verstärkt die Ableitung der Strategien in umsetzbare Lösungen.

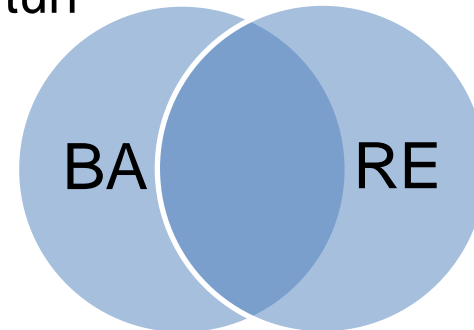
In der Praxis spielt diese Einteilung aber kaum eine Rolle – RE und BA werden gleichermaßen benutzt.



Zusammenfassend wird hier die Abkürzung „BA&RE“ für „Requirements Engineering und Business Analysis“ verwendet. In der deutschen Literatur findet sich meistens der Begriff „Requirements Engineering“ während „Business Analysis“ häufiger im englischen Sprachraum zu finden ist.

Anmerkungen:

- Hier wird durchgängig der Begriff „Business Analysis“ benutzt und auf die Schreib- und Sprechweisen „Business Analyse“ oder „Business-Analysis“ verzichtet
- „Business Analysis“ ist nicht zu verwechseln mit „Business Analytics“
- „Business Analysis“ hat nur indirekt mit dem „Business Process Management (BPM) / Prozessmanagement“ zu tun





Als Gründe für die Durchführung des REs können genannt werden:

- Produktivitätsverbesserung
- Verbesserte Projektplanung und Ressourceneinteilung
- Bessere Beherrschung der Projektrisiken
- Kürze Durchlaufzeiten
- Weniger Nacharbeiten
- Wiederverwendung von Anforderungen
- Bessere Kundenzufriedenheit
- Insgesamt: Mehr Geld verdienen (durch geringere Herstellungs- und Pflegekosten)

Die wirtschaftliche Wirkung von Requirements Engineering ist immer indirekt; das RE selbst kostet nur!



Mangelhaftes oder fehlendes Requirements Engineering (in Projekten) erkennt man an folgenden Symptomen:

- Es fehlen Anforderungen
- Anforderungen sind lückenhaft
- Anforderungen sind unklar beschrieben / formuliert
- Die RE-Unterlagen sind nicht vorhanden oder werden nicht gepflegt
- Es treten Kommunikationsprobleme zwischen den Beteiligten auf
- Hohe Fehlerrate in allen Phasen des Projekts
- Insgesamt: Es kommt zu Verzögerungen und Verteuerungen im Projektablauf

Daumenzahl:

60 % der Fehler bei „üblichen“ IT-Projekten entstehen in der Analysephase!



Teil I

- Arten von Anforderungen
- Die Unterteilung der Anforderungsarten nach IREB
- Die Unterteilung der Anforderungsarten nach PMI-PBA
- Requirements Engineering und Vorgehensmodelle
- Die Anforderungsentwicklung und -verwaltung nach IREB (mit den vier Hauptaktivitäten)
- Die vier Hauptaktivitäten des Requirements Engineerings nach IREB
- Der Ablauf der Anforderungsentwicklung nach Wiegers (Abfolge, Vergleich mit IREB)
- Unterschiedliche Bezeichnungenweisen
- Hauptprobleme der Anforderungsentwicklung

Kapitel 2

Seite
32–44



Anforderungen können auf unterschiedliche Weise klassifiziert/unterschieden werden; Beispiele sind:

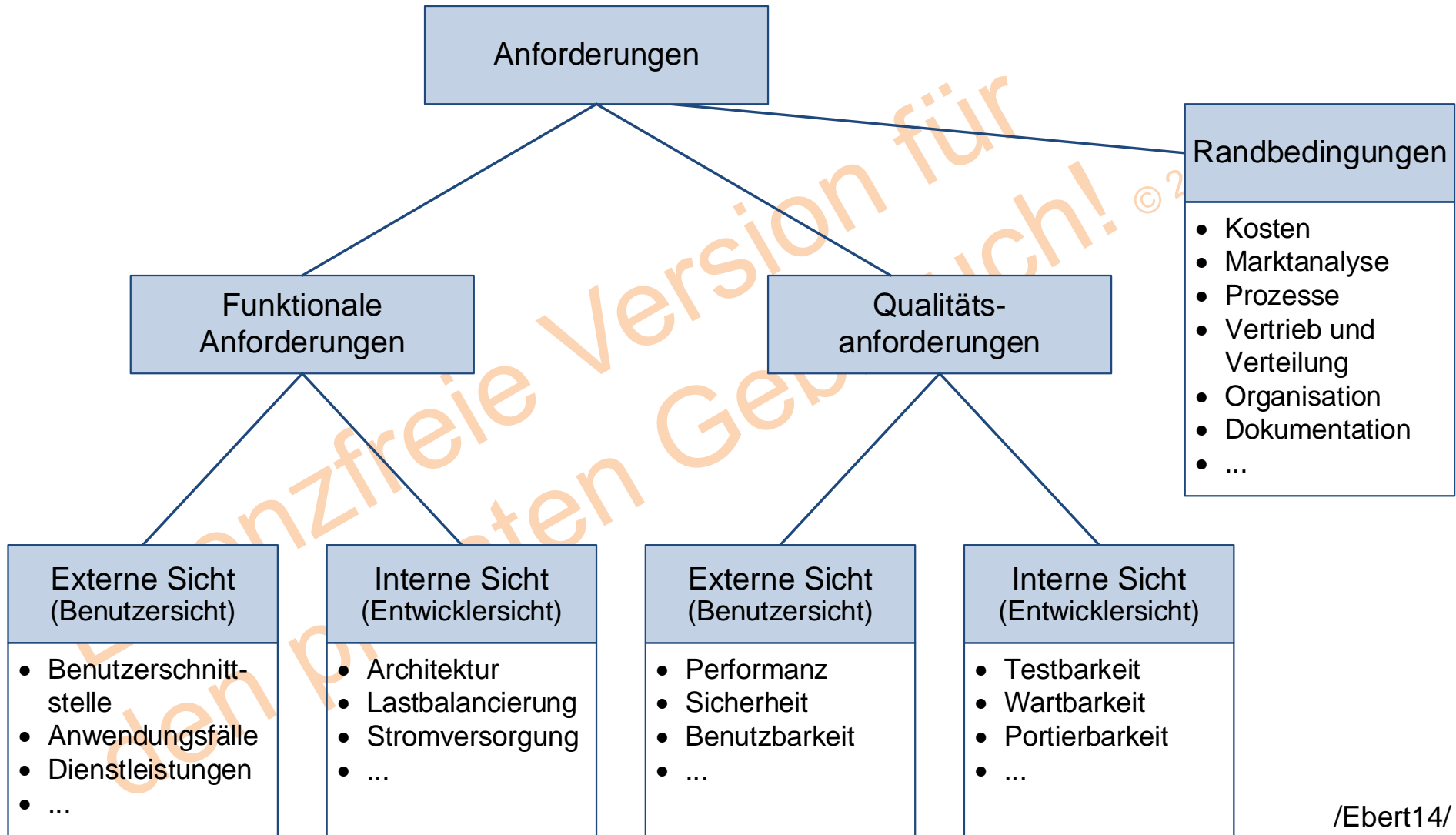
- **Funktionale Anforderungen**
- **Nicht-funktionale Anforderungen**
- Technische Anforderungen
- Anforderungen an die Benutzerschnittstelle
- Qualitätsanforderungen
- Anforderungen an sonstige Lieferbestandteile
- Anforderungen an die Durchführung der Entwicklung
- Rechtlich-vertragliche Anforderungen
- Anforderungen an Personen/Mitarbeiter
- Kunden- oder Geschäftsanforderungen
- Lösungs-, Marketing- oder Systemanforderungen
- Produkt- oder Komponentenanforderungen
- ...

Die Unterscheidung zwischen funktionalen und nicht-funktionalen Anforderungen ist wesentlich!



Das IREB unterteilt die Anforderungen in die drei Arten „funktionale Anforderungen“, „Qualitätsanforderungen“ und „Randbedingungen“ /IREB15/, wobei die beiden letzten Arten auch als „nicht-funktionale Anforderungen“ bezeichnet werden können.

Art	Beschreibung	
Funktionale Anforderung (<i>Functional requirement</i>)	Eine funktionale Anforderung ist eine Anforderung bezüglich des Ergebnisses eines Verhaltens, das von einer Funktion des Systems bereitgestellt werden soll	
Qualitätsanforderung (<i>Quality requirement</i>)	Ein Qualitätsanforderung ist eine Anforderung, die sich auf ein Qualitätsmerkmal bezieht, das nicht durch funktionale Anforderungen abgedeckt wird	Werden häufig auch als „nicht-funktionale Anforderung“ zusammengefasst
Randbedingung (<i>Constraint</i>)	Eine Randbedingung ist eine Anforderung, die den Lösungsraum jenseits dessen einschränkt, was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen	



/Ebert14/



Das PMI unterteilt die Anforderungen in sechs Arten von Requirements („Six Types of Requirements“) /BAPG15/:

- Business requirements („Geschäftsanforderungen“)
- Stakeholder requirements („Anforderungen der Stakeholder“)
- Solution requirements („Lösungsanforderungen“)
 - Functional requirements („Funktionale Lösungsanforderungen“)
 - Non-functional requirements („Nicht-funktionale Lösungsanforderungen“)
- Transition requirements („Übergangsanforderungen“)
- Project requirements („Projektanforderungen“)
- Quality requirements („Qualitätsanforderungen“)



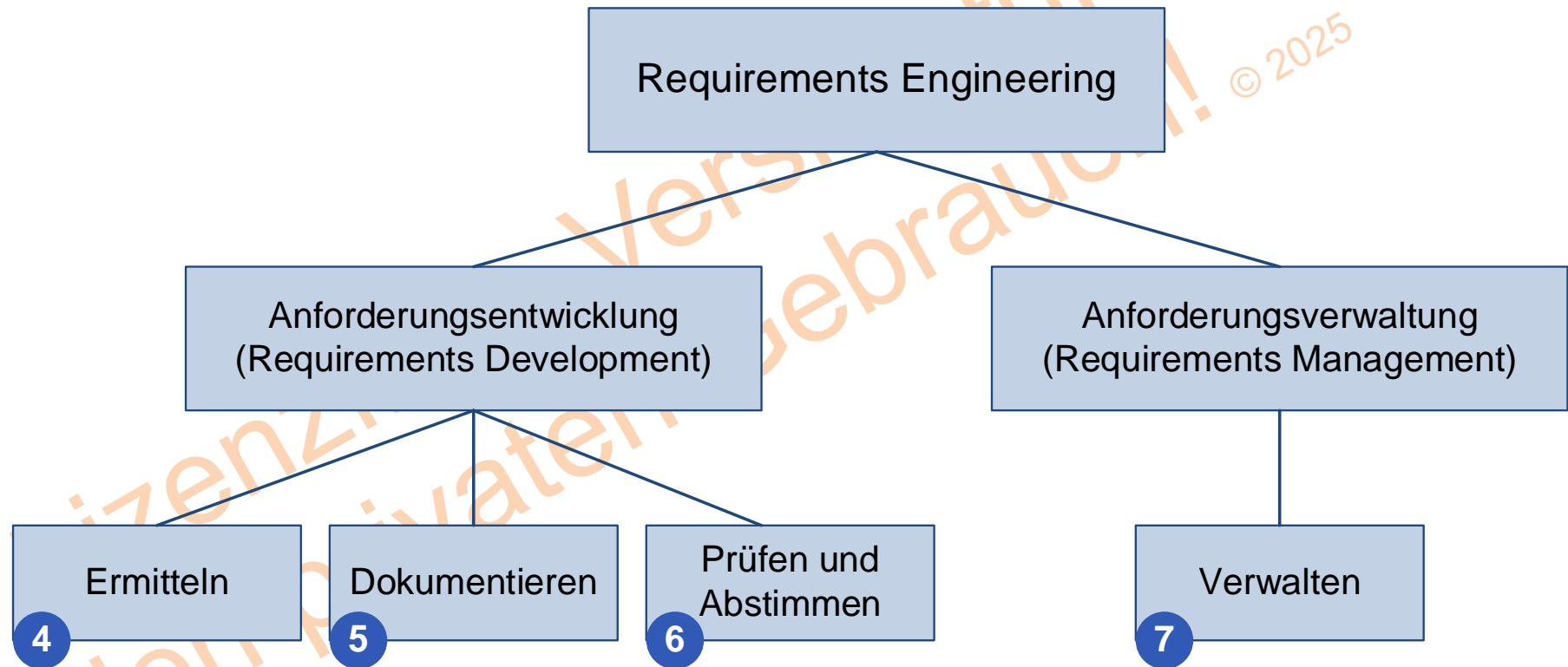
Das Requirements Engineering schreibt kein Vorgehen zur Entwicklung und Verwaltung von Requirements vor. Je nach Einsatzgebiet und Vorgaben werden daher entsprechende spezifische Modelle verwendet.

„Ein Vorgehensmodell organisiert einen Prozess der gestaltenden Produktion in verschiedene, strukturierte Phasen, denen wiederum entsprechende Methoden und Techniken der Organisation zugeordnet sind. Aufgabe eines Vorgehensmodells ist es, die allgemein in einem Gestaltungsprozess auftretenden Aufgabenstellungen und Aktivitäten in einer sinnfälligen logischen Ordnung darzustellen.“ /Wiki-d/

Bekannte Vorgehensmodelle im RE-Kontext sind:

- Das Wasserfallmodell
- Das V-Modell XT
- Volere
- Der Rational Unified Process (RUP)
- Agile Ansätze (Scrum, Extreme Programming = XP)

Das Requirements Engineering ist nach IREB /IREB15/ folgendermaßen aufgeteilt:



Die Zahl im blauen Kreis gibt jeweils an, in welchem Kapitel dieser Ausarbeitung die jeweilige Hauptaktivität beschrieben wird.



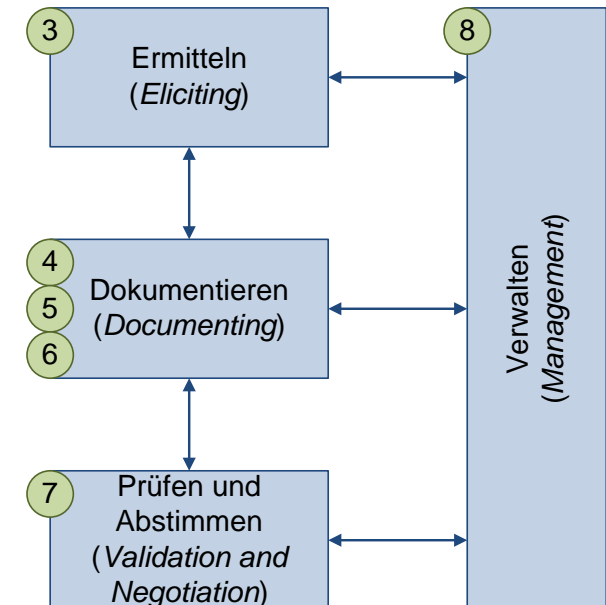
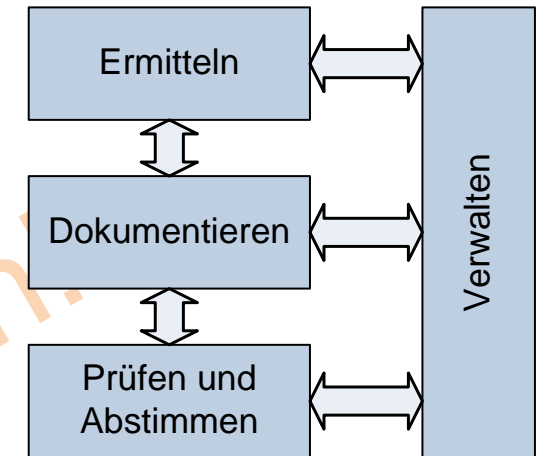
Nach dem IREB /IREB15/ umfasst das Requirements Engineering folgende vier Hauptaktivitäten:

1. Ermitteln
2. Dokumentieren
3. Prüfen und Abstimmen
4. Verwalten

Die vier Hauptaktivitäten werden nicht unbedingt sequentiell durchgeführt, sondern Rücksprünge sind üblich (und erlaubt).

Eine ausführliche Beschreibung der vier Hauptaktivitäten erfolgt im zweiten Teil dieser Ausarbeitung.

Die hier in der unteren Grafik eingetragenen Zahlen im grünen Kreis geben die Kapitelnummern im IREB-Buch /IREB15/ an.





Tätigkeit	Beschreibung	
Ermitteln (<i>Elicitation</i>)	Beim Ermitteln der Anforderungen werden verschiedene Techniken genutzt, um die Anforderungen der Stakeholder und anderer Quellen zu gewinnen, zu detaillieren und zu verfeinern	Requirements Development (Anforderungsentwicklung)
Dokumentieren (<i>Documentation</i>)	Durch die Dokumentation werden erarbeitete Anforderungen adäquat beschrieben. Hierfür können unterschiedliche Techniken eingesetzt werden, um Anforderungen in natürlicher Sprache oder in Modellen zu dokumentieren	
Prüfen und Abstimmen (<i>Validation and Negotiation</i>)	Dokumentierte Anforderungen müssen frühzeitig geprüft und abgestimmt werden, um zu gewährleisten, dass sie allen geforderten Qualitätskriterien genügen	
Verwalten (<i>Management</i>)	Die Anforderungsverwaltung (Requirements Management) geschieht flankierend zu allen anderen Aktivitäten und umfasst alle Maßnahmen, die notwendig sind, um Anforderungen zu strukturieren, für unterschiedliche Rollen aufzubereiten sowie konsistent zu ändern und umzusetzen	Requirements Management



Die Anforderungsentwicklung (*Requirements Development*) wird in der englischsprachigen Literatur meist als vierstufiger Prozess dargestellt, der wie der dreistufige IREB-Prozess iterativ durchlaufen werden kann.



/Wiegers13/

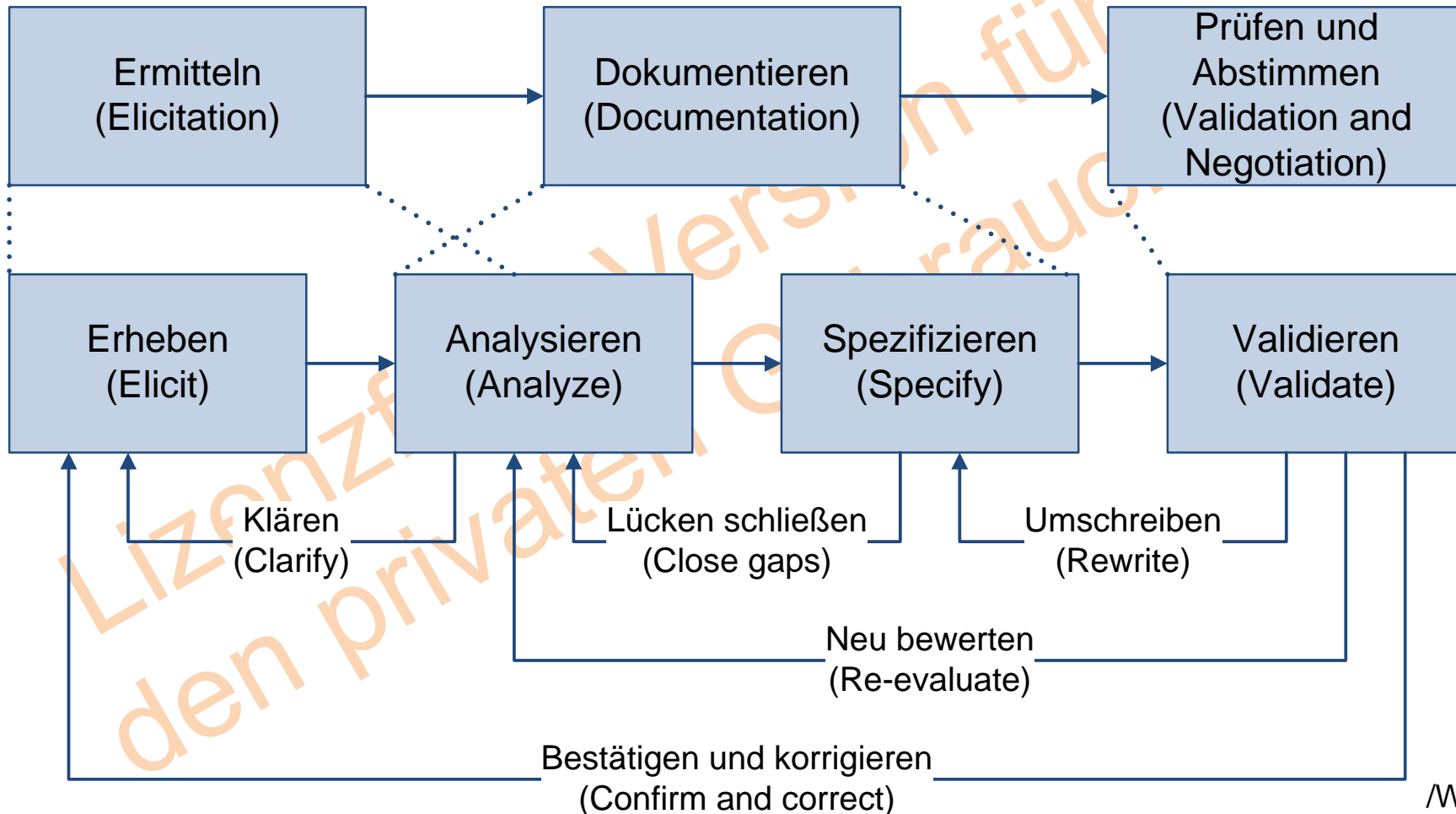
Diese Darstellung ist bedeutsam, da sich die Software-Tool-Hersteller häufig an diesen Bezeichnungsweisen orientieren.

Auch in beim CMMI heißt es:

“The purpose of Requirements Development (RD) is to elicit, analyze, and establish customer, product, and product component requirements.” /Chrissis11a/

Auf der nächsten Folie werden die Prozesse nach IREB (oben) und Wiegers (mit den iterativen Einzelaktivitäten) gegenübergestellt.

Eine Zuordnung der Entwicklungstätigkeiten könnte in etwa so aussehen:



/Wiegers13/



Die deutschen Begriffe „Ermittlung“, „Erhebung“ und „Entwicklung“ können leicht durcheinander gebracht werden:

- „Ermittlung“ und „Erhebung“ sind synonym zu gebrauchen, während
- „Entwicklung“ mehrere Tätigkeiten umfasst (wie z.B. die Ermittlung/Erhebung).

Die englischen Begriffe „Elicitation“ (Ermittlung), „Gathering“ (Erhebung) sowie „Development“ (Entwicklung) sind besser unterscheidbar.



Bei Anforderungsentwicklung treten oftmals Probleme auf, wobei folgende Probleme sehr oft als wesentlich genannt werden:

- Unklare Zielvorstellungen
- Hohe Komplexität
- Sprachbarrieren
- Veränderliche Anforderungen
- Schlechte Qualität
- Unnötige Merkmale
- Ungenaue Planung

Anmerkungen:

- Das sind im Wesentlichen die gleichen Probleme wie beim Projektmanagement!
- RE „löst“ viele Probleme des Projektmanagements

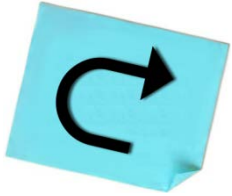


Teil I

- Übung 1: Anforderungen an ein neues Produkt
- Übung 2: Stakeholdermanagement
- Checkliste: Machen Sie „richtiges RE“ in Ihrer Organisation
- Merkgeregeln zum Requirements Engineering
- Fragen zum Teil I

Kapitel 3

Seite
45–50



Stellen Sie die Anforderungen an ein (neues) Produkt zusammen.

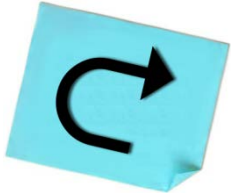
Alternativ:

Stellen Sie Anforderungen an eine (neu zu entwickelnde) Kaffeemaschine zusammen, die Sie gerne hätten.

Dauer:
30 Min.

- Was sind die funktionalen und was sind die nicht-funktionalen Merkmale Ihres Produkts?

Keine
Muster-
lösung!



Erstellen Sie eine Liste der Stakeholder in Ihrem Projekt.

- Welche Stakeholder sind bei Ihnen besonders kritisch und müssen daher gesondert beachtet und behandelt werden?

Dauer:
30 Min.

Keine
Muster-
lösung!

Checkliste: Machen Sie „richtiges RE“ in Ihrer Organisation?



Frage	Ja	Nein	Offen	Maßnahmen
Erfassen Sie alle wesentlichen Anforderungen (in Ihren Projekten)?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Haben Sie eine (einheitliche) Vorgehensweise bei der Entwicklung und Verwaltung von Anforderungen?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Können Sie Anforderungen aus anderen / alten Projekten wiederverwenden?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ist den Mitarbeitern die Bedeutung des REs klar?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sind die (involvierten) Mitarbeiter im RE geschult?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Kennen die Mitarbeiter die Grundbegriffe des REs?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Weiß „das Management“, warum RE gemacht werden muss – und was es kostet?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Werden die Anforderungen in der „passenden Zeit“ (Aufwand und Dauer) entwickelt?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	



- Der größte – vertragsrelevante – Fehler des REs ist es, dem Kunden zu geben, was er braucht und nicht was er will
- ABC-Regel des REs: „Allocation before Commitment“ – Zuweisung vor Zustimmung
- Eine einzelne Anforderung sollte auf einer Seite beschrieben werden
- Unzureichendes RE hat überproportional hohe Folgekosten
- RE benötigt 10 % des Projektaufwands
- Die Requirements-Phase vor Projekt-Start beansprucht 10-50 % der Entwicklungsdauer
- Anforderungen ändern sich im laufenden Projekt typischerweise mit 1-5 % des Projektumfangs (Aufwand) pro Monat
- Die Produktivität eines Softwareentwicklungsprojekts hängt von der Änderungsrate der Anforderungen ab
- Softwareentwicklung folgt dem Pareto-Prinzip
- Änderungen nach der Übergabe betragen: 5-8 % neue Funktionen sowie 10 % geänderte Funktionen pro Jahr

/Ebert12/



1. Was sind Anforderungen?
2. Erklären Sie die Aufgaben des Requirements Engineerings!
3. Wie unterscheiden sich Business Analysis und Requirements Engineering?
4. Was grenzt die Systemgrenze ab?
5. Was sind funktionale und nicht-funktionale Anforderungen? Worin unterscheiden sie sich?
6. Was sind Stakeholder?
7. Wer führt das Requirements Engineering durch?
8. Was sind die vier Haupttätigkeiten des Requirements Engineerings nach IREB?



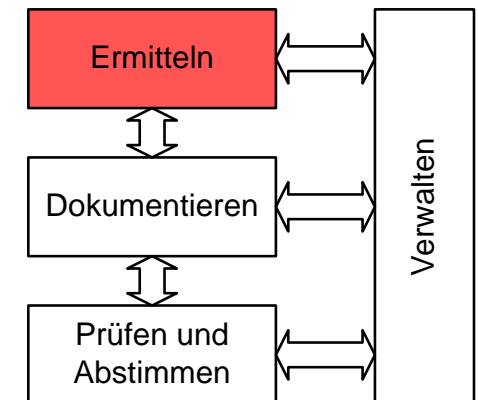
Teil II

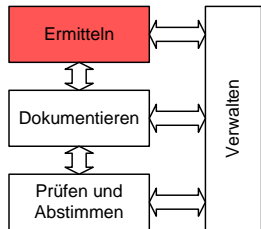
- Kapitel 4 Anforderungen ermitteln
- Kapitel 5 Anforderungen dokumentieren
- Kapitel 6 Anforderungen prüfen und abstimmen
- Kapitel 7 Anforderungen verwalten
- Kapitel 8 Fragen und Übungen zum Teil II

Teil II

Kapitel 4

- Quellen der Anforderungen
- Der Ermittlungsprozess nach Robertson
- Die W-Fragen für die Anforderungsermittlung
- System und Systemkontext (Grundsätzliches, Begriffe)
- Das Kano-Modell (Grundlagen, Diagrammdarstellung, Erläuterung, Beispiele)
- Ermittlungs- und Sammlungstechniken (Grundsätzliches, Übersicht, Die vier Arten der Ermittlungstechniken nach IREB, Wann kann was eingesetzt werden?)
- Ausgewählte Ermittlungstechniken (Interviews, Feldbeobachtung, Apprenticing)
- Anmerkungen und Tipps zu den Ermittlungstechniken
- Das Glossar (Grundsätzliches, Was gehört hinein? Inhalte, Möglicher Aufbau, Regeln für den Umgang)

Seite
52–73

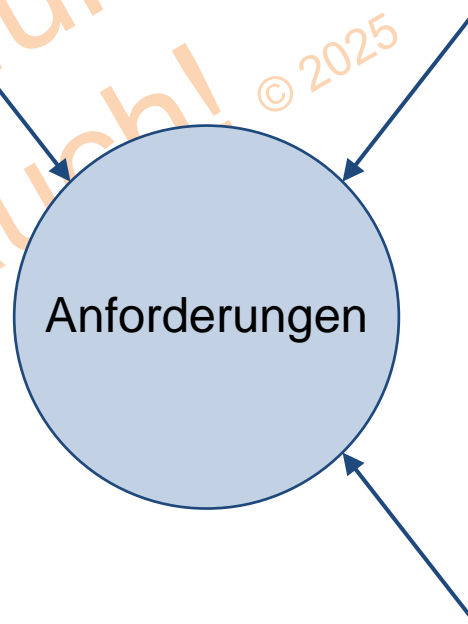


Anforderungen können aus unterschiedlichen Quellen stammen.

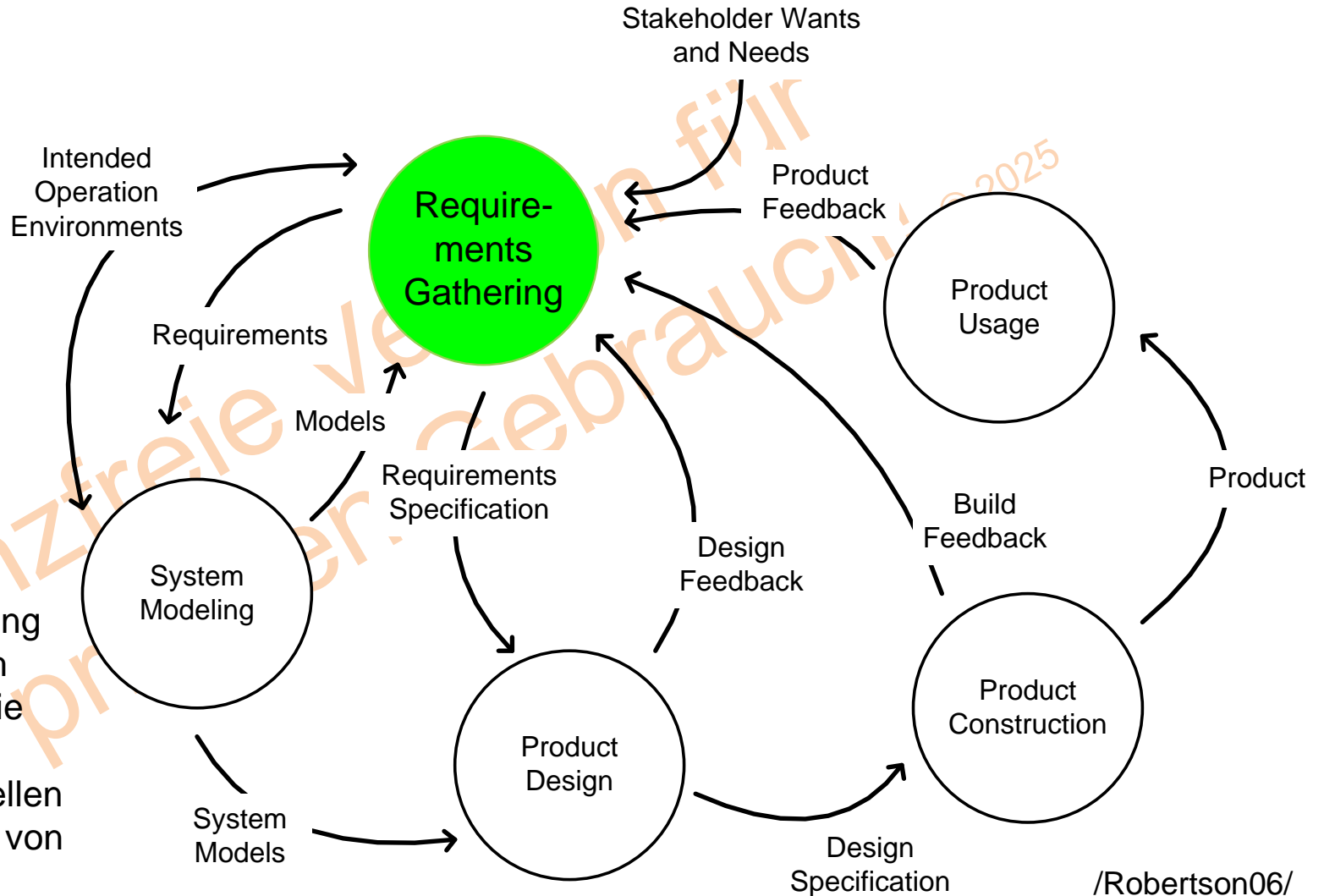
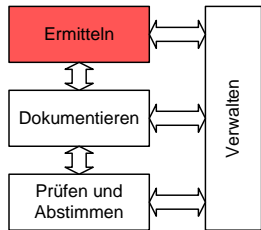
Das IREB /IREB15/ empfiehlt, die Stakeholder, bereits existierende Dokumente und sich im Betrieb befindende Systeme bei der Anforderungsermittlung zu involvieren und anzuschauen.

Stakeholder

Dokumente

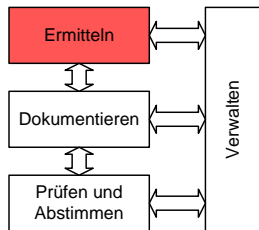


Systeme im Betrieb



Das Modell zur Ermittlung/Erhebung („Gathering“) nach Robertson zeigt die umfangreichen Einflüsse und Quellen bei der Ermittlung von Requirements.

/Robertson06/



Fragetechniken sind bei der Anforderungsermittlung besonders wichtig. Die W-Fragen helfen bei der systematischen Ermittlung des Wissens und der Bedürfnisse der Stakeholder:

WAS – Was ist das Problem, was bereitet Sorgen?

WARUM – Warum besteht dieses Problem, was ist die zugrunde liegende Zielsetzung?

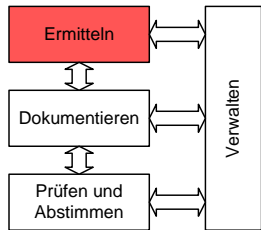
WER – Wer ist von dem Problem und seinen Auswirkungen direkt oder indirekt betroffen?

WO – Wo trat das Problem auf (Stellen, Informationsobjekte, Prozessschritte ...)?

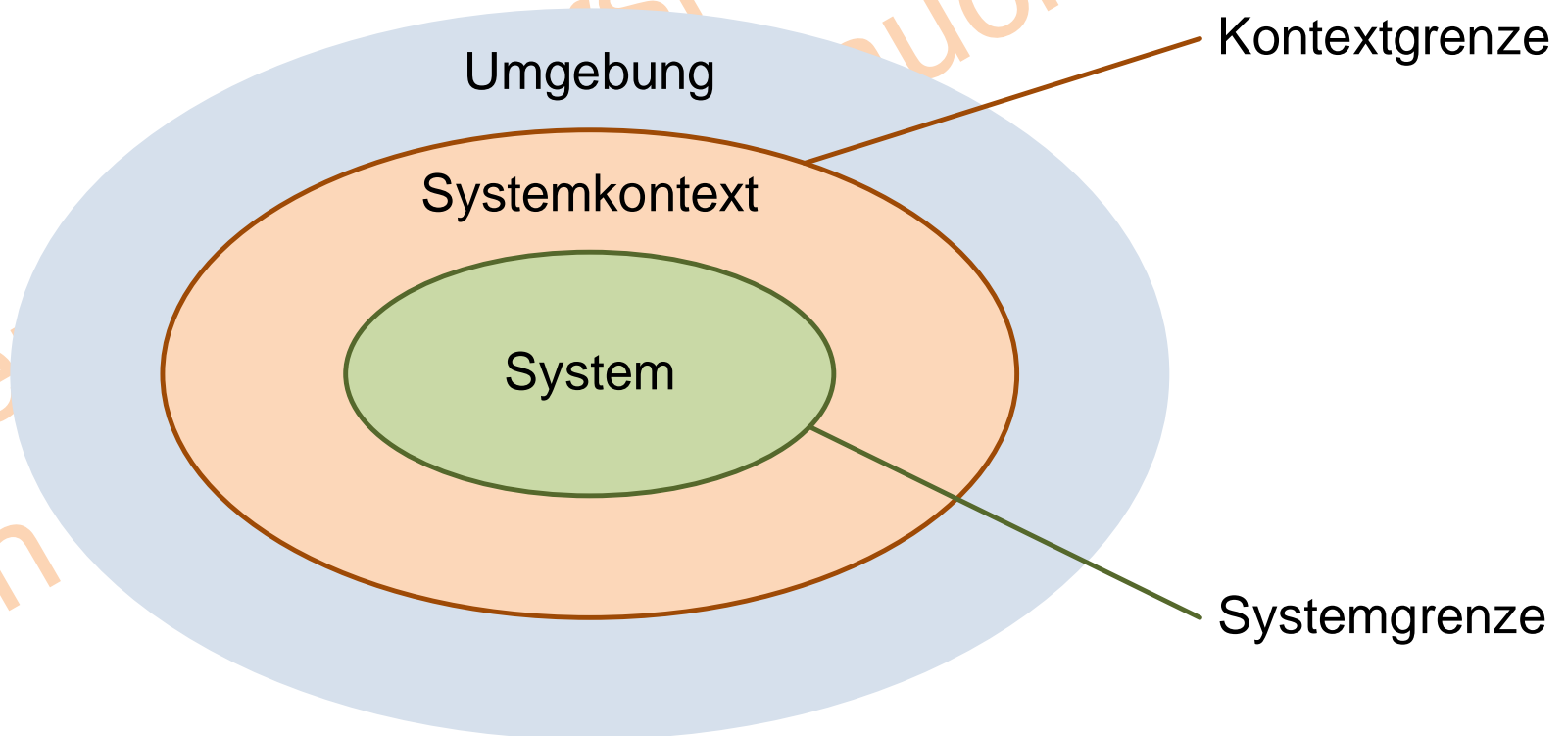
WIE – Wie kann das Problem beseitigt werden, welche Anforderungen ergeben sich?

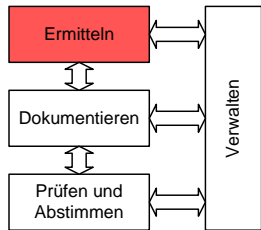
WOMIT – Womit wird das Problem (derzeit) bewältigt oder umgangen (Anwendungen, Dokumente ...)?

WIE VIEL – Wie viel Aufwand darf die Problemlösung hervorrufen (Kosten, Dringlichkeit ...)?



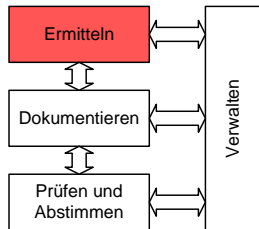
Anforderungen dienen zur Beschreibung eines Systems, welches durch einen Kontext und eine Umgebung entsprechende Grenzen besitzt. Nur für das System relevante Teile (siehe nächste Folie) müssen detailliert erfasst und beschrieben werden.





Nur die relevanten Teile einer „Realität“ müssen beim Requirements Engineering betrachtet und erfasst werden. Die Begriffe haben folgende Bedeutung:

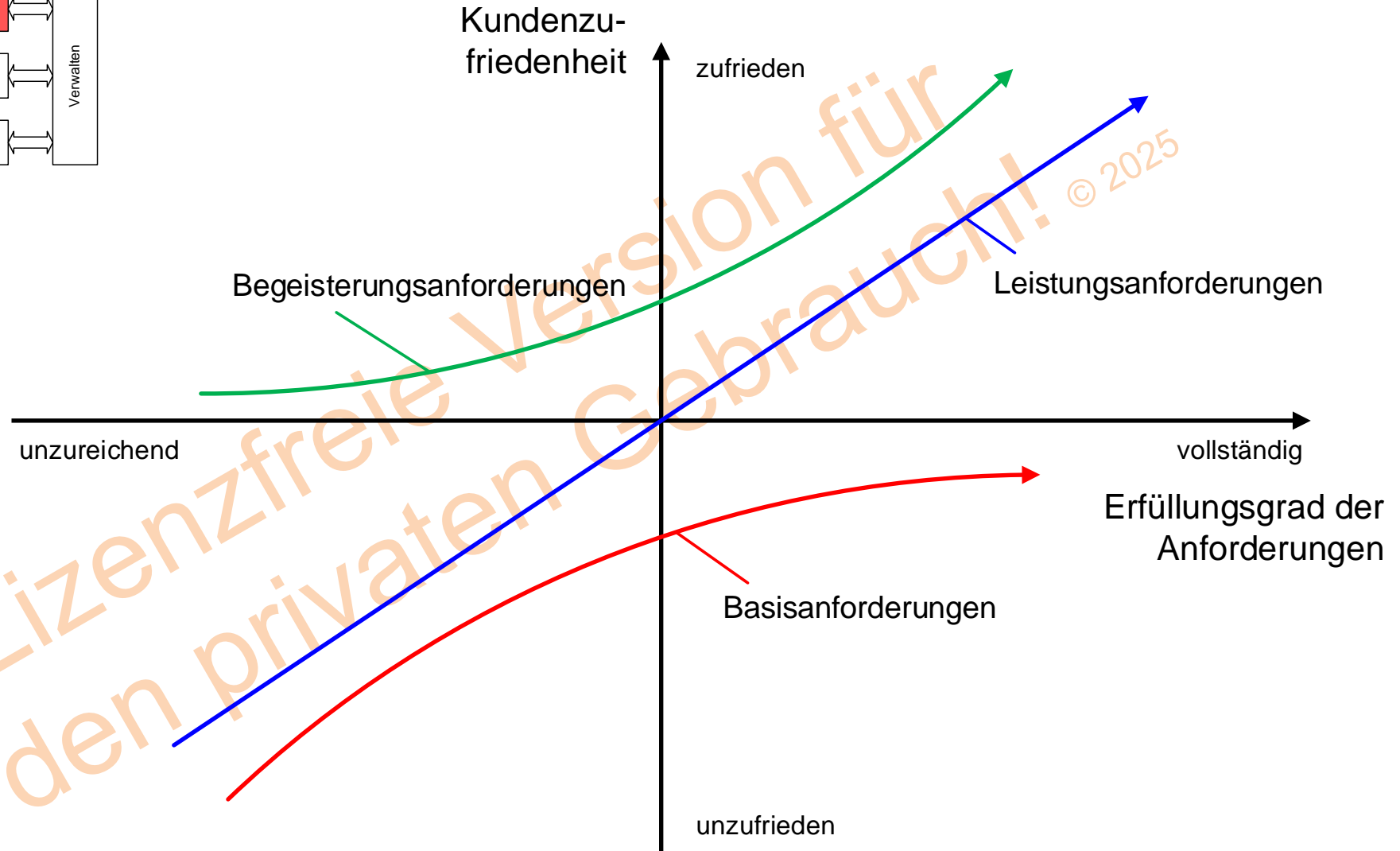
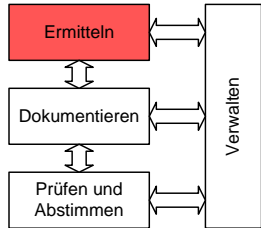
- **System:** Dies ist der gestalt- oder veränderbare Teil der Realität
- **Systemgrenze:** Die Systemgrenze trennt den gestaltbaren und veränderbaren Teil der Realität („das System“) von den nicht veränderbaren Aspekten
- **Systemkontext:** Der Teil der Umgebung des Systems, der für Definition und Verständnis der Systemanforderungen relevant ist, selbst aber nicht gestalt- oder veränderbar ist, wird als Systemkontext bezeichnet
- **Kontextgrenze:** Die Kontextgrenze grenzt den relevanten vom irrelevanten Teil der Systemumgebung ab
- **Umgebung:** Für Anforderungen irrelevanter Teil der Systemumgebung

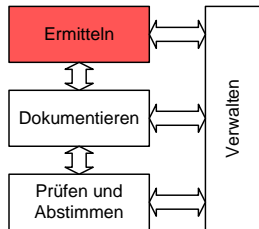


Das Kano-Modell (nach Noriaki Kano, 1978) dient der Analyse von Kundenwünschen /#Wiki-Kano-Modell/ und basiert auf Kundenbefragungen und statistischen Auswertungen.

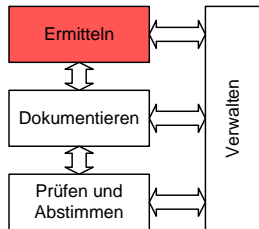
Im IREB-Kontext /IREB15/ wird eine reduzierte Variante des Kano-Modells als gedanklicher Rahmen für die Anforderungsermittlung genutzt. Dabei basieren die Kundenanforderungen auf drei Arten von Faktoren:

- **Basisfaktoren** sind selbstverständlich vorausgesetzte Systemmerkmale (unbewusstes Wissen)
- **Leistungsfaktoren** sind die explizit geforderten Systemmerkmale (bewusstes Wissen)
- **Begeisterungsfaktoren** sind Systemmerkmale, die der Stakeholder nicht kennt und erst während der Benutzung als angenehme und nützliche Überraschung entdeckt (unterbewusstes Wissen)



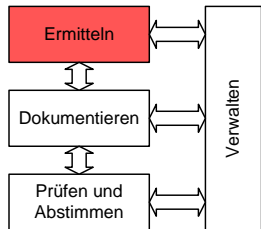


Basisanforderungen (<i>Basic Requirements</i>)	Leistungsanforderungen (<i>Performance Requirements</i>)	Begeisterungsanforderungen (<i>Excitement Requirements</i>)
<ul style="list-style-type: none"> • Kunden setzen ihre Erfüllung stillschweigend voraus • Werden von Kunden nicht explizit genannt • Erzielen keine volle Zufriedenheit, auch wenn sie erreicht werden • Führen zu starker Unzufriedenheit, wenn sie nicht erreicht werden 	<ul style="list-style-type: none"> • Kunden erwarten ihre Erfüllung • Werden von Kunden explizit genannt • Ihr Erfüllungsgrad (erfüllt, übererfüllt, nicht erfüllt) bestimmt den Grad an Zufriedenheit oder Unzufriedenheit 	<ul style="list-style-type: none"> • Kunden erwarten ihre Erfüllung nicht • Werden von Kunden nicht explizit verlangt • Ihre Erfüllung löst Begeisterung aus und hat einen überproportional starken Einfluss auf Kundenzufriedenheit, Loyalität, Wiederverkauf und Weiterempfehlung



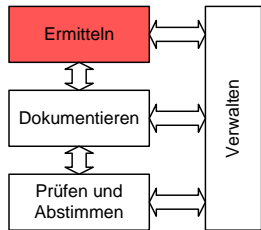
	Basisanforderungen	Leistungsanforderungen	Begeisterungsanforderungen
Auto	<ul style="list-style-type: none"> • Sicherheit • Fahrtauglichkeit 	<ul style="list-style-type: none"> • Hohe Lebensdauer • Rostschutz 	<ul style="list-style-type: none"> • Sonderausstattungen • Klimaanlage
Handy	<ul style="list-style-type: none"> • Telefoniefähigkeit • Lesbares Display • Gute Tastatur 	<ul style="list-style-type: none"> • Hohe Sprachqualität • Stabilität • Telefonbuchfunktion 	<ul style="list-style-type: none"> • Kamera • MP3-Player • Radio • Spiele

- Begeisterungsanforderungen werden im Laufe der Zeit zu Leistungsanforderungen und schließlich zu Basisanforderungen
- Die Ermittlung dieser Anforderungen erfolgt durch den Kano-Fragebogen; es werden je zwei Fragen (funktional – Eigenschaft gegeben und dysfunktional – Eigenschaft nicht gegeben) mit fünf Antwortmöglichkeiten vorgegeben („das würde mich sehr freuen“, „das setze ich voraus“, „das ist mir egal“, „das könnte ich in Kauf nehmen“, „das würde mich stören“)



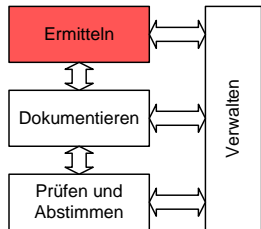
Es gibt eine Reihe von Ermittlungs- und Sammlungstechniken (Werkzeugen und Methoden zur Ermittlung von Anforderungen, *Elicitation Tools and Techniques*).

Je nach Autor und Standard werden unterschiedliche Techniken genannt – insgesamt finden sich bis zu 30 unterschiedliche Techniken. In der Praxis werden dann für den konkreten Einsatzfall die Techniken ausgewählt, die passend erscheinen; in der Regel sind dies nicht mehr als 10.



- Traditionelle Techniken: Fragebögen, Interviews, Dokumentenanalyse (Handbücher, Beschwerdebögen), Beobachtungen
- Gruppenarbeit: Brainstorming, Fokusgruppen, Konsensbildung, Workshops, Rollenspiele
- Prototyping (Rapid Prototyping): Simulationsumgebung, Ablaufmodell, Benutzerschnittstelle
- Modelle: Szenarien, Bilder, Diagramme, Mind Maps
- Kognitive und diskursorientierte Verfahren: Analyse von Interviews oder Gruppendiskussion, um herauszufinden, ob an wichtigen Stellen Informationen verschleiert wurden
- Kontextstudien: Kulturelle Besonderheiten, Farben, Symbolik

nach /Rupp14/



Art	Besonders geeignet für	Beispiele
Befragungstechniken	Bewusstes / Explizites Wissen = Leistungsfaktoren nach Kanon	Interview
		Fragebogen
Kreativitätstechniken	Unbewusstes Wissen = Begeisterungsfaktoren nach Kanon	Brainstorming
		Brainstorming paradox
		Perspektivenwechsel
		Analogietechnik
Dokumenten-zentrierte Techniken	Unterbewusstes / Implizites Wissen = Basisfaktoren nach Kanon	Systemarchäologie
Beobachtungstechniken		Perspektivenbasiertes Lesen
		Wiederverwendung von Anf.
		Feldbeobachtung
		Apprenticing

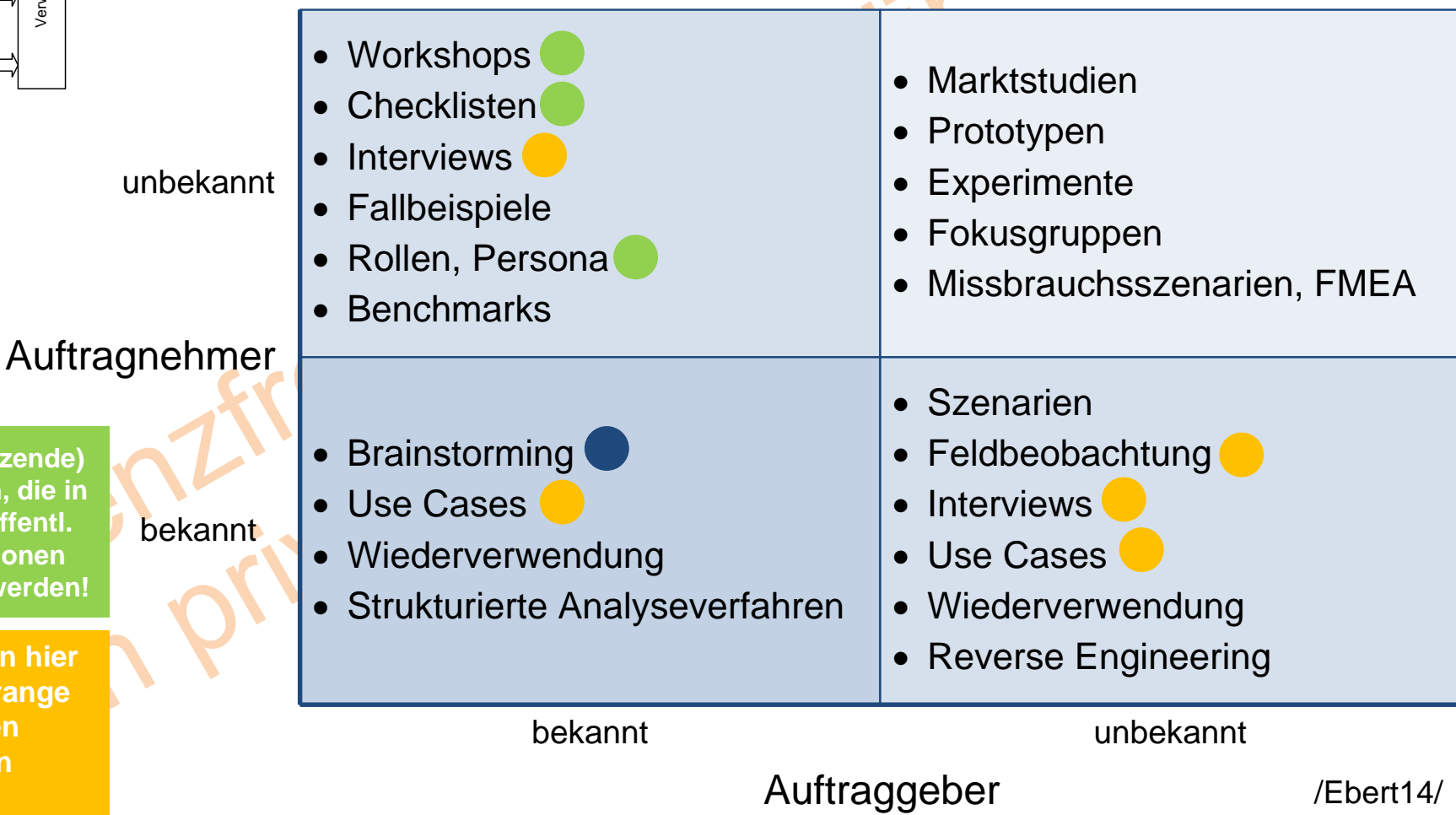
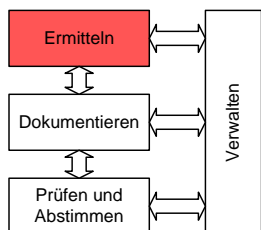


Das Brainstorming wird in der (öffentlichen) Präsentation zum Risikomanagement erläutert



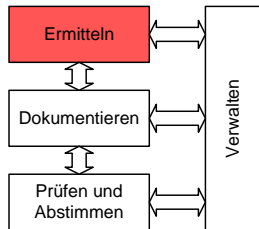
Es werden hier die mit orange markierten Techniken erklärt!

/IREB15/



(Unterstützende) Techniken, die in anderen öffentl. Präsentationen erläutert werden!

Es werden hier die mit orange markierten Techniken erklärt!



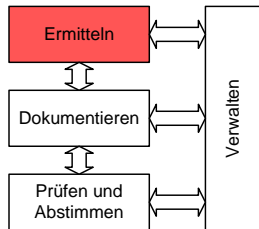
Interviews werden im RE-Kontext mit Stakeholdern geführt. Dabei sollten vorbereitete Fragen mit in das Interview genommen werden – die können ggf. auch vorab an die Interview-Teilnehmer verschickt werden. Die Ergebnisse des Interviews werden protokolliert, zusammengefasst und später weiter verwertet.

Interviews sind keine „Plauschrunden“, sondern dienen der gezielten Erfassung von Bedürfnissen. Daher sollten Interviews nur von erfahrenen Requirements Engineers durchgeführt werden.

Am Ende eines Interviews sollte der Interviewer dem Stakeholder folgende drei Fragen stellen /Ebert14/:

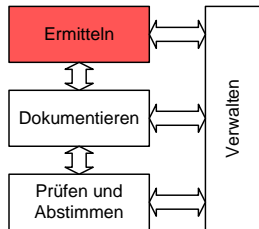
- „Sind Sie der richtige Ansprechpartner?“
- „Welche anderen Personen sollte ich noch befragen?“
- „Haben Sie den Eindruck, ich habe vergessen, Sie noch etwas zu fragen?“

Interviews können in allen Phasen des REs durchgeführt werden.



Bei der Feldbeobachtung erfasst der Requirements Engineer das Arbeitsumfeld der Stakeholder durch Betrachten der Arbeitsabläufe während der Arbeit. Hierdurch kann (für den Einzelnen) ein umfassendes Bild der Abläufe entstehen. Dieses Verfahren ist besonders hilfreich, wenn sich die Abläufe durch Sprache schwer vermitteln lassen.

Die Aufnahme von Bildern oder Videos unterstützt dieses Verfahren.



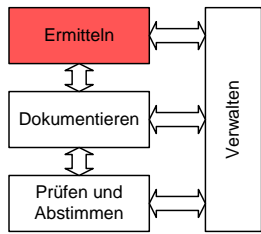
Unter Apprenticing (übersetzt in etwa „in die Lehre gehen“) versteht man das Begleiten eines Stakeholders durch den Requirements Engineer, wobei der Stakeholder einzelne Schritte seiner Arbeit dem Requirements Engineer erläutert. Mit den so erworbenen Kenntnissen und Fähigkeiten sollte der Requirements Engineer dann (einfache) Tätigkeiten aus dem Fachgebiet des Stakeholders ausüben können.



Achten Sie bei der Auswahl der Ermittlungstechniken auf die „Befindlichkeiten“ der Stakeholder.

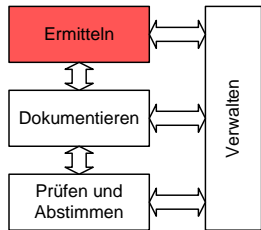
Nicht jeder Stakeholder ...

- lässt sich gerne interviewen.
- hat (unbegrenzt) Zeit für Erklärungen.
- lässt sich gerne bei der Arbeit „über die Schulter schauen“.



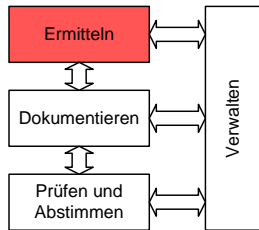
Folgende Randbedingungen sollten beachtet werden:

- Stimmen Sie vorher den Zeitrahmen und den Zielkorridor ab – auch in der langfristigen Perspektive
- Lassen Sie nur ausgebildete Requirements Engineers die Ermittlung durchführen
- Achten Sie auf die Ergebnisse: Ist das Ergebnis des Treffens mit dem Stakeholder so ausgefallen wie erwartet?



Ein Glossar ist eine Sammlung von Begriffsdefinitionen. Beim Requirements Engineering (für ein Produkt oder ein Projekt) sollte möglichst frühzeitig mit der Erstellung eines Glossars begonnen werden, um so allen Beteiligten von Beginn an ein gemeinsames Verständnis der zentralen Begriffe zu ermöglichen.

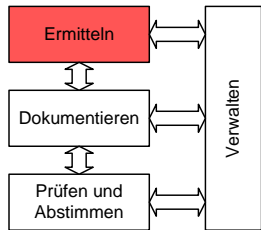
Ohne Glossar kommt es häufig zu Missverständnissen, gerade bei Begriffen, die jedermann kennt, die aber dennoch unterschiedlich interpretiert werden (können).



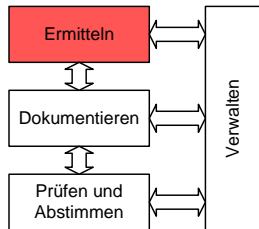
In ein Glossar gehören:

- (Kontextspezifische) Fachbegriffe
- Abkürzungen und Akronyme
- Alltägliche Begriffe, die im gegebenen Kontext eine spezifische Bedeutung haben
- Synonyme (mehrere Worte haben eine Bedeutung); hier sollte auch festgelegt werden, dass nur ein Begriff/Wort verwendet werden soll
- Homonyme (ein Wort hat mehrere Bedeutungen)

/#Capri-13/



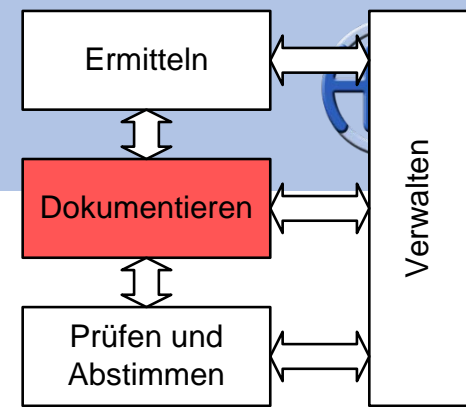
- Begriff (deutsch, englisch)
- Abkürzung
- Definition, Erläuterung
- Quelle
- Norm, Standard (falls vorhanden)
- Erfasst durch
- Erfasst am
- Zuletzt geändert durch
- Zuletzt geändert am
- Priorität (A, B, C)
- Status (Abgeschlossen, OK, in Diskussion)



- Das Glossar muss zentral verwaltet werden
- Es müssen Verantwortlichkeiten zur Glossarpflege definiert werden
- Das Glossar muss projektbegleitend gepflegt werden
- Das Glossar muss allgemein zugänglich sein
- Das Glossar muss verbindlich verwendet werden
- Die Herkunft der Begriffe sollte im Glossar enthalten sein
- Das Glossar muss mit den Stakeholdern abgestimmt sein
- Die Einträge des Glossars müssen eine einheitliche Struktur aufweisen

/#Capri-13/

Kapitel 5: Anforderungen dokumentieren

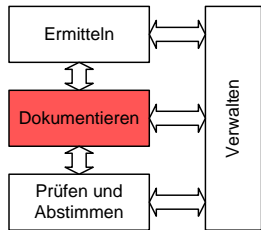


Teil II

- Die Anforderungsspezifikation
- Die drei Perspektiven von Anforderungen (Zusammenhänge, Beschreibung)
- Notationen
- Dokumentenstrukturen
- Das Volere-Template (Grundsätzliches, Aufbau englisch/deutsch)
- Die Anforderungsspezifikation nach IEEE 830-1998 (Grundsätzliches, Struktur)
- Natürlichsprachlich dokumentieren: Übersicht
- Sprachliche Effekte des REs
- Anforderungsumgestaltung (Effekte, Formen von Tilgung, Arten der Generalisierung, Varianten der Verzerrung, Abschließende Bemerkungen)
- Die Satzschablone (Generelles, Schritte 1 und 2, Prozesswortliste, Schritt 3, Schritt 4, Schritt 5, Einsetzbarkeit)
- Übung 3: Satzschablone
- Modellbasiert dokumentieren (Definitionen, Sichten, Stärken und Schwächen)
- Anmerkungen zur Modellierung
- Modellierungsansätze
- Das Entity-Relationship-Diagramm (Grundsätzliches, Beispiel)
- Die UML (Grundsätzliches, 14 Diagramme)
- Das Use-Case-Diagramm (Grundsätzliches, Elemente, Beispiel „Download-Bereich der Website“, Anmerkungen)
- Übung 4: Use-Case-Diagramm
- Anmerkungen und Tipps zum Dokumentieren von Anforderungen

Kapitel 5

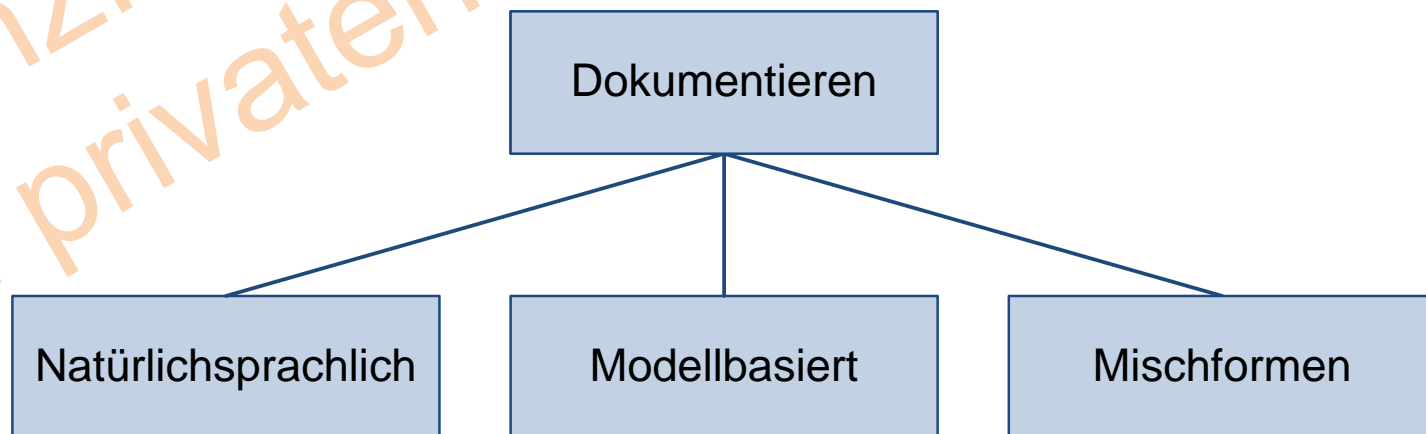
Seite
74–115

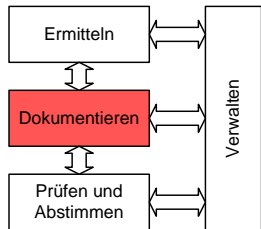


Beim IREB /IREB15/ wird das Ergebnis der Dokumentation als Anforderungsdokument oder Anforderungsspezifikation bezeichnet und folgendermaßen definiert:

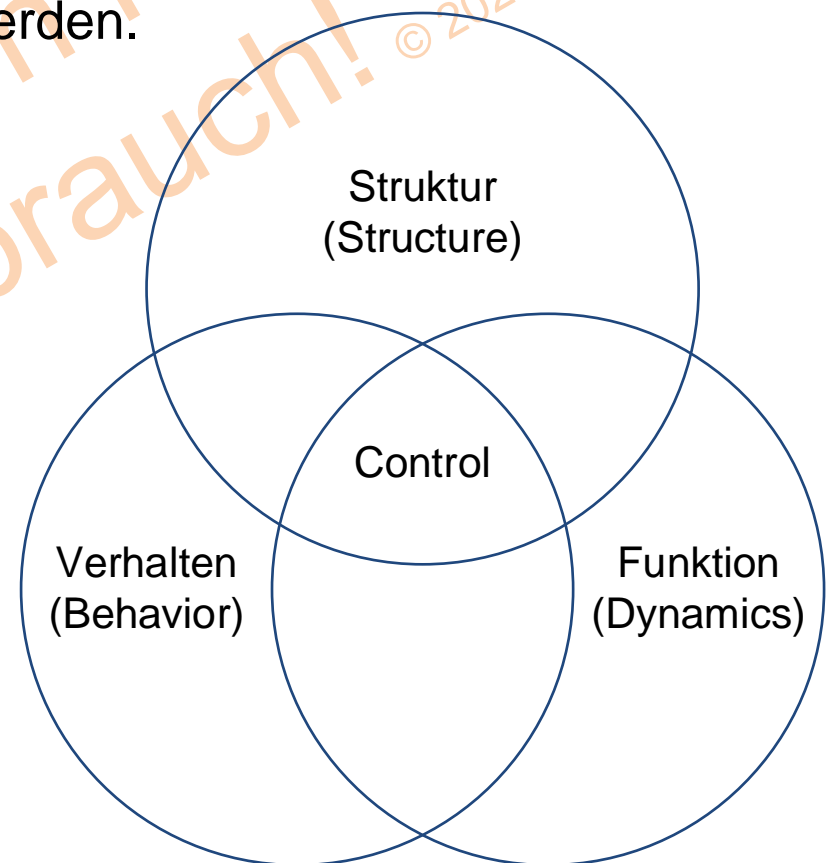
„Eine **Anforderungsspezifikation** ist eine systematisch dargestellte Sammlung von Anforderungen (typischerweise für ein System oder eine Komponente), die vorgegebenen Kriterien genügt.“

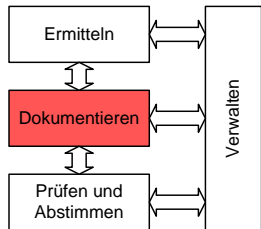
Anforderungsspezifikationen können dabei in natürlicher Sprache oder durch konzeptuelle Modelle (oder als Mischform) verfasst werden.





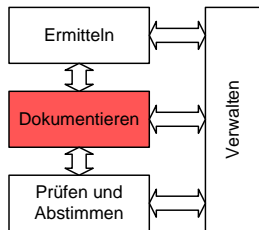
Anforderungen können aus drei Sichten (Perspektiven) beschrieben werden: Die Struktur-, die Funktions- und die Verhaltenssicht. Die drei Perspektiven hängen zusammen und sollten als Ganzes betrachtet werden.





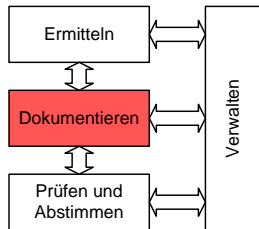
Perspektive	Beschreibung	Modelle (Beispiele)
Struktur	In der Strukturperspektive wird eine statisch-strukturelle Perspektive auf die Anforderungen an das System eingenommen. In dieser Perspektive werden z.B. die Struktur von Ein- und Ausgabedaten sowie die statisch-strukturellen Aspekte von Nutzungs- und Abhängigkeitsbeziehungen des Systems im Systemkontext dokumentiert (z.B. die zu nutzenden Dienste eines externen Systems)	UML- Klassendiagramme, ER-Diagramme
Funktion	In der Funktionsperspektive wird dokumentiert, welche Informationen (Daten) aus dem Systemkontext durch das zu entwickelnde System bzw. dessen Funktionen manipuliert werden und welche Daten vom System in den Systemkontext fließen. Gegebenenfalls wird auch die Systematik der Funktionsausführung zur Verarbeitung der Eingabedaten dokumentiert	UML- Aktivitätsdiagramme, Datenfluss- diagramme
Verhalten	In der Verhaltensperspektive wird das System und dessen Einbettung im Systemkontext zustandsorientiert dokumentiert, indem z.B. die Reaktion des Systems auf Ereignisse im Systemkontext, Bedingungen eines Zustandswechsels sowie Effekte dokumentiert werden, die das System in der Umgebung erbringen soll (z.B. Effekt des betrachteten Systems, der für ein anderes System im Systemkontext ein Ereignis darstellt)	UML- Zustandsdiagramme, Statecharts

/IREB15/



- Natürliche Sprache
- Satzschablonen (zur Verbesserung der natürlichen Sprache)
- Anwendungsfalldiagramme (Use Cases)
- Entscheidungstabellen
- Klassendiagramme
- Verhaltensdiagramme (Zustandsdiagramme, Aktivitätsdiagramme)
- Programmcode
- Weitere Notationsformen, falls diese im konkreten Kontext sinnvoll erscheinen
- „DSL“ – Domain Specific Language: Spezielle Sprach- und Notationsformen des Problembereichs
- ... und „alles, was sinnvoll ist“

vgl. /Rupp14/

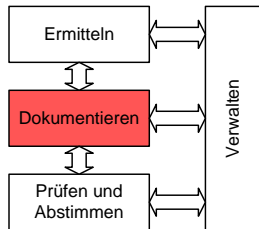


Das Aussehen, d.h. die Gliederung und die Inhalte des zentralen Dokuments für die / alle Anforderungen eines Projekts ist von erheblicher Bedeutung im Requirements Engineering.

In der Praxis existieren (im deutschsprachigen Raum) drei Gliederungsvorlagen; dies sind ...

- das Volere-Template von Suzanne und James Robertson,
- die Struktur für Software Requirements Specification (SRS) nach IEEE 830-1998 und
- die Dokumentenstruktur nach dem V-Modell XT.

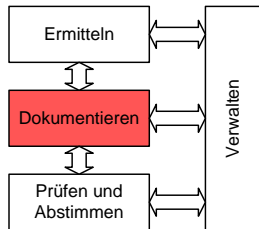
Die Dokumentenstrukturen werden auf den nachfolgenden Folien vorgestellt. In der konkreten Umsetzung werden dann diese Strukturen miteinander vermischt.



Das Volere-Template ist eine Vorlage zur Erstellung von Anforderungsdokumenten (in englischer Sprache). Gleichzeitig ist es ein Vorgehensmodell zur Erstellung von Anforderungen nach Robertson & Robertson /Robertson06, Robertson12/.

Für die konkrete Umsetzung kann das Volere-Template /Volere/ aus dem Internet heruntergeladen (siehe Weblink-Liste im Anhang) und angepasst werden. Bis 2008 war dies noch kostenfrei möglich, inzwischen verlangen die Autoren etwa 50 € für die nicht-universitäre Nutzung.

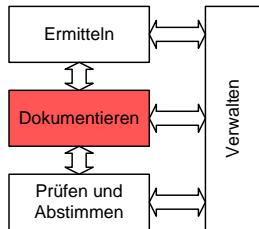
Das Volere-Template ist für den gesamten RE-Lebenszyklus gedacht und in 4 Kategorien mit 27 Teilbereichen unterteilt (siehe nächste Folie).



Project Drivers
1. The Purpose of the Project
2. The Stakeholders
3. Mandated Constraints
4. Naming Conventions and Terminology
5. Relevant Facts and Assumptions
6. The Scope of the Work
7. Business Data Model and Data Dictionary
8. The Scope of the Product
Functional Requirements
9. Functional Requirements

Nonfunctional Requirements
10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural Requirements
17. Compliance Requirements
Project Issues
18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

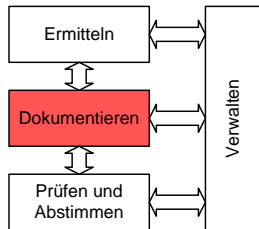
/Volere, V17, 2014/



Projekttreiber
1. Der Zweck des Projekts
2. Die Stakeholders
3. Geforderte Randbedingungen
4. Namenskonventionen und Terminologie
5. Relevante Fakten und Annahmen
6. Arbeitsumfang
7. Business Datenmodell und Data Dictionary
8. Abgrenzung des Produkts
Funktionale Anforderungen
9. Funktionale Anforderungen

Nichtfunktionale Anforderungen
10. Look-and-Feel-Anforderungen
11. Benutzbarkeitsanforderungen
12. Performanceanforderungen
13. Operative und Umwelтанforderungen
14. Wartungs- und Supportanforderungen
15. Sicherheitsanforderungen
16. Kulturelle Anforderungen
17. Rechtliche Anforderungen
Projektangelegenheiten
18. Offene Punkte
19. Fertiglösungen
20. Neue Probleme
21. Aufgaben
22. Migration zum neuen Produkt
23. Risiken
24. Kosten
25. Benutzerdokumentation und Schulung
26. Warteraum
27. Lösungsideen

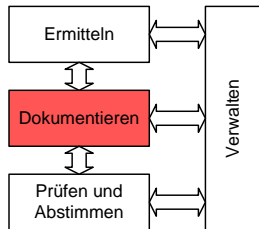
/Volere, V17, 2014/



Das IEEE legt in dem Standard IEEE 830-1998 (unter anderem) fest, wie eine Software Requirements Specification (Anforderungsspezifikation) aufgebaut sein soll.

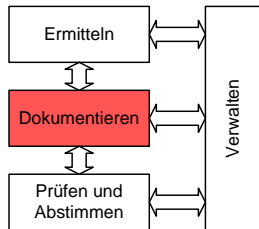
Generell gibt es drei Hauptkapitel sowie Anhänge und Indexe (die hier nicht dargestellt sind).

In der Praxis wird dieser Standard oftmals modifiziert eingesetzt.



Die Kapitel 1 und 2 (Einführung und Überblick) sind von der Struktur her festgelegt, Kapitel 3 jedoch nicht.

1 Introduction	3 Specific Requirements
1.1 Purpose	3.1 External Interface Requirements
1.2 Scope	3.1.1 User Interfaces
1.3 Definitions, Acronyms, and Abbreviations	3.1.2 Hardware Interfaces
1.4 References	3.1.3 Software Interfaces
1.5 Overview	3.1.4 Communications Protocols
2 Overall Description	3.2 Functional Requirements
2.1 Product Perspective	3.2.1 ...
2.2 Product Functions	3.2.x <i>depends on specific approach</i>
2.3 User Characteristics	3.2.x ...
2.4 Constraints	3.3 Performance Requirements
2.5 Assumptions and Dependencies	3.4 Design Constraints
	3.5 Software System Attributes
	3.6 Other Requirements



Natürlich-
sprachlich
dokumentieren

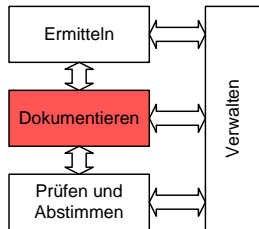
Die natürlichsprachliche Dokumentation ist im RE stark verankert, insbesondere da sie fast voraussetzungslos in allen Gebieten eingesetzt werden kann.

Stärken:

- Universell einsetzbar
- Erfordert keine Einarbeitungszeit
- Kann von jedem gelesen (und verstanden) werden

Schwächen:

- Nicht (immer) eindeutig
 - Bei komplexen Sachverhalten nicht ausreichend
- Hier wird auf zwei Aspekte der natürlichsprachlichen Dokumentation eingegangen:
- Sprachliche Effekte
 - Die Satzschablone



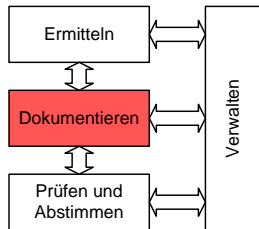
Natürlich-
sprachlich
dokumentieren

Das Ziel eines RE-Prozesses sollte es sein, die Anforderungen so zusammenzustellen, dass sie unabhängig vom Ersteller (und vom Umsetzer) verständlich sind, d.h. den vorher benannten Kriterien genügen.

Dies bedeutet aber, dass eine Vereinheitlichung der (verwendeten) Sprache stattfinden muss, indem die sprachlichen Unzulänglichkeiten (Defekte) eliminiert werden. Automatisch, d.h. durch ein System, ist dies nicht machbar, aber es gibt Möglichkeiten, mit bestimmten Vorgehensweisen sprachliche Defekte zu entdecken.

Einen (systematischen) Ansatz hierzu bietet das NLP (Neurolinguistische Programmieren). Die Umsetzung des NLP im RE wird in /Rupp14/ ausführlich beschrieben.

Hierbei werden **Anforderungen so „umgestaltet“**, dass sie qualitativ hochwertig sind, d.h. **Defekte (auch Effekte) eliminiert** worden sind.



Natürlich-
sprachlich
dokumentieren

Tilgung (engl. *Deletion*):

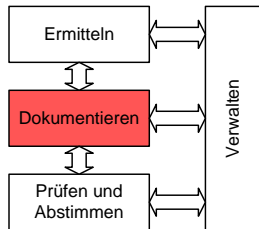
Tilgung ist ein Prozess, durch den wir unsere Aufmerksamkeit selektiv bestimmten Dimensionen unserer Erfahrung zuwenden und andere ausschließen. Tilgung reduziert die Welt auf Ausmaße, mit denen wir umgehen können.

Generalisierung (engl. *Generalization*):

Generalisierung ist der Prozess, durch den Elemente oder Teile eines persönlichen Modells von der ursprünglichen Erfahrung abgelöst werden, um dann die gesamte Kategorie, von der diese ein Beispiel darstellt, zu verkörpern.

Verzerrung (engl. *Distortion*):

Verzerrung ist der Prozess, der es uns ermöglicht, in unserer Erfahrung sensorischer Einzelheiten eine Umgestaltung vorzunehmen.



Natürlich-
sprachlich
dokumentieren

Unvollständig spezifizierte Prozesswörter: Manche Wörter benötigen mehr als ein Substantiv, z.B. das Verb „übertragen“. „Ich übertrage die Verantwortung an Dich“.

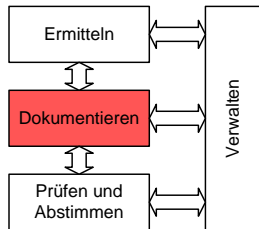
Modaloperatoren der Möglichkeit: Diese sollen beschreiben, wie etwas erreicht werden könnte, z.B. „die Tür kann dauerhaft gesichert werden können. Dies könnte durch ein Schloss erreicht werden.“

Unvollständige Vergleiche: Bei Vergleichen (Komparativen) und Steigerungen (Superlativen) muss der Bezugspunkt angegeben werden, so z.B. „das System ist schneller als“ (das Altsystem bei der Verarbeitung von Dateien).

Modaloperatoren der Notwendigkeit: Hier werden Sachverhalte dadurch ergänzt, dass der Ausnahmefall beschrieben wird. „Das System muss immer den vollständigen Sicherheitscode einlesen; falls dies nicht möglich ist, wird ein Code generiert.“

Implizite Annahmen: Hierüber werden Aussagen erfasst, die das Vorwissen des Autors benötigen würden.

vgl. /Rupp14/



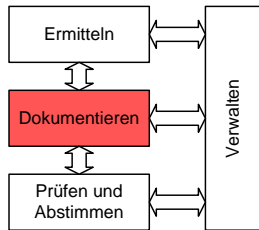
Natürlich-
sprachlich
dokumentieren

Universalquantoren: Hiermit sind Aussagen über Häufigkeiten gemeint. Worte wie „immer“, „alle“, „irgendeiner“ deuten darauf hin. Die Universalquantoren sollten immer hinterfragt werden („wirklich alle?“)

Unvollständig spezifizierte Bedingungen: Werden Aussagen getroffen über ein Verhalten, wenn etwas geschieht, so müssen auch Aussagen getroffen werden, wenn es nicht eintritt. Typische einleitende Worte sind hier beispielsweise „wenn“, „dann“, „falls“ oder „abhängig von“. In diesem Fall sollte die Alternative ausformuliert werden.

Substantive ohne Bezugsindex: Einige Substantive enthalten einen impliziten Bezug, der zu unterschiedlichen Interpretationen führen kann. Beispiele sind „die Daten“, „der Anwender“, „das Terminal“, „die Meldung“ etc. In diesen Fällen sollten die Substantive zumindest hinterfragt (z.B. „welcher Anwender genau“, „welche Fehlermeldung“) und ggf. ergänzt werden („die Fehlermeldung Nr. 212, die auf den Eingabefehler hinweist.“).

vgl. /Rupp14/

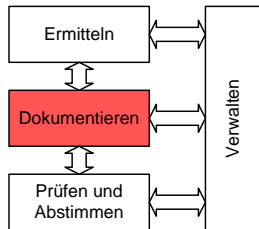


Natürlich-
sprachlich
dokumentieren

Nominalisierungen: Dies ist eine (komplexe) Transformation von Verben zu Substantiven, die benutzt wird, um komplexe Prozesse zu einem einmaligen Ereignis zu machen. Dabei kann – das ist ja im Ansatz auch so gewollt – einiges an Informationen verloren gehen. Beispiele hierfür sind: „Die Reservierung“, „die Buchung“, „der Neustart“. Lassen die verwendeten Nominalisierungen Interpretationsspielraum zu, so müssen sie entweder erklärt (im Glossar) oder aufgelöst werden.

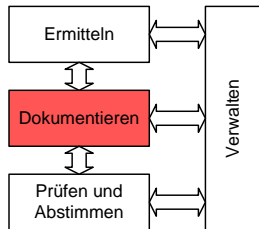
Funktionsverbgefüge: Die Kombination aus inhaltsarmen Verben (wie machen, können, haben, sein) und sinngebenden Substantiven führt zu ungenauen Aussagen. Besser ist es, hier die Sätze mit aktiven Verben zu formulieren. Beispiele: „Am Ende sein“ -> „beendet sein“ (passiv) -> „enden“ (aktiv); „im Bau sein“ -> „gebaut werden“ (passiv) -> „bauen“ (aktiv).

vgl. /Rupp14/



Natürlich-
sprachlich
dokumentieren

- Obwohl die meisten Anforderungen mit der Sprache (und nicht mit Diagrammen) erfasst und modelliert werden, wird der Bereich der „richtigen“ Verwendung der Sprache in der Literatur kaum angesprochen
- Der Einsatz der Umgestaltungsregeln bedarf einiger Übung und sollte separat geschult werden
- Zur Erstellung „guter“ Anforderungen (Sätze) kann das SOPHIST REGelwerk verwendet werden, siehe /Rupp14/



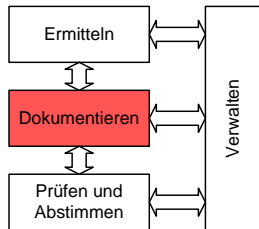
Natürlich-
sprachlich
dokumentieren

„Eine Satzschablone (engl. *Requirements Template*) ist ein Bauplan für die syntaktische Struktur einer einzelnen Anforderung.“ /IREB15/

Eine Satzschablone wird eingesetzt, um von vornherein ein gleiches Verständnis von den einzelnen Anforderungen zu erhalten.

Die richtige Anwendung einer Satzschablone wird in fünf Schritten durchgeführt /Rupp14/:

1. Festlegen der rechtlichen Verbindlichkeit
2. Den Kern der Anforderung bestimmen
3. Charakterisieren der Aktivität des Systems
4. Objekte einfügen
5. Formulierung von logischen und zeitlichen Bedingungen



Natürlich-
sprachlich
dokumentieren

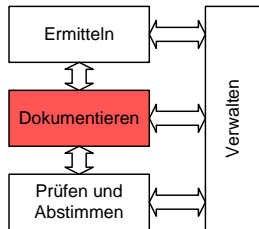
1. Festlegen der rechtlichen Verbindlichkeit

Die rechtliche Verbindlichkeit kann über die Verwendung der Verben MUSS, SOLLTE, WIRD und KANN hergestellt werden, wobei auf das Verb KANN verzichtet werden kann.

Verbindlichkeit	Schlüsselwort
Pflicht	muss
Wunsch	sollte
Absicht	wird

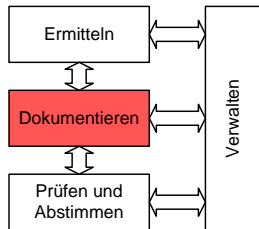
2. Den Kern der Anforderung bestimmen

Die Funktionalität der Anforderung kann durch Prozesse beschrieben werden; hierzu gibt es eine Reihe von Prozessworten wie Drucken, Speichern, Übertragen, etc. Über das Glossar (des Projekts) werden diese Prozessworte (vorab) bestimmt und definiert. Eine Prozesswortliste ist auf der nächsten Folie zu finden.



Natürlich-
sprachlich
dokumentieren

Prozesswort	Semantische Definition des Prozessworts	Selbst- ständ. System- aktivität	Benut- zer- inter- aktion	Poten- zielle Fähig- keit
auswählen	Der NUTZER selektiert eines oder mehrere Elemente aus einer endlichen Menge von Elementen (siehe auch „bestimmen“).	Nein	Ja	Nein
bestimmen	Das SYSTEM selektiert anhand bestimmter Selektionskriterien aus einer endlichen Menge (zum Beispiel aus einer Datenbank) ein oder mehrere Elemente.	Ja	Nein	Nein
darstellen	Das SYSTEM zeigt dem Nutzer Informationen an.	Ja	Nein	Nein
einfügen	Ausschließlich das SYSTEM gibt neue Daten ein oder überschreibt vorhandene Daten (siehe auch „eingeben“, „einsetzen“).	Ja	Nein	Nein
eingeben	Ausschließlich der NUTZER gibt neue Daten ein oder überschreibt vorhandene Daten (siehe auch „einfügen“, „einsetzen“).	Nein	Ja	Nein
einsetzen	Sowohl das SYSTEM als auch der NUTZER gibt neue Daten ein oder überschreibt vorhandene Daten (siehe auch „einfügen“, „eingeben“).	Ja	Ja	Nein
empfangen	Das SYSTEM ist IMSTANDE, Daten von einem externen System elektronisch entgegenzunehmen.	Nein	Nein	Ja
erstellen	Sowohl das SYSTEM (selbständig) als auch der NUTZER erzeugen Objekte.	Ja	Ja	Nein



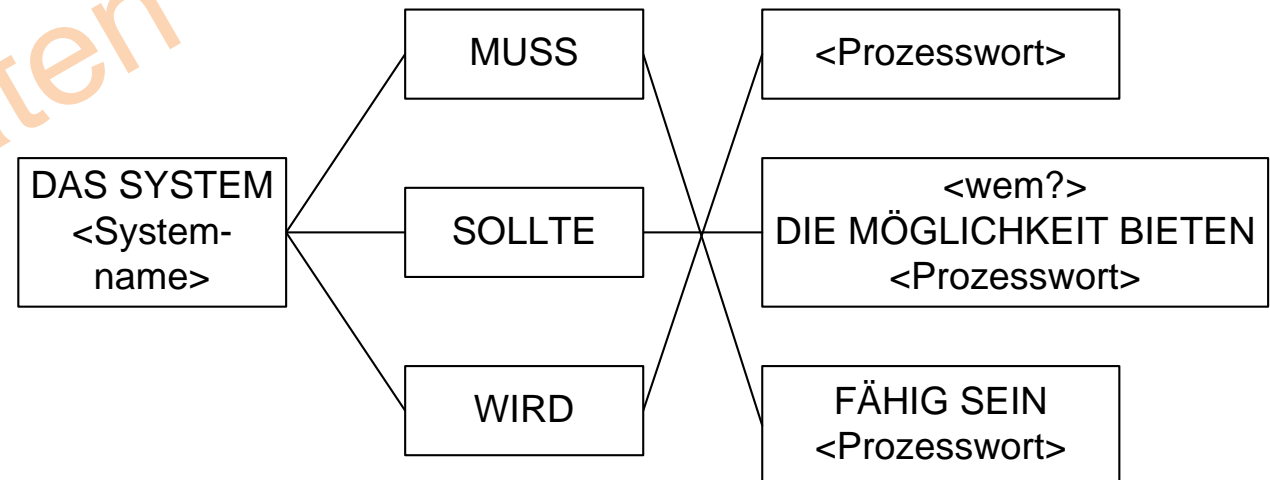
Natürlich-
sprachlich
dokumentieren

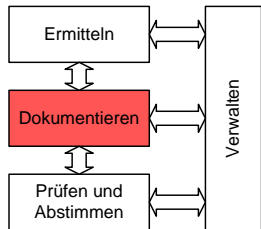
3. Charakterisieren der Aktivität des Systems

Die Aktivitäten des Systems werden in drei Arten unterteilt:

1. Selbstständige Systemaktivität: Das System führt den Prozess selbstständig durch
2. Benutzerinteraktion: Das System stellt dem Benutzer die Prozessfunktionalität zur Verfügung
3. Schnittstellenanforderung: Das System führt einen Prozess in Abhängigkeit von einem Dritten aus, ist an sich passiv und wartet auf ein externes Ereignis

Damit ergibt sich folgende Satzschablone zur Beschreibung von einfachen Anforderungen:



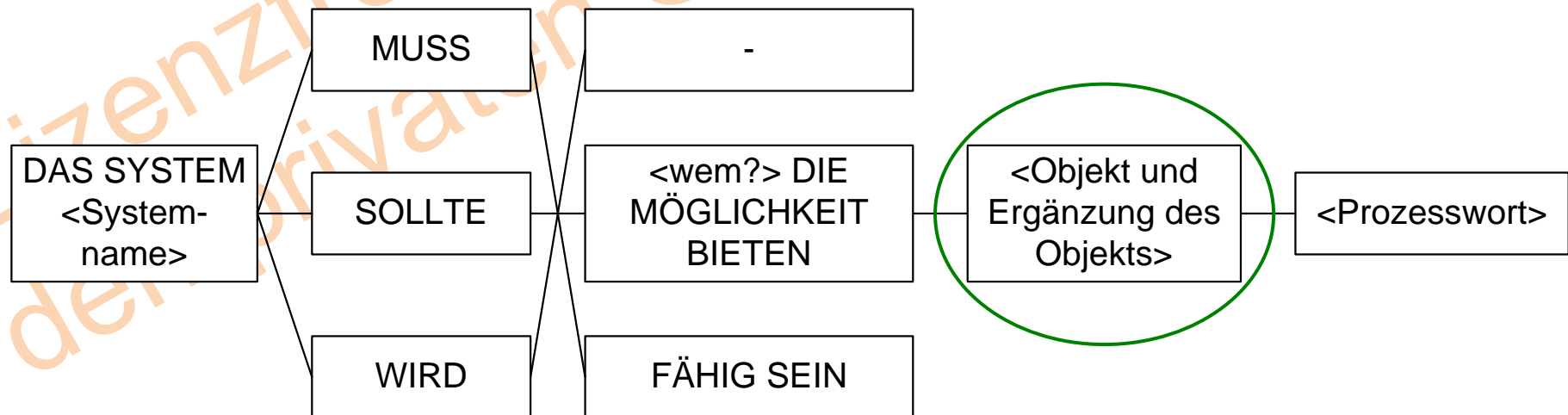


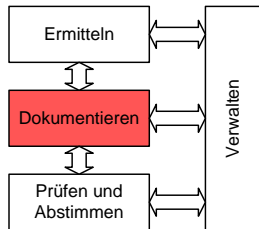
Natürlich-
sprachlich
dokumentieren

4. Objekte einfügen

Manche Prozesswörter benötigen zur Ergänzung ein oder mehrere Objekte. Im Schritt 4 werden die noch fehlenden Objekte und Ergänzungen der Objekte in der Anforderung identifiziert und ergänzt.

Hierdurch erweitert sich die Satzschablone:



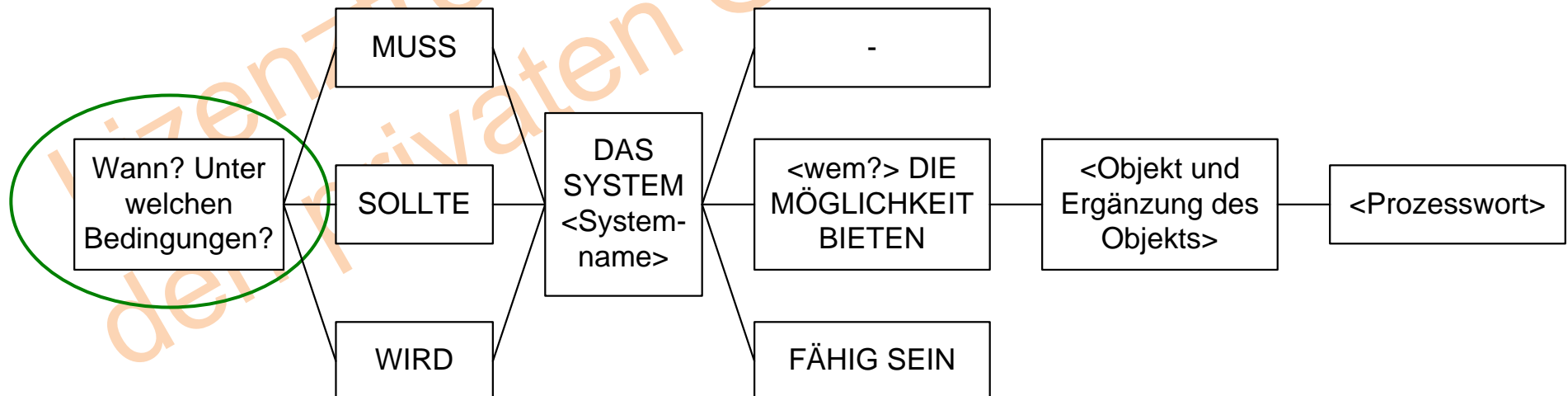


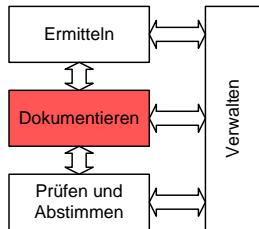
5. Formulierung von logischen und zeitlichen Bedingungen

Sind die Anforderungen nur in einem bestimmten zeitlichen oder logischen Rahmen gültig, so werden diese vorangestellt.

Natürlich-
sprachlich
dokumentieren

Nach Schritt 5 hat die Satzschablone folgenden Aufbau:

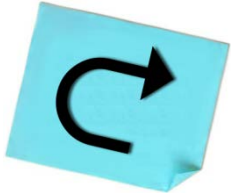




Die Verwendung von Satzschablonen ist vergleichsweise einfach, schränkt jedoch die Kreativität des Einzelnen ein. Dafür ist das Ergebnis normiert und reduziert Mehrfachinterpretationen.

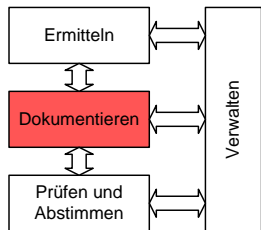
Natürlich-
sprachlich
dokumentieren

Lizenzfreie Version für
den privaten Gebrauch! © 2025



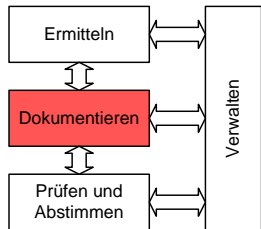
Erstellen Sie (die vier bis sechs wichtigsten) Anforderungen für den Betrieb einer Kaffeemaschine mit Hilfe der Satzschablone.

Dauer:
15 Min.



Natürlich-
sprachlich
dokumentieren

Keine
Muster-
lösung!

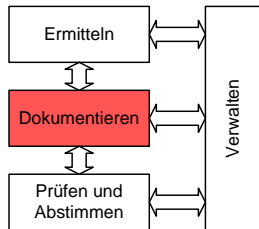


Modellbasiert
dokumentieren

Die modellbasierte Dokumentation (oder kurz: die Modellierung) macht einen Großteil des Requirements Engineering aus. Vielfach ist es erst hierüber möglich, eine genügend gute Beschreibung des umzusetzenden Systems zu erhalten.

Definitionen:

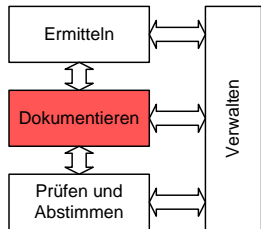
- „Ein Modell ist ein abstrahierendes Abbild einer existierenden Realität oder Vorbild für eine zu schaffende Realität.“
- „Ein Modell ist eine Abstraktion eines beliebigen Originals. Es beschreibt die für den Einsatzzweck des Modells relevanten Aspekte des Originals.“ /Beneken-Vorles-13/



Modellbasiert
dokumentieren

Ein Modell beschreibt nur bestimmte Aspekte. Diese werden aus einer bestimmten Sicht (= Perspektive) beschrieben. Typische Sichten sind:

- Kontextsicht
- Funktionssicht
- **Struktursicht**
- **Verhaltenssicht**
- Schnittstellensicht



Modellbasiert
dokumentieren

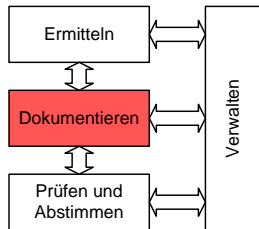
Die modellbasierte Dokumentation weist einige Stärken, aber auch Schwächen auf.

Stärken:

- Hohe Eindeutigkeit
- Auch für komplexe Sachverhalte geeignet

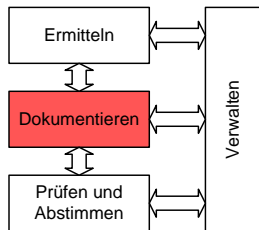
Schwächen:

- Häufig sehr aufwendig zu erstellen
- Erfordert eine Einarbeitungszeit
- Kann nur von Fachleuten gelesen (und verstanden) werden



Modellbasiert
dokumentieren

- Modellierung ist eine eigenständige Disziplin, die sehr umfangreich ist. Daher werden hier nur Einblicke und Ausschnitte gezeigt
- Im Internet ist eine Vielzahl von Anleitungen und Beispielen zur Modellierung zu finden
- Die Modellierung von Anforderungen ist auch „Mode-Schwankungen“ unterworfen. Daher haben Modellierungsmethoden aus den 70er bis 90er Jahren des letzten Jahrhunderts größtenteils kaum noch praktische Relevanz



Modellbasiert
dokumentieren

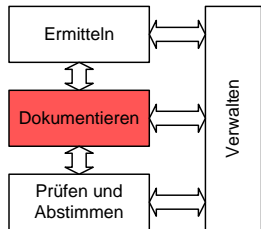
Es gibt eine große Anzahl von Modellierungsansätzen. Diese lassen sich generell den Strukturierten Methoden oder den Objektorientierten Methoden zuordnen.

In der Strukturierten Analyse (SA aus den 70er Jahren des 20. Jahrhunderts, nach Tom DeMarco) werden folgende Elemente verwendet:

- Kontextdiagramm
- Datenflussdiagramm
- Glossar/Datenwörterbuch

Zudem wird zu den SA-Methoden noch hinzugefügt:

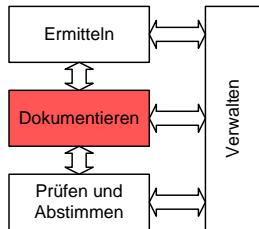
- Entity-Relationship-Diagramm



Modellbasiert
dokumentieren

Aus der Objektorientierten Analyse und dem Objektorientierten Design (beide mit Ursprüngen in den 90er Jahren des 20. Jahrhunderts) hat sich die UML (Unified Modeling Language) als die Standardsprache zur Modellierung von Software-Systemen entwickelt.

Die UML wird hier auf einigen Folien detaillierter beschrieben.

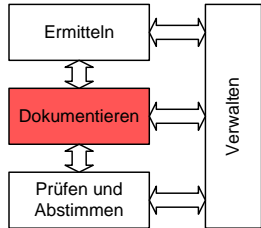


Modellbasiert
dokumentieren

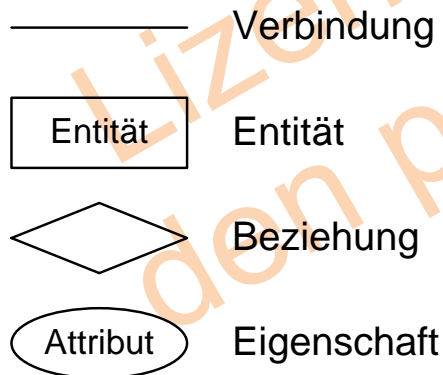
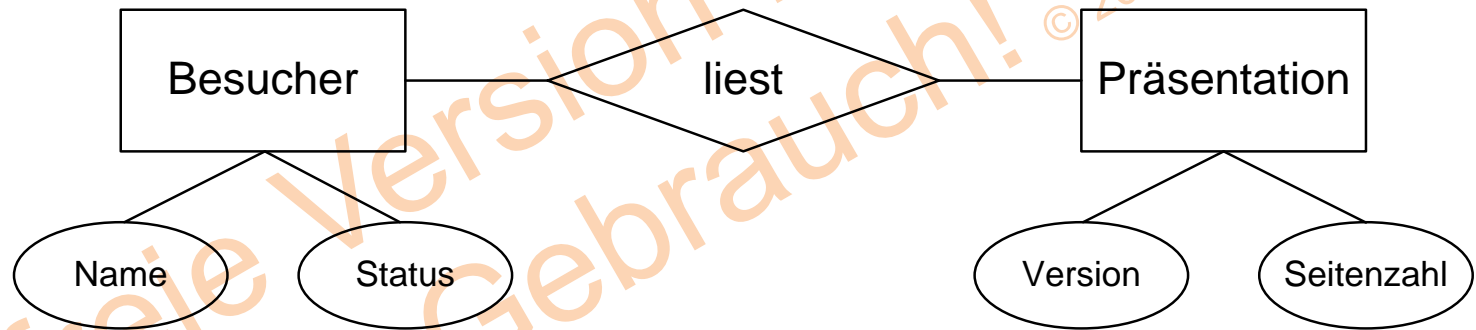
Das Entity-Relationship-Diagramm (ERD) oder Entity-Relationship-Modell (ERM) wurde in den 70er Jahren von Peter Chen für die frühen Entwurfsphasen in der Datenbankentwicklung entwickelt. Das ERD kann aber auch in anderen Bereichen eingesetzt werden und ist aufgrund seiner Einfachheit nach wie vor im Einsatz zu finden.

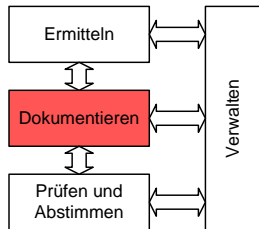
Ein ERD besteht aus folgenden Elementen:

1. Entität (*Entity*): Ein Objekt der realen Welt, für das Informationen zu speichern sind
2. Beziehung (*Relationship*): Verknüpfung zwischen zwei oder mehreren Entitäten
3. Eigenschaft (*Attribute*): Zusätzliche Eigenschaft einer Entität, die in dem gewählten Kontext wichtig oder interessant ist



Modellbasiert
dokumentieren



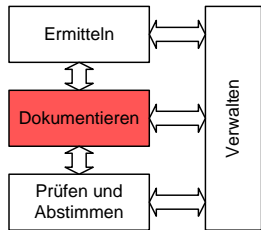


Modellbasiert
dokumentieren

Auszüge aus der Wikipedia /#Wiki-UML/:

„Die Unified Modeling Language, kurz UML, ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen. Sie wird von der Object Management Group (OMG) entwickelt und ist von der ISO (ISO/IEC 19505 für Version 2.1.2) standardisiert. Im Sinne einer Sprache definiert UML dabei Bezeichner für die meisten bei einer Modellierung wichtigen Begriffe und legt mögliche Beziehungen zwischen diesen Begriffen fest. Die UML definiert weiter grafische Notationen für diese Begriffe und für Modelle statischer Strukturen und dynamischer Abläufe, die man mit diesen Begriffen formulieren kann.“

UML ist heute die dominierende Sprache für die Software-System-Modellierung.



Modellbasiert
dokumentieren

Die blau markierten
Diagramme
gehören laut
IREB zum Basis-
Wissen des REs.

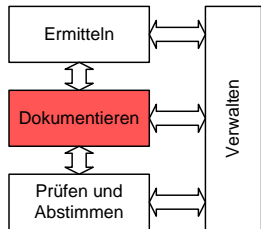
Es werden hier
nur die orange
markierten
Diagramme
beschrieben!

Die UML kennt ab der Version 2.3 sieben Strukturdiagramme /#Wiki-UML/:

1. Das Klassendiagramm, ●
2. das Kompositionsstrukturdiagramm (auch: Montagediagramm),
3. das Komponentendiagramm,
4. das Verteilungsdiagramm,
5. das Objektdiagramm,
6. das Paketdiagramm und
7. das Profildiagramm.

Dazu kommen sieben Verhaltensdiagramme:

1. Das Aktivitätsdiagramm, ●
2. **das Anwendungsfalldiagramm (auch: Use-Case-Diagramm), ● ●**
3. das Interaktionsübersichtsdiagramm,
4. das Kommunikationsdiagramm,
5. das Sequenzdiagramm,
6. das Zeitverlaufdiagramm und
7. das Zustandsdiagramm. ●

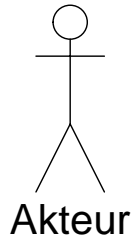
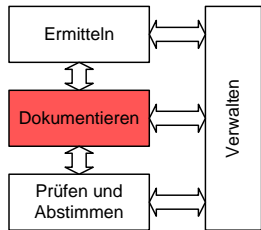


Modellbasiert
dokumentieren

Das bekannteste der UML-Diagramme ist das Use-Case-Diagramm (Anwendungsfalldiagramm). Mit einzelnen Use Cases werden Funktionalitäten eines geplanten oder existierenden Systems aus der Nutzersicht beschrieben.

Ein Use-Case-Diagramm besteht zumindest aus folgenden Elementen:

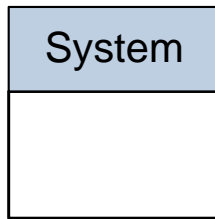
- Einer Systemgrenze
 - Den Use Cases (einer oder mehrere)
 - Den Akteuren (außerhalb des Systems)
 - Den Beziehungen (auch Assoziationen oder Kommunikation) zwischen Akteuren und Use Cases
- und gegebenenfalls
- extend- und
 - include-Beziehungen



benennt die handelnde Person (oder Einheit)

Ein Use-Case-Diagramm kennt sechs Modellelemente: Drei davon beschreiben Objekte und drei Beziehungen.

Modellbasiert dokumentieren



benennt das System (mit den Systemgrenzen)



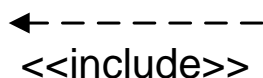
benennt den Anwendungsfall



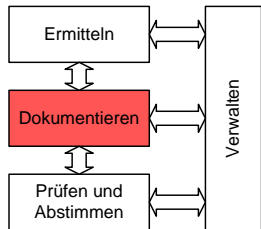
Beziehung



Erweitert

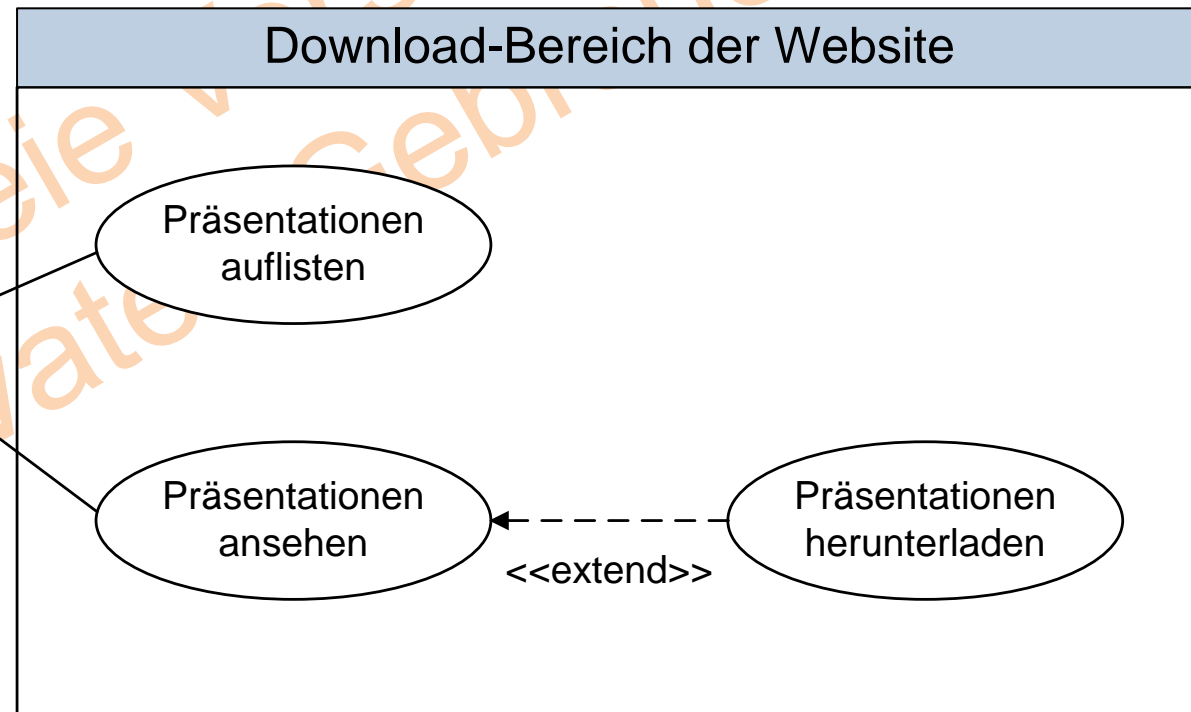


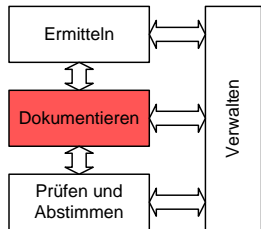
Inkludiert



Im diesem einfachen Beispiel wird der Download-Bereich der Website Peterjohann-Consulting dargestellt. Der Besucher kann dort unmittelbar die (vorhandenen) Präsentationen auflisten und ansehen und bei Bedarf diese auch herunterladen.

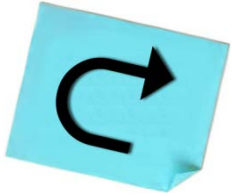
Modellbasiert
dokumentieren





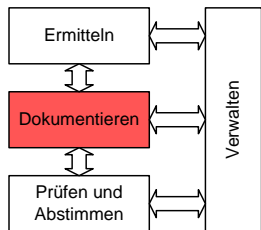
Modellbasiert
dokumentieren

- Verwechseln Sie nicht Use Cases mit Use-Case-Diagrammen: Ein Use-Case-Diagramm kann mehrere Use Cases enthalten
- Die (einzelnen) Use Cases können sowohl tabellarisch wie grafisch dargestellt werden, wobei die grafische Darstellung häufiger zu finden ist
- Beschränken Sie die Anzahl der Use Cases in einem Use-Case-Diagramm (auf nicht mehr als 10), um so die Übersichtlichkeit zu bewahren



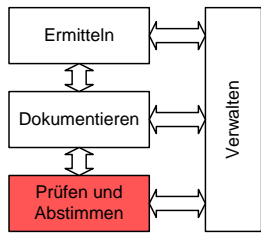
Erstellen Sie ein Use-Case-Diagramm für die Kaffeemaschine (aus den vorhergehenden Übungen).

Dauer:
15 Min.



Modellbasiert
dokumentieren

Keine
Muster-
lösung!



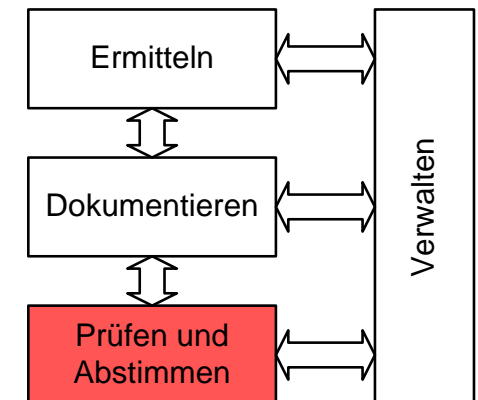
- Das Dokumentieren von Anforderungen ist keine einfache Disziplin und erfordert Kenntnisse, die vorab geschult und geübt werden sollten
- Richten Sie die Dokumentation (auch) am Zielpublikum aus: Fragen Sie vorab bei den Adressaten an, welche Notationen bevorzugt werden
- Verwenden Sie mindestens ein grafisches Modellierungselement (wie das Use-Case-Diagramm)

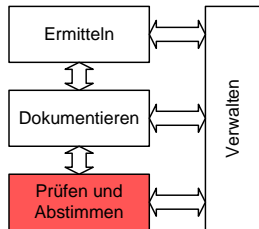


Teil II

Kapitel 6

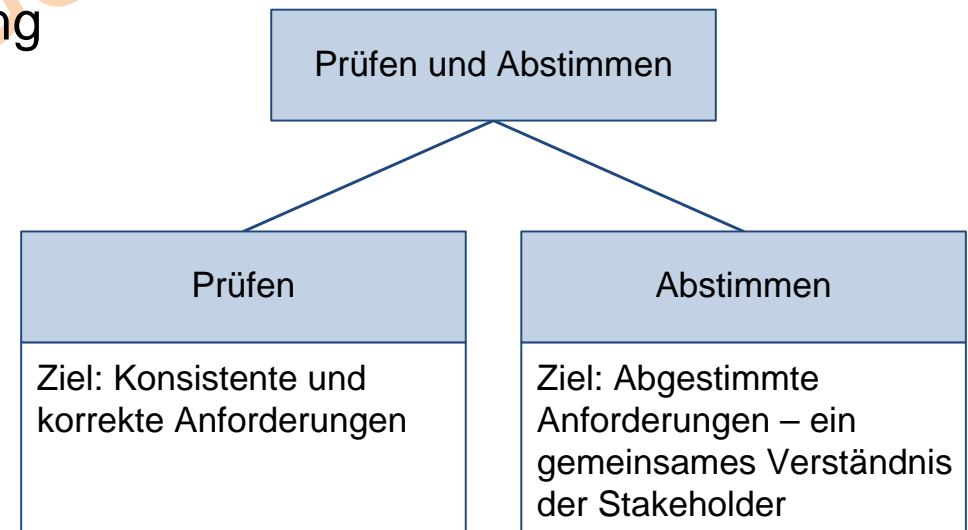
- Ziele beim Prüfen und Abstimmen von Anforderungen
- Prüfungen (Motivation, Organisation, Wer darf prüfen, Sechs Prinzipien)
- Reviews (Einteilung, Stellungnahme, Inspektion, Walkthroughs)
- Checkliste: Ist die Spezifikation formal korrekt?
- Was bedeutet „Abstimmung von Anforderungen“?
- Warum Abstimmungen?
- Anmerkungen und Tipps zum Prüfen und Abstimmen von Anforderungen

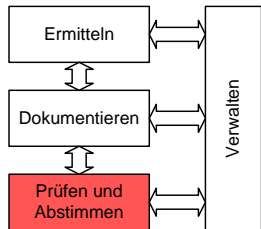




Das Ziel beim Prüfen und Abstimmen ist die Erhöhung und Sicherstellung der Qualität der Anforderungen nach vorgegebenen Qualitätskriterien – dies sind insbesondere:

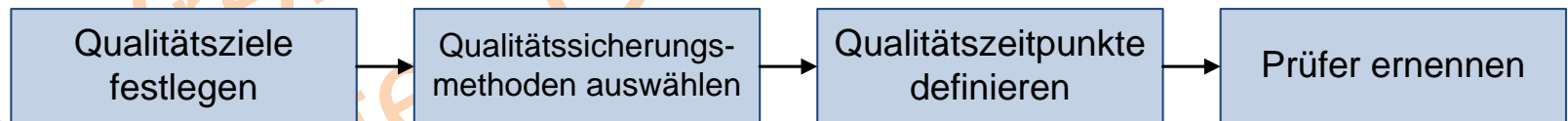
- **Korrektheit:** „Anforderungen sind dann korrekt, wenn sie widerspruchs- und fehlerfrei sind“
- **Abgestimmtheit:** „Eine Anforderung ist dann abgestimmt, wenn Sie für alle Stakeholder korrekt ist und alle Stakeholder sie als notwendige Anforderung akzeptieren“ /IREB15/

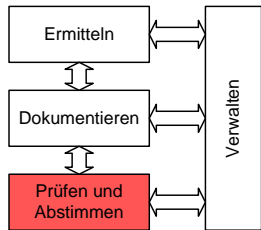




Das Prüfen und Abstimmen bezieht sich immer auf einzelne oder mehrere Anforderungen und kann zu jedem Zeitpunkt durchgeführt werden. Jedoch ist eine Zeitplanung insbesondere wegen der meistens benötigten Stakeholderressourcen sinnvoll.

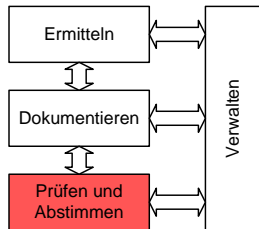
Eine mögliche Vorgehensweise beim Prüfen könnte folgendermaßen aussehen /Rupp16/:





Prüfungen werden durchgeführt um folgendes zu erreichen:

- Abweichungen von der geforderten Qualität der Anforderungen feststellen
- Möglichst viele Fehler, Lücken, Unklarheiten, Mehrdeutigkeiten, etc. in den Anforderungen finden und beheben
- Später: Konflikte zwischen den Wünschen und Forderungen verschiedener beteiligter Personen erkennen und lösen
- Später: Verdeckte Wünsche, Erwartungen und Befürchtungen erkennen und thematisieren



Wann wird geprüft?

- Fortlaufend (informale Prüfung oder Reviews)
- Zu bestimmten Zeitpunkten (formale Reviews), spätestens bei Fertigstellung der Spezifikation oder des Produkts

Prüfzeitpunkt

Wer darf prüfen?

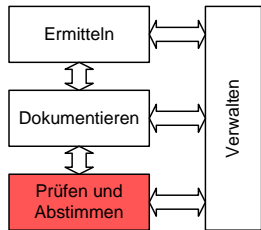
- Der Requirements Engineer selbst
- Mitglieder des Projektteams (nächste Folie)
- Der Kunde

Prüfer

Was wird geprüft?

- Die Dokumentation
- Der Projektplan (für die Erstellung der Dokumente oder des Produkts)
- Die Vorgehensweise
- Weitere Ergebnisse

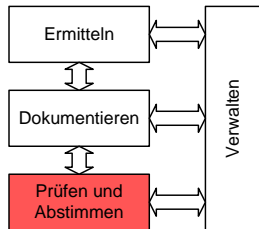
Prüfobjekt



++	Sehr gut geeignet
+	Gut geeignet
o	Gar nicht geeignet

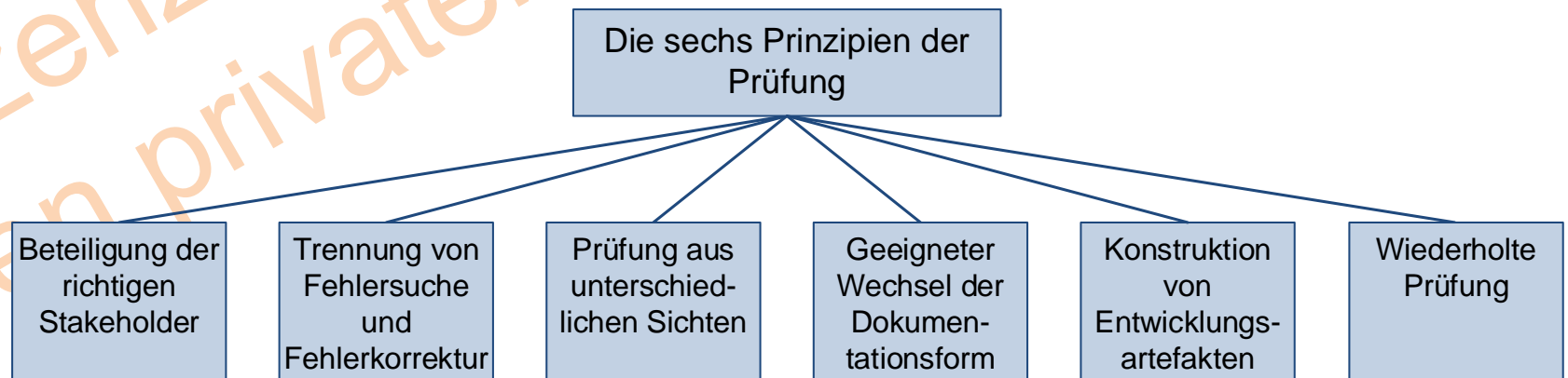
	Fachliche Richtigkeit	Auftrag des Projekts / Notwendigkeit / rechtliche Verbindlichkeit	Vollständigkeit	Verständlichkeit, gute Struktur	Eindeutigkeit, Widerspruchsfreiheit und Konsistenz	Traceability / Nachvollziehbarkeit	Realisierbarkeit	Testbarkeit	Kritikalität
Anwender	++	+	++	++	o	o	o	o	+
Fachlicher Verantwortlicher	++	++	o	o	o	o	o	o	++
Analytiker	o	o	++	+	++	++	+	+	o
Projektleiter	+	++	o	+	o	o	o	o	+
Entwickler	o	o	o	o	+	++	++	o	o
Product Owner	+	+	++	+	o	o	o	o	+
Scrum Developer	o	o	+	+	+	++	++	++	o
Tester	o	o	+	+	++	++	o	++	o
Projektexterne Auditoren	o	o	+	+	+	+	o	+	o

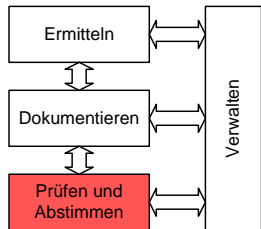
/Rupp14/



Das IREB /IREB15/ benennt folgende sechs Prinzipien der Prüfung von Anforderungen:

1. Beteiligung der richtigen Stakeholder
2. Trennung von Fehlersuche und Fehlerkorrektur
3. Prüfung aus unterschiedlichen Sichten
4. Geeigneter Wechsel der Dokumentationsform
5. Konstruktion von Entwicklungsartefakten
6. Wiederholte Prüfung



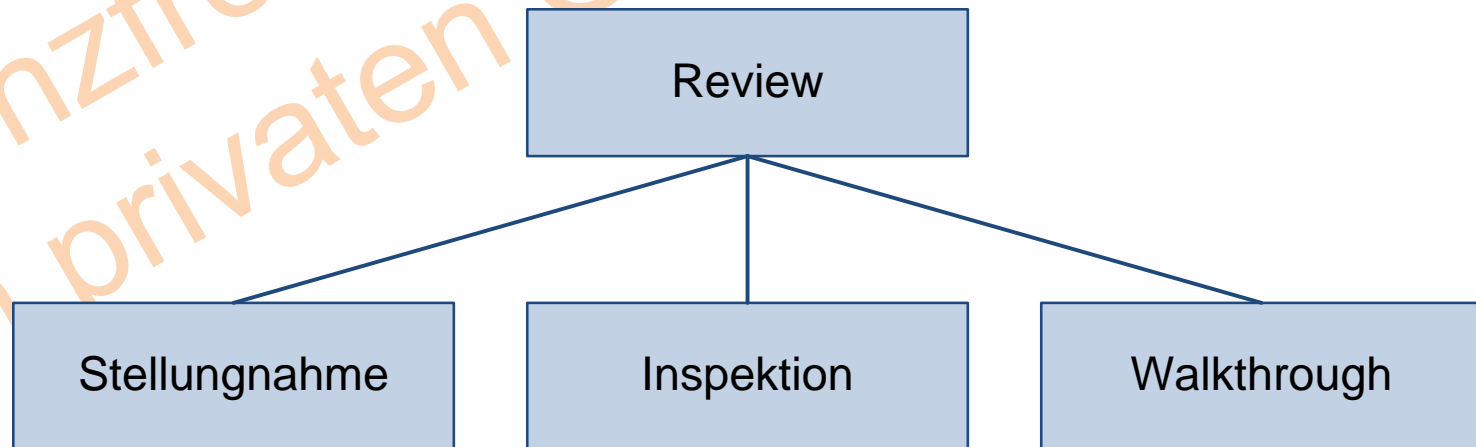


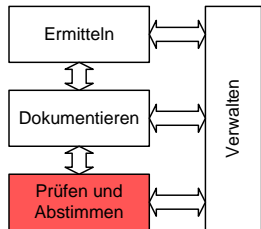
Die manuellen Techniken zur Prüfung werden beim IREB /IREB15/ zusammenfassend als Reviews bezeichnet.

Dabei sind ...

- die Stellungnahme,
- die Inspektion
- und der Walkthrough

unterschiedliche Varianten.

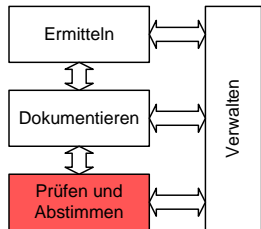




Bei der Stellungnahme werden einzelne Stakeholder um eine schriftliche Stellungnahme zum Inhalt und zur Qualität von einzelnen oder mehreren Anforderungen gebeten. Dabei verteilt der Requirements Engineer die zu überprüfenden Anforderungen an die involvierten Stakeholder.

Technisch wird die Stellungnahme durch Kommentare an den entsprechenden Textstellen realisiert. Mit Abschluss der Stellungnahme erhält der Requirements Engineer die einzelnen Stellungnahmen zurück und arbeitet die Kommentare ein.

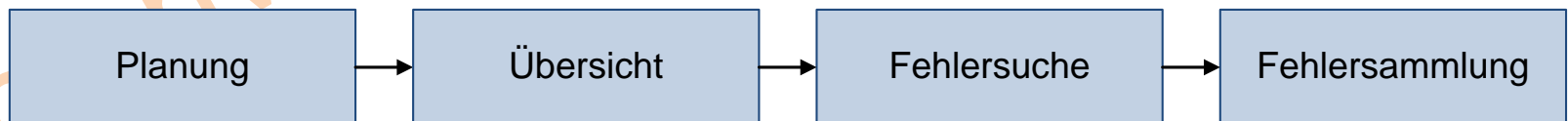
Die Stellungnahme ist (zunächst) ein sehr einfaches Verfahren, welches jedoch nicht immer zuverlässig funktioniert, da je nach Vorkenntnis die Qualität der Anforderungen der Stakeholder stark schwanken kann.

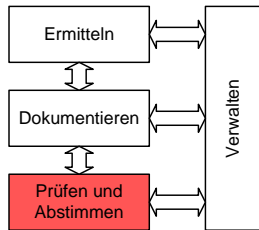


Die Inspektion ist die formalste und aufwendigste der Prüftechniken. Es werden Prüfungen anhand formaler (Qualitäts-)Kriterien und zugehöriger Checklisten durchgeführt und dann anschließend ausgewertet.

Bei der Inspektionssitzung gibt es verschiedene Rollen – dies sind: Organisator, Moderator, Autor, Vorleser, Inspektoren und Protokollant. Der Moderator sollte neutral sein und kann auch gleichzeitig als Vorleser fungieren.

Die Inspektion folgt einem (vierstufigen) Prozessschema, welches bei der Fehlersammlung endet: Die Fehlerkorrektur und Nachkontrolle gehört nicht (unbedingt) zur Inspektion.

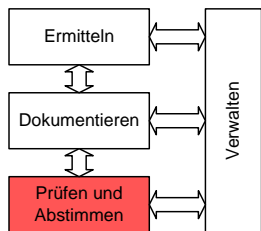




Beim Walkthrough wird in einer Sitzung durch den Moderator (in der Regel der Requirements Engineer) das Anforderungsdokument vorgestellt. Dabei werden Fragen durch die anwesenden Reviewer (Gutachter) gestellt und diese wiederum durch den Moderator beantwortet. Die Antworten werden durch den Protokollanten protokolliert und später in die Anforderungen / das Anforderungsdokument eingearbeitet.

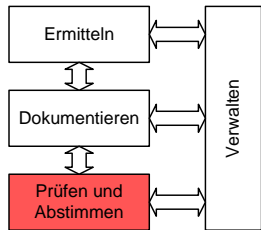
Der Walkthrough ist eine schnelle Methode, um erste Fehler und Unklarheiten zu erkennen.

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/

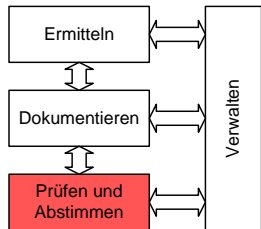


Bei der Abstimmung von Anforderungen geht es darum, die Anforderungen in die Umsetzung zu bringen. Hierzu müssen die beteiligten Interessengruppen (Stakeholder) ihre Zustimmung zur Umsetzung der einzelnen Anforderungen geben. Dies geschieht gewöhnlich in einer Abstimmungsrunde eines Kontrollgremiums. Das Ergebnis sind eine einzelne oder mehrere Bündel von Anforderungen, die dann in die Umsetzung gelangen (können).

Achtung:

Beim Abstimmen von Anforderungen kann es sein, dass sich die beteiligten Stakeholder nicht einigen können oder gar Konflikte auftreten.

Zum **Konfliktmanagement** (in Projekten) gibt es eine Präsentation des Autors, die auf der Website unter https://www.peterjohann-consulting.de/_pdf/peco-pm-konfliktmanagement.pdf frei herunterladbar ist.

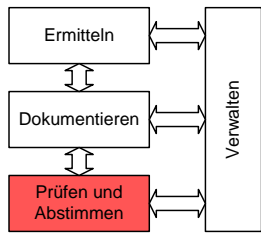


Projektprobleme resultieren häufig aus Fehlern, die entstehen, weil „man“ meint, mit den bereits erfassten Anforderungen „schon mal starten“ zu müssen. Die Folgen oder Merkmale sind dann /Ebert12/:

- Anforderungen sind nicht gut genug, aber man startet, um weiterzukommen
- Anforderungen schleichen sich in das Projekt unkontrolliert ein
- Anforderungen werden ohne Analyse, Prüfung und Vereinbarung zugesichert



Das Prüfen und Abstimmen wird in der Praxis häufig nur sehr rudimentär durchgeführt und führt dann zu unzureichenden Ergebnissen.



Daher:

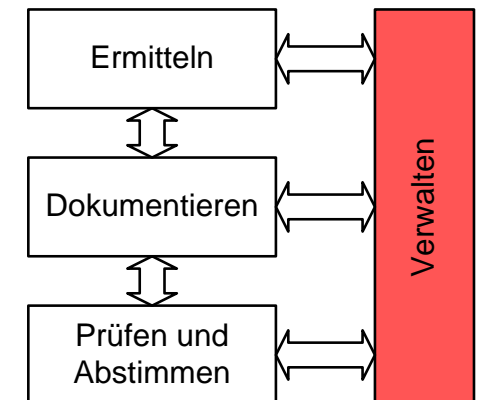
- Machen Sie bei Ihrem Vorhaben die gewünschten / notwendigen Prüfungen mit den Prüfprozessen für alle (potenziellen) Beteiligten transparent
- Werben Sie für das Prüfen und Abstimmen
- Planen Sie unbedingt Zeit für das Prüfen und Abstimmen ein und stimmen Sie diese mit den Stakeholdern ab. Unterscheiden Sie dabei zwischen Dauer und Aufwand
- Klären Sie im Vorfeld, was passieren soll, falls Anforderungen nicht geprüft oder abgestimmt werden können (Eskalations- und Einzelentscheidungsmechanismen)

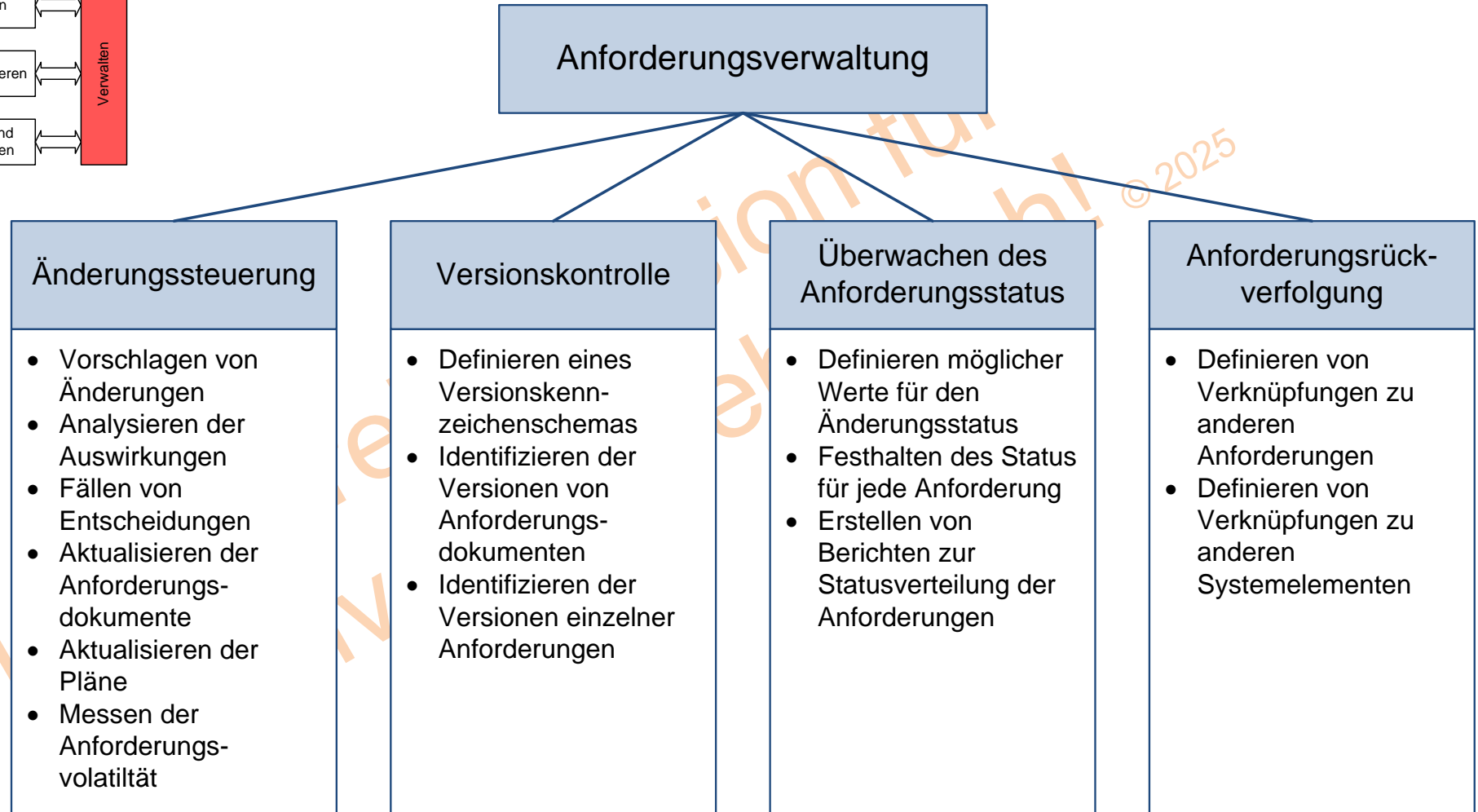
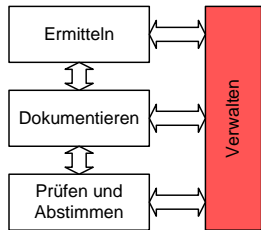


Teil II

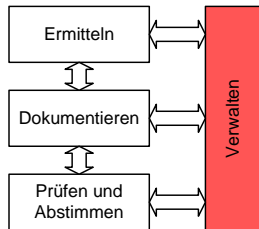
- Aufgaben der Anforderungsverwaltung
- Aufgaben der Anforderungs-/Änderungsverwaltung
- Der Anforderungsstatus (Grundsätzliches, Das Zustandsmodell, Grafische Darstellung, Tabellarische Darstellung)
- Übung 4: Zustandsmodell
- Änderungen (Ursachen, Abschätzformel für den Aufwand, Wann wird es kritisch?)
- Anforderungsrückverfolgung
- Versionierung
- Konfigurationsmanagement
- Baselines
- Anmerkungen und Tipps zum Verwalten von Anforderungen

Kapitel 7

Seite
131–146



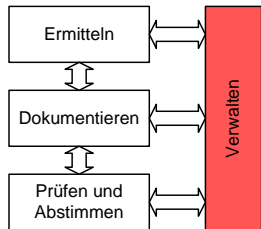
/Wieggers05/



- Erfassen der Änderungsanforderungen (in einem definierten, systematischen Prozess)
- Möglichkeit der Bewertung der Änderungsanforderungen
- Planung der Umsetzung der Änderungsanforderungen
- Etablierung einer (für alle Beteiligten) transparenten Versionskontrolle von Anforderungen und Anforderungsdokumenten

Anmerkungen:

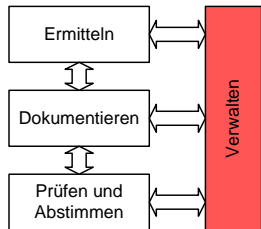
- Anforderungen und Änderungen (an Anforderungen) werden hier gleichgesetzt
- Die Anforderungsverwaltung wird zumeist mit dem Einsatz eines Anforderungsverwaltung-Tools (auch Software- oder RE-Tool genannt) gleichgesetzt – dies ist in der Regel auch sinnvoll



Jede Anforderung sollte systematisch bearbeitet werden, damit ...

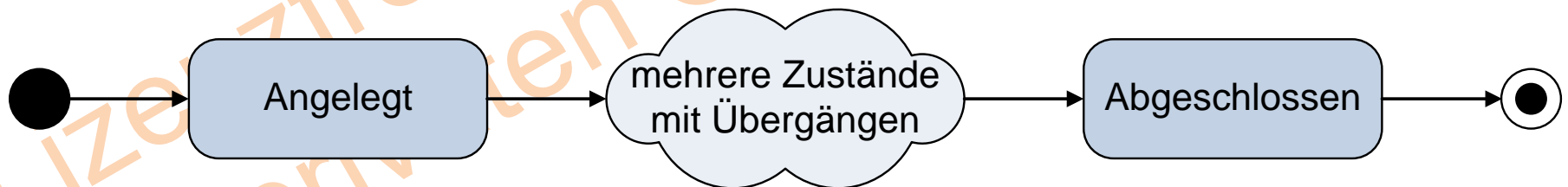
- alle Anforderungen tatsächlich erfasst werden und keine Anforderung verloren geht.
- alle erfassten Anforderungen einen definierten Endstatus erhalten, sie also nicht irgendwo „hängenbleiben“.
- die Anforderungen einer Genehmigungs- oder Kontroll-Instanz vorgelegt werden können, die über das weitere Vorgehen entscheidet.

Der (Bearbeitungs-)Status von Anforderungen mit den entsprechenden Übergängen häufig über ein „Zustandsmodell“ (Zustandsgraph) beschrieben (siehe nächste Folien).

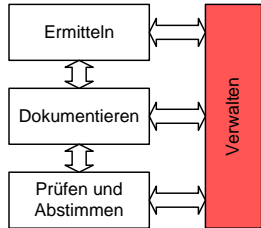


Über ein Zustandsmodell erhält eine Anforderung immer genau einen Status, an dem erkennbar ist, wie der Stand der Bearbeitung dieser Anforderung ist. Um von einem Status in den nächsten zu gelangen, muss „irgendetwas“ passieren (z.B. Abnahme durch die Qualitätssicherung, Vorlage beim Steuerungsgremium).

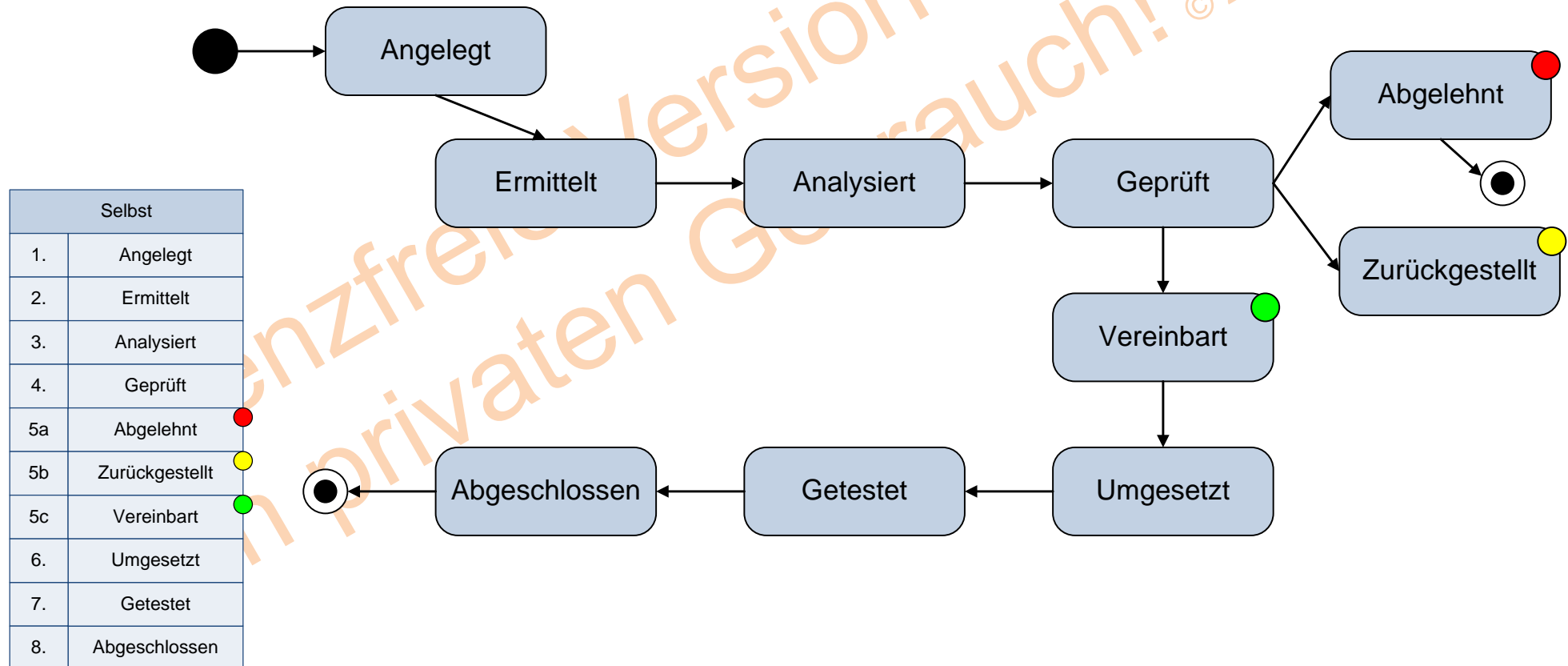
Ein Zustandsmodell gibt den RE-Prozess vor und sollte daher mit Bedacht eingeführt und verwendet werden.

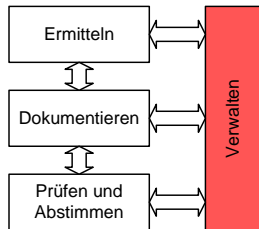


Eine Löschung wird nicht vorgenommen: Einmal angelegte Anforderungen bleiben erhalten. Daher ist es wichtig, dass nur autorisierte Personen Anforderungen anlegen dürfen.

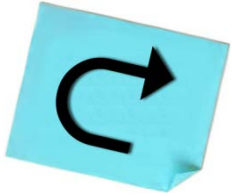


Ein Zustandsmodell mit 8-10 Zuständen und einer Entscheidungsstufe (nach „Geprüft“) ist in der Praxis häufig zu finden.



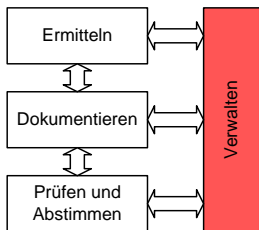


	Zustand	Beschreibung	
1.	Angelegt	Die Anforderung wurde angelegt	
2.	Ermittelt	Die Anforderung wurde gemäß IREB-Verfahren weiter beschrieben	
3.	Analysiert	Die Anforderung wurde gemäß IREB-Verfahren analysiert	
4.	Geprüft	Die Anforderung wurde gemäß IREB-Verfahren geprüft	
5a	Abgelehnt	● Die Anforderung wurde einem Kontroll-Gremium vorgelegt	Ablehnung. Anforderung wird nicht weiter verfolgt
5b	Zurückgestellt		Zurückstellung. Anforderung wird zurzeit nicht weiter betrachtet
5c	Vereinbart		Zustimmung. Die Umsetzung kann erfolgen
6.	Umgesetzt	Die Umsetzung der Anforderung ist erfolgt	
7.	Getestet	Die Anforderung wurde abschließend getestet	
8.	Abgeschlossen	Die Anforderung wurde abgeschlossen (z.B. durch Abnahme), d.h. es bedarf keiner weiteren Bearbeitung	

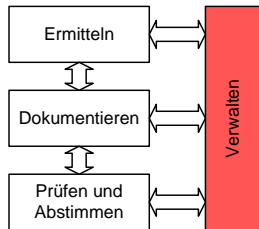


Erstellen Sie ein Zustandsmodell für die Anforderungsverwaltung in Ihrer Organisation.

Dauer:
20 Min.



Keine
Muster-
lösung!

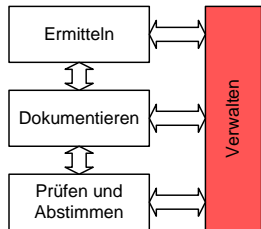


Die inneren Ursachen für Änderungen sind:

- Nicht-Erfassen von Anforderungen zu Beginn des Projekts
- „Vergessen“ einiger Anforderungen
- Teile der Anforderungen (und Annahmen) sind nicht umsetzbar

Aufgrund vielfältiger, äußerer Ursachen unterliegen Anforderungen einer Veränderung:

- Fortschritte in der Technologie
- Änderung der unternehmensinternen Organisation
- Veränderte Bedürfnisse von Kunden
- Veränderung von Märkten
- Neue oder geänderte politische oder rechtliche Rahmenbedingungen



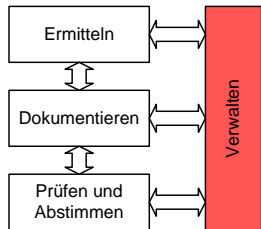
„2 bis 5 % des Gesamtaufwands werden pro Monat für Änderungen benötigt“

Rechenbeispiel:

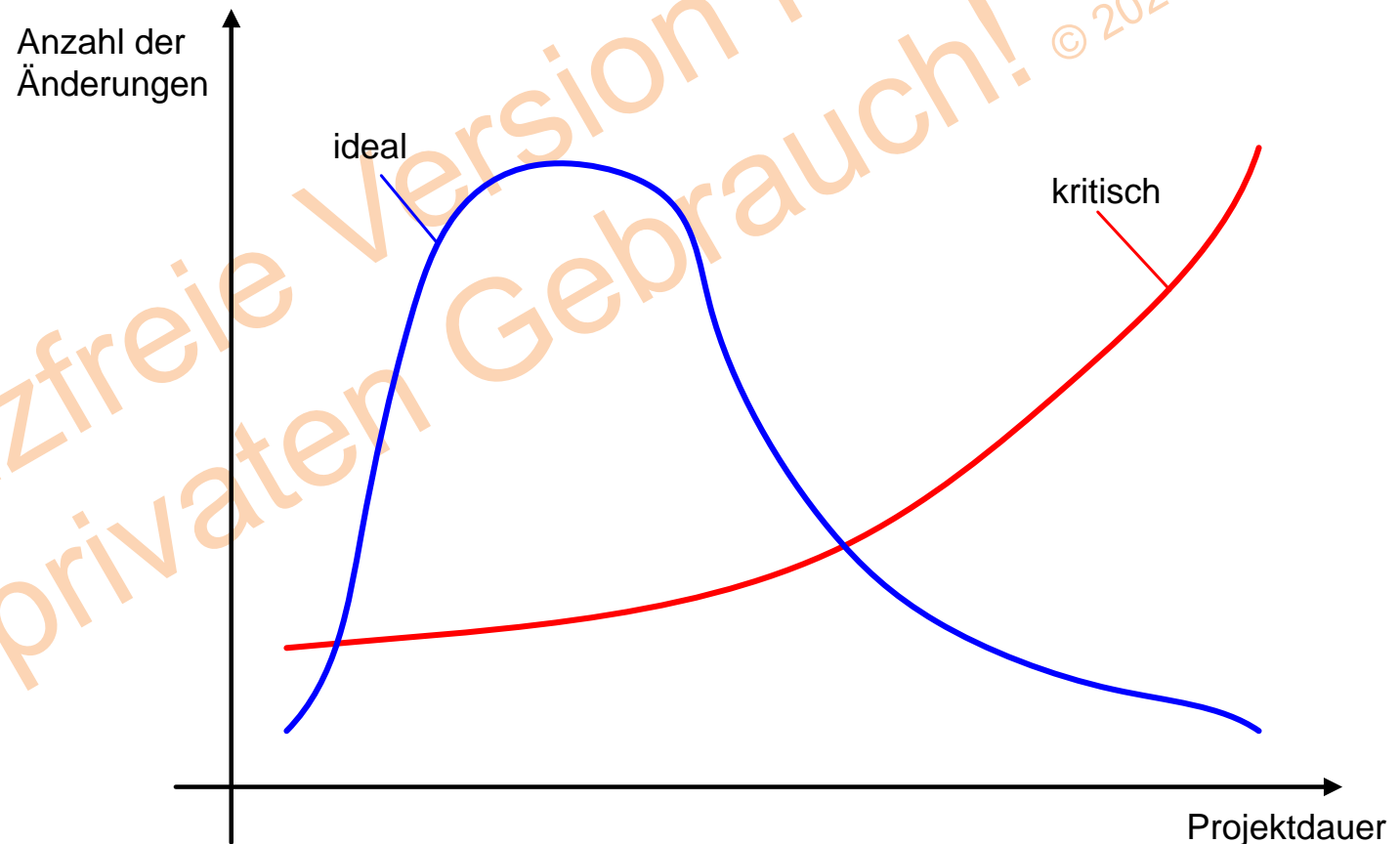
- Ein Projekt mit 120 Personenwochen Gesamtaufwand, 3 Projektmitarbeiter und damit etwa 1 Jahr Entwicklungszeit
- 5 % bedeutet 6 Wochen Aufwand pro Monat für die Änderungsbearbeitung, d.h. mehr als die Hälfte der 10 Personenwochen (pro Monat) könnten für die Änderungen benötigt werden

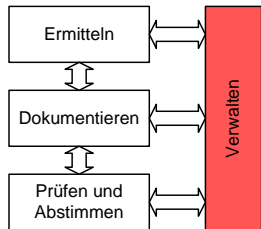
Daher:

- Möglichst die Anforderungen stabil halten
- Änderungen nur kontrolliert einfließen lassen

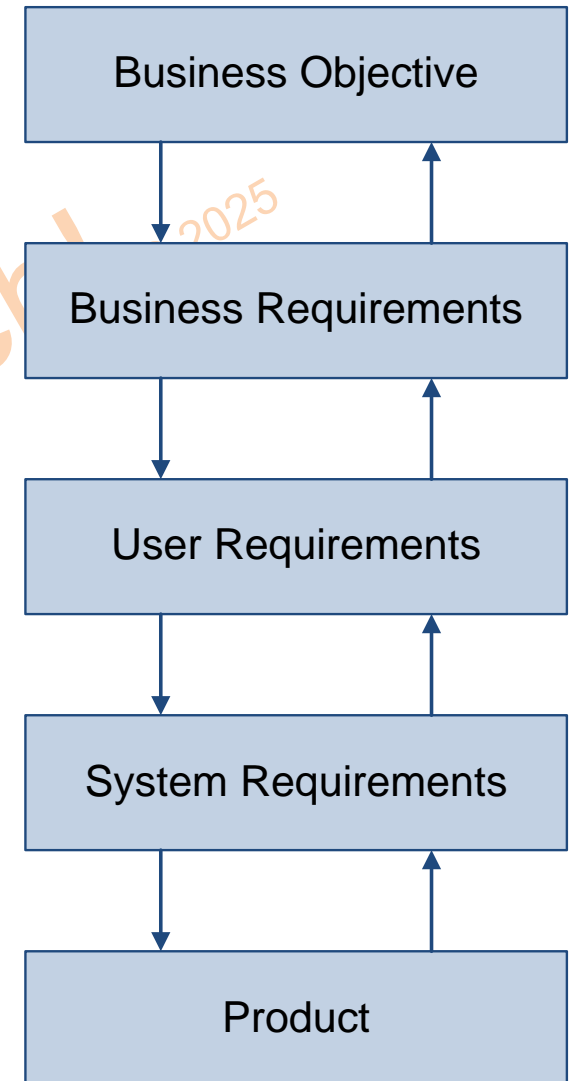


Ab einer bestimmten Anzahl von Änderungen ist das Projektteam fast ausschließlich damit beschäftigt, die Änderungen einzuarbeiten. Ein Projektfortschritt ist dann kaum noch möglich.

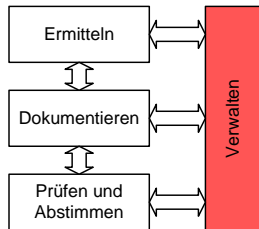




Durch das „Rückverfolgen“ (engl. *Tracing*) von Anforderungen wird gesichert, dass jede einzelne Anforderung den entsprechenden RE-Produkten zugeordnet werden kann. Dadurch wird gewährleistet, dass jederzeit geklärt werden kann, welche übergeordneten Ziele (*Business Objectives*) einer umgesetzten Produktanforderung zugrunde liegen.



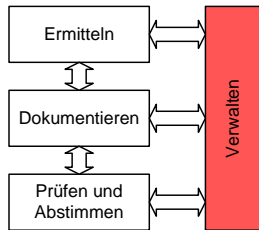
/Jonasson12/



Um festzuhalten, wann sich welche Anforderungen im Laufe der Ermittlung und Umsetzung verändert haben, sollten diese versioniert werden. Im Idealfall wird jede einzelne Anforderung versioniert (d.h. hat eine eigene Versionsnummer); ebenso können zugehörige Artefakte wie Code-Segmente, Test-Skripte oder Prüfprotokolle versioniert werden.

Dabei werden nach Veränderungen die Versionsnummern (in Inkrementen) hochgezählt.

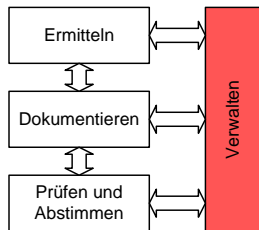
Die Versionierung wird insbesondere dann wichtig, wenn eine Anforderung einen „gewissen Status“ erlangt hat – in unserem Zustandsmodell (siehe einige Folien vorher), wenn sie „vereinbart“ worden ist, da dann mit der Umsetzung begonnen wird.



Die einzelnen Anforderungen mit ihren Versionsnummern sollten als Ganzes zentral gesammelt und abgelegt werden. Versieht man diese Zusammenstellungen jeweils mit einer eigenen Versionsnummer so spricht man von Konfigurationen. Die Disziplin, die sich mit der Erstellung und Verwaltung von Konfigurationen beschäftigt, heißt Konfigurationsmanagement.

Bei einer großen Anzahl von Anforderungen erfolgt in der Praxis die Ablage von Konfigurationen in datenbankgestützten Versions- und Konfigurationsmanagementsystemen.

Das Konfigurationsmanagement wird häufig zusammen mit dem Änderungsmanagement betrachtet und ist auch eine Teildisziplin des Projektmanagements – Konfigurationen können eine Schnittstelle oder einen Übergabepunkt vom Requirements Engineering zum Projektmanagement darstellen.



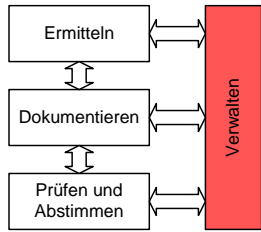
Eine Baseline (der deutsche Begriff Basislinie wird seltener verwendet, auch der Begriff Release kann benutzt werden) stellt eine Zusammenstellung (oder Konfiguration) von (allen) Anforderungen zu einem Zeitpunkt dar, die dann eine zentrale Versionsnummer erhält.

Pro Anforderung wird festgehalten:

- Das Datum
- Die Versionsnummer

Eine Baseline gibt in der Regel einen „abgenommen Stand“ (geprüft und abgestimmt) wieder, der zur weiteren Bearbeitung gelangt.

In der Regel ist es nicht wünschenswert, häufig neue Baselines zu erstellen, da jede Baseline komplett „abgenommen“, d.h. überprüft und abgestimmt sein sollte.



- Überlegen Sie frühzeitig, welche Software-Tools sie bei der Verwaltung von Anforderungen einsetzen möchten. Ganz ohne Software-Tool-Unterstützung ist das Requirements Management nur bei kleinen Projekten einsetzbar
- Spielen Sie vor dem Start des Vorhabens (mit den wesentlichen Beteiligten) durch, wie Sie mit Versionierung, Konfigurationen, Baselines und Releases umgehen wollen

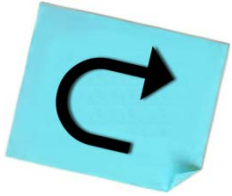


Teil II

- Übung 5: Beliebte Aktivitäten
- Fragen zum Teil II

Kapitel 8

Seite
147–150



Überlegen Sie, welche der vier Aktivitäten der Anforderungs-entwicklung besonders gerne von Ihnen / von anderen Requirements Engineers ausgeübt werden.

Was bedeutet das?

Dauer:
20 Min.

Keine
Muster-
lösung!



1. Nennen Sie die vier Hauptaktivitäten des REs und ihre Aufgaben!
2. Was sind die Stärken und Schwächen (Vor- und Nachteile) bei der Verwendung natürlicher Sprache im RE?
3. Was sind die Stärken und Schwächen (Vor- und Nachteile) bei der Verwendung von Modellen im RE?
4. Wozu dient ein Zustandsmodell für Anforderungen?
5. Wie lange kann eine Anforderung in einem bestimmten Status bleiben?
6. Warum ist die Rückverfolgbarkeit (Traceability) von Anforderungen wichtig?



7. Welche Aufgaben sollte ein Anforderungstool erfüllen?
8. Wie hängen die Versionierung und das Konfigurationsmanagement zusammen?
9. Was ist eine Baseline?
10. Wann darf eine Baseline verändert oder neu erstellt werden?

Lizenzfreie Version für
den privaten Gebrauch! © 2025



Teil III

- Kapitel 9 Tools für das Requirements Engineering
- Kapitel 11 Agiles Requirements Engineering
- Kapitel 12 Die Einführung von Requirements Engineering
- Kapitel 13 Weitere Praxis-Aspekte
- Kapitel 14 Fragen und Übungen zum Teil III

Teil III

Seite
151–188



Teil III

- Aufgaben von RE-Tools
- Anforderungen an RE-Tools
- Aufbau eines RE-Tools
- Einbettung eines RE-Tools
- Basismerkmale von RE-Tools
- Zur Einführung eines RE-Tools
- Hersteller (Auswahl)

Kapitel 9

Seite
152–159



Die wesentlichen Aufgaben eines RE-Tools sind ...

- die Aufnahme der Anforderungen,
- die ID-Vergabe,
- die Priorisierung (Kostenabschätzung, Risikobewertung),
- die Etablierung der Nachvollziehbarkeit der Anforderungsverwendung (woher kommt eine Anforderung, wo wird sie verwendet),
- die Verfolgung von Abhängigkeiten der Änderungen (Tracing) und
- die Historisierung (wer hat wann was warum wie verändert).

Somit:

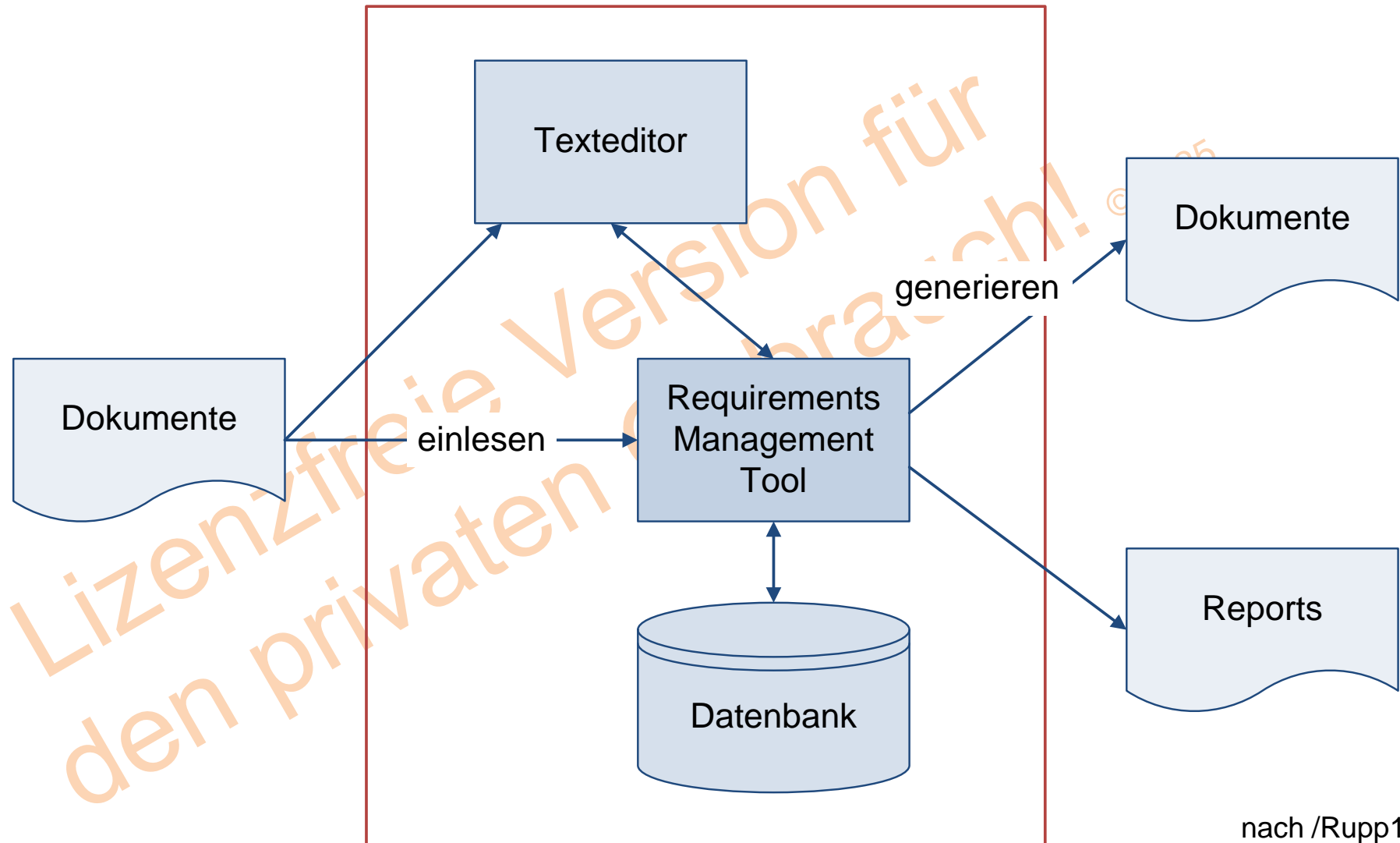
- Zur reinen Aufnahme von Anforderungen (ohne Änderungshistorie) reicht auch ein Tabellenkalkulationsprogramm

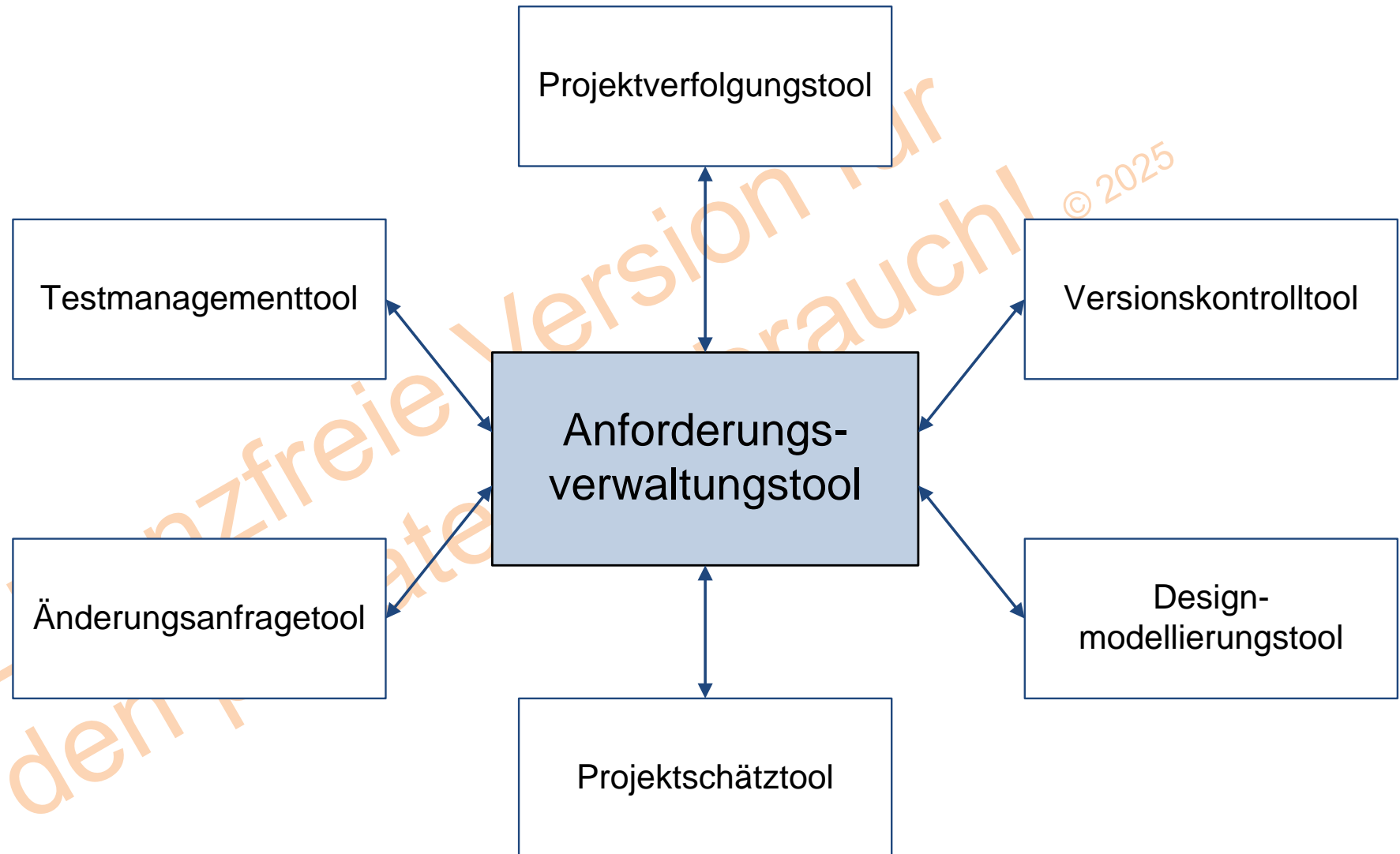


Sinnvoll sind weiterhin folgende Anforderungen an ein RE-Tool:

- Die Benutzung von Rollen
- Die Verwendung von Berechtigungen
- Das Erstellen von Berichten
- Das Erstellen von Glossar, Liste der verbotenen Wörter, Synonyme

Lizenzfreie Version für
den privaten Gebrauch! © 2025







Die wesentlichen Merkmale eines RE-Tools sind:

- Die Anforderungen sind eindeutig identifizierbar
- Es erfüllt die wesentlichen Aufgaben
- Es ist datenbankbasiert
- Es besitzt Multi-User- und Multi-linguale Fähigkeiten

Weitere Merkmale:

- Eigenschaften müssen über (frei definierbare) Attribute festgehalten werden können



- Die Auswahl und Einführung eines RE-Tools ist eine umfangreiche Aufgabe; typischerweise muss man hierfür mehrere Monate einplanen
- Mit der Einführung eines RE-Tools müssen auch die RE-Prozesse umgestaltet werden
- Ein „Tool für alle“: Alle Projektbeteiligten müssen mit dem Tool arbeiten dürfen und können
- Ein „Tool für alles“: Das Tool muss für alle Anforderungen eingesetzt werden – „Nebentools“ sind nicht erlaubt
- Einen Kriterienvergleich einiger RE-Tools liefert /Hood07/

Aber:

- Prozesse & Kommunikation müssen unabhängig vom Tool implementiert werden
- „Tools schreiben keine Anforderungen und etablieren keine Prozesse“



Tool	Firma	Bemerkung
DOORS	IBM	Marktführer; früherer Hersteller: Telelogic
CaliberRM	Microfocus	Früherer Hersteller: Borland
Polarion	Siemens	Früherer Hersteller: Polarion
IrQA	Visure Solutions	Früherer Hersteller: QA-Systems
objectiF RM	microTOOL	Preiswertes Tool
Jama	jamasoftware	
CARE	SOPHIST	Templates, kostenfrei
Volere	Robertson	Templates, kostenfrei

Eine umfassendere Liste liefert beispielsweise [/#RE-Software-Tools/](#).



Teil III

Kapitel 10

- Motivation für Agiles Requirements Engineering
- Das magische Dreieck beim klassischen und Agilen Requirements Engineering
- Unterschiede klassisches und Agiles Requirements Engineering
- Einige zentrale Begriffe im Agilen Requirements Engineering
- Gemeinsamkeiten klassisches und Agiles Requirements Engineering
- Was macht der Requirements Engineer im agilen Kontext?
- Das Product Backlog (Beschreibung, Aussehen)
- User Stories (Grundsätzliches, INVEST, Story Maps, Epics und Themes)
- Weitere Anmerkungen zum Agilen Requirements Engineering
- Tipps zum Agilen Requirements Engineering

Seite
160–174



Beobachtung:

„Etwa zwei Drittel aller umgesetzten Anforderungen in (großen) SW-Systemen werden selten oder nie benötigt.“

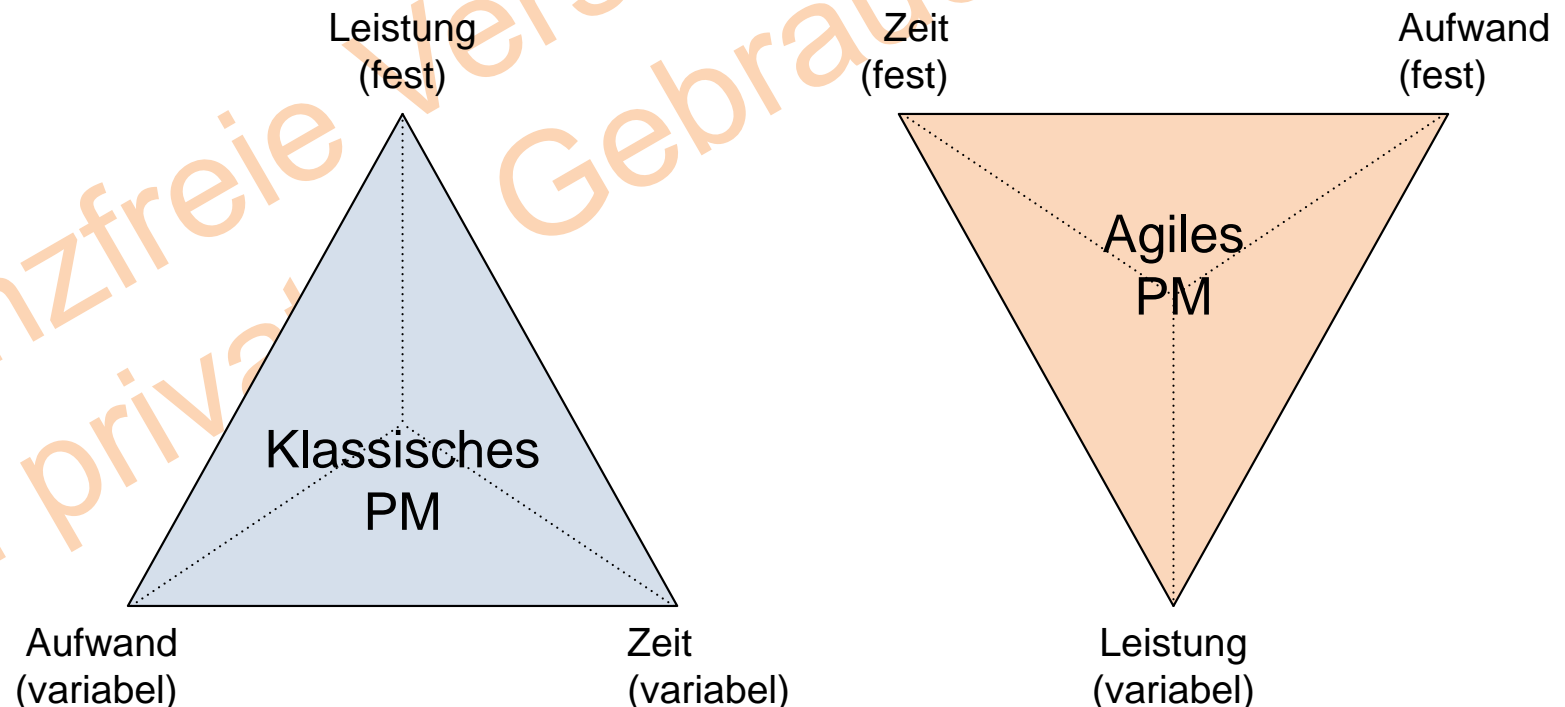
Daher:

Beim „Agilen Projektmanagement“ wird im Vorfeld nicht das gesamte Projekt durchgeplant, sondern nur die Themen, die im nächsten Schritt benötigt werden und dann auch unmittelbar zur Umsetzung gelangen. Hierdurch wird nur das im Detail (aufwendig) geplant, was tatsächlich für den nächsten Umsetzungsschritt benötigt wird.

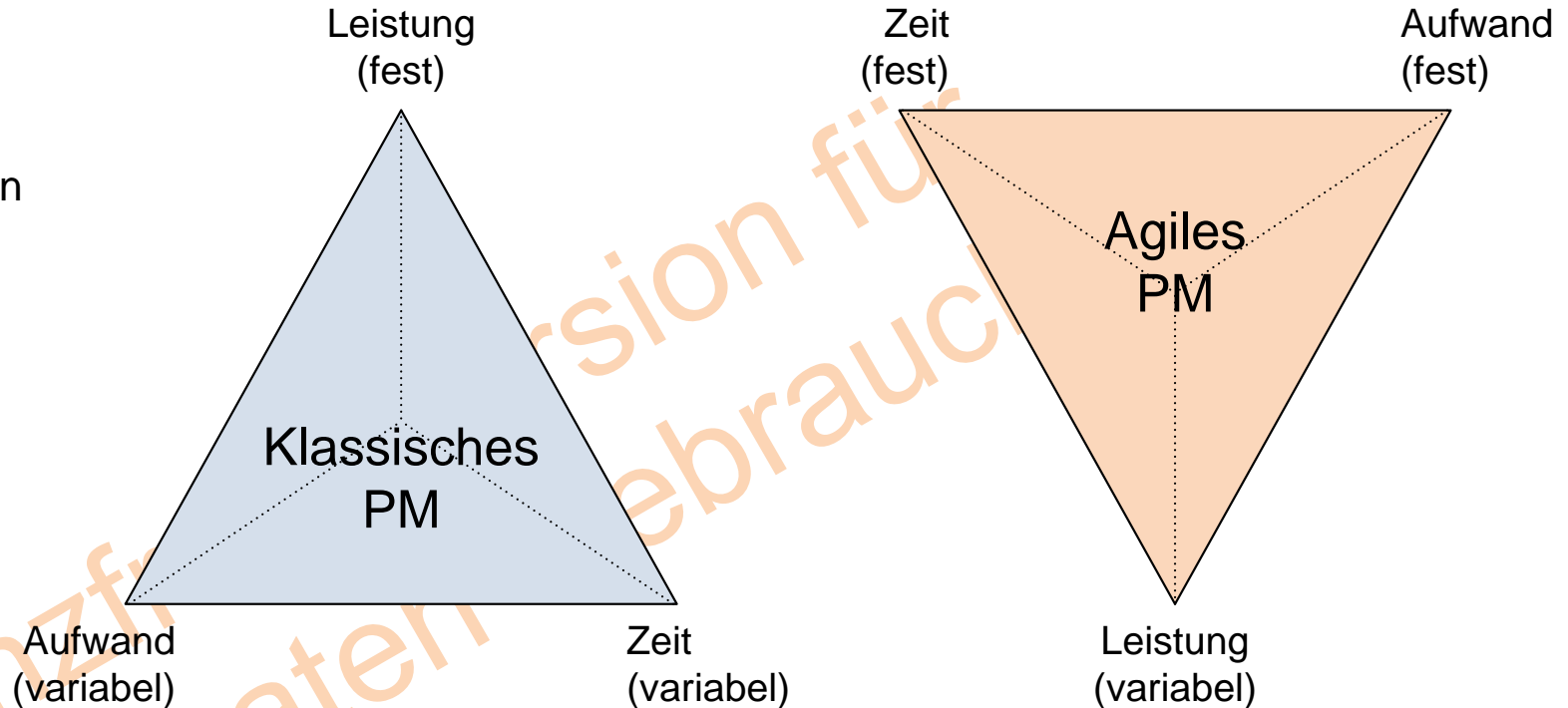
Somit muss ein Agiles Requirements Engineering (kurz: ARE) anders funktionieren als klassisches („traditionelles“) Requirements Engineering.

Zur **Agilität und zu Scrum** gibt es einige Präsentationen des Autors, die ebenfalls auf der Website unter <https://www.peterjohann-consulting.de/scrum> frei herunterladbar sind. Ebenso gibt es dort eine Übersicht zum **Agilen Requirements Engineering**.

Der Zusammenhang von Leistung, Aufwand und Zeit wird im Projektmanagement häufig durch das magische Dreieck (engl. „*Iron Triangle*“ oder „*Triple Constraint*“) beschrieben: Während beim klassischen PM der Leistungsumfang die einzig feste Größe ist, wird beim Agilen PM auf feste Zeit(abschnitte) und festen Aufwand Wert gelegt.



Aufwand = Kosten
und Mitarbeiter



- Alle Anforderungen (die die Leistung beschreiben) sind vollständig definiert und festgeschrieben
- Der Zeitraum ist geplant
- Der Aufwand kann erhöht werden, um die Fertigstellung zu beschleunigen

- Die Anforderungen (nicht unbedingt alle) sind grob beschrieben und teilweise priorisiert
- Im festgelegten Zeitraum werden die wichtigsten Anforderungen umgesetzt
- Der Aufwand ist festgelegt und wird gesteuert, um innerhalb des Aufwandrahmens zu bleiben



Beim klassischen Requirements Engineering ...

- werden Anforderungen als Ganzes möglichst vollständig vor Projektbeginn (oder vor Beginn einer Projektphase) ermittelt.
- orientiert sich der Entwickler an den Anforderungen, die er (technisch) verstehen muss.
- sind Änderungen vergleichsweise schwer und aufwendig einzubringen.

Beim Agilen Requirements Engineering ...

- werden Anforderungen kurzfristig ermittelt.
- wird die Anforderungsermittlung kontinuierlich durchgeführt.
- werden Anforderungen dann vollständig bestimmt, wenn sie zur Realisierung gelangen sollen.
- werden selten schwergewichtige Werkzeuge, sondern eher „leichte“ Tools eingesetzt.
- sind Änderungen weniger schwierig umzusetzen.



Bei den Agilen Methoden gibt es einige zentrale Elemente, über die die Entwicklung des zu erstellenden Systems gesteuert wird. Dies sind (unter anderem):

- Product Backlog: Eine Liste mit allen Anforderungen, die nach und nach zur Umsetzung gelangen
- User Stories, Epics, Themes: Elemente, um das zu entwickelnde System zu gliedern und so zu unterteilen, dass es besonders einfach umgesetzt werden kann
- (Scrum) Task Board: Wandtafel, die sich häufig in Agilen Umfeldern findet, auf der die einzelnen Prozessschritte über Spalten erfasst werden und auf die User Stories (als Karten) angeheftet und entsprechend ihres Bearbeitungszustands umgehängt werden

Diese Elemente werden hier im Nachfolgenden erläutert.



Die beim klassischen Requirements Engineering eingesetzten Werkzeuge und Methoden können zum Teil auch beim Agilen Requirements Engineering eingesetzt werden.

Hierzu zählen insbesondere:

- Das Brainstorming
- Die Verwendung von Glossaren
- Prüftechniken
- ...

Ein erheblicher Teil der Werkzeuge und Methoden des klassischen Requirements Engineerings wird jedoch nicht mehr benötigt.



Dem Requirements Engineer kommt beim Agilen Requirements Engineering – und hier speziell bei Scrum – die Rolle des Product Owners zu.

Der Product Owner hat folgende Aufgaben:

- Zusammenstellen aller Anforderungen im Product Backlog
- Koordinieren des Schätzens und des Priorisierens der umzusetzenden Anforderungen (Items) im Product Backlog
- Initiieren der Umsetzung (für den nächsten Sprint)
- Koordinieren des Einführungsprozesses
- Ansprechpartner für das Realisierungsteam und für den Kunden / das Management

Damit übernimmt der Product Owner mehr Aufgaben als der „klassische“ Requirements Engineer; einige Funktionen des klassischen Projektmanagers werden auf den Product Owner übertragen.



Alle Anforderungen werden beim Agilen Requirements Engineering in dem sogenannten **Product Backlog** festgehalten. Dieses enthält die Beschreibungen der (von den Anwendern gewünschten) Merkmale (**Features**) des zu erstellenden Produkts. Diese werden in einzelnen Punkten, den sogenannten **Product Backlog Items**, „tabellenartig“ festgehalten (siehe nächste Folie).

Die Backlog Items selbst sind bevorzugt **User Stories** (siehe hierzu übernächste Folie), es können aber auch Mindmaps o.Ä. verwendet werden. Typischerweise wird das Product Backlog über eine Wandtafel (Scrum Task Board), seltener über ein Software-Tool verwaltet.

Das Product Backlog ist das zentrale Element/Tool zum Verwalten der Anforderungen im Agilen Requirements Engineering.



Das Product Backlog kann als Tabelle aufgebaut sein und dann folgenden, minimalen Aufbau besitzen:

ID	Beschreibung	Priorität	Story Points	Notizen
1				
2				
3				
4				



Eintrag durch
das Team

Eintrag durch
Product Owner

ID	Fortlaufende ID, durch das System generiert
Beschreibung	Freitext - oder besser: Story-Text, wenn es User Stories sind
Priorität	Priorität; typische Skalen [A, B, C] oder [hoch, mittel, niedrig]
Story Points	Abschätzung für den Aufwand
Notizen	Freitext; kann auch leer bleiben

Ein Product Backlog Item entspricht genau einer Zeile in der Tabelle.



Über User Stories („Anwendererzählungen“) werden einzelne Inhalte des zu entwickelnden Systems in normaler Sprache beschrieben und häufig auf einfachen Karten notiert. Die Form lautet immer:

Als *<Rolle>*
möchte ich *<Ziel/Wunsch>*,
um *<einen Wert/Nutzen zu erlangen>*

Präsentation

Als Leser

*möchte ich die Präsentation mit einem
frei verfügbaren pdf-Reader lesen können
damit ich schnell darauf zugreifen kann*

Backautomat

Als Kunde

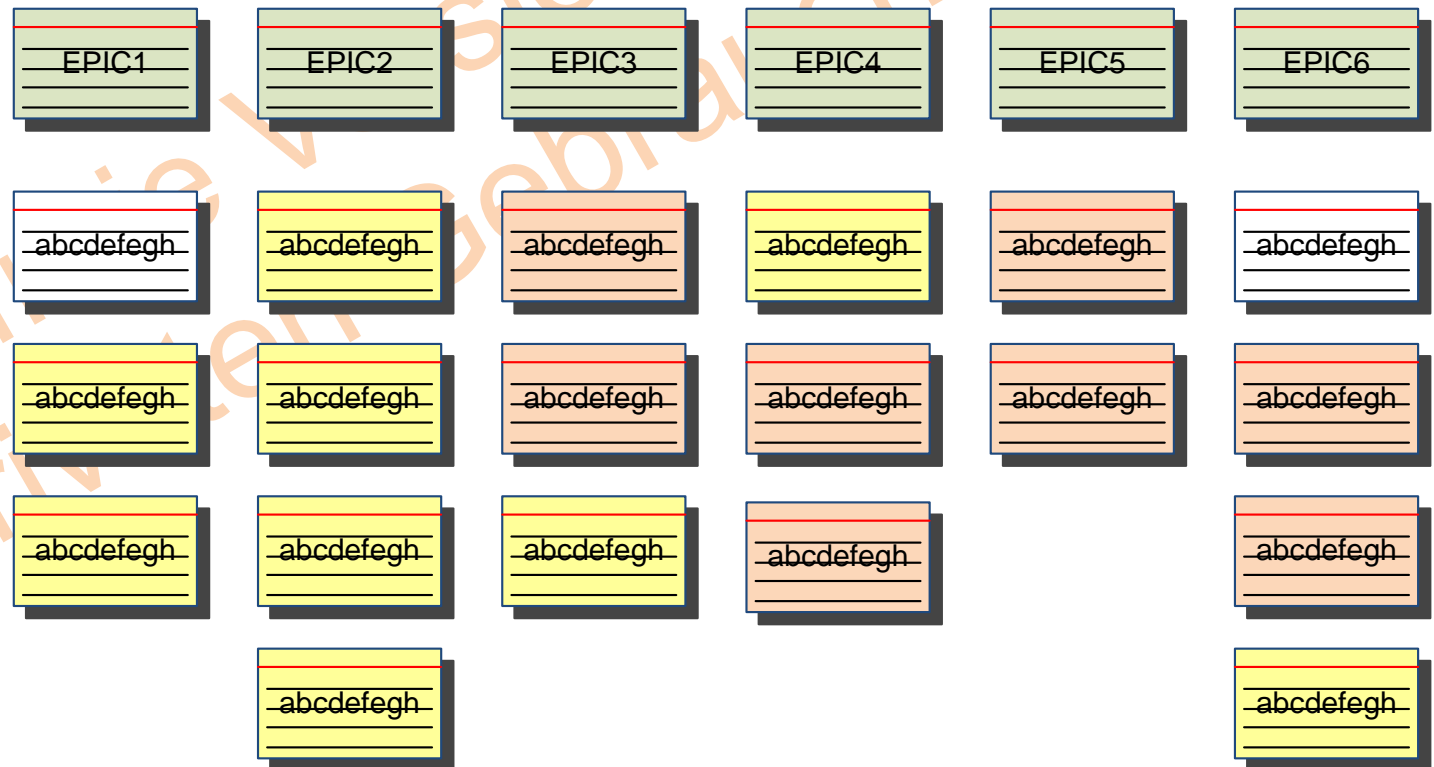
*möchte ich nicht länger als
3 Minuten auf mein fertiges Brot warten
da ich ansonsten abgepackte Ware kaufe*

(Gute) User Stories haben Eigenschaften, die den INVEST-Vorgaben folgen /Cohn10/. Dabei steht das Acronym INVEST für folgende Eigenschaften:

	Kurzform	Bedeutung
I	Independent	User Stories sollen unabhängig voneinander sein.
N	Negotiable	User Stories sollen verhandelbar sein.
V	Valuable	User Stories sollen einen Wert für den Kunden haben.
E	Estimatable	User Stories sollen schätzbar sein.
S	Small	User Stories sollen klein sein.
T	Testable	User Stories sollen testbar sein.



User Stories werden häufig in größeren Kontexten zusammengestellt. Über Story Maps werden User Stories in einer Art Landkarte zusammengestellt. Die Zusammenfassung mehrerer, verwandter User Stories wird als Epic bezeichnet, die grobe Beschreibung eines Anwendungsbereichs als Theme.





Beim Agilen Requirements Engineering (in agilen Projekten) ist zu beachten:

- Agile Methoden setzen stark auf „Kommunikation“ (und „Programmierung“) statt auf „Dokumente“: Daher sollte der Schwerpunkt bei ARE auch auf der Kommunikation liegen
- Merkregel: „Nicht mehr als 3 Dokumente!“ Die Reduktion auf wesentliche Dokumente ist zentraler Bestandteil des Agilen Requirements Engineering
- Aber: Ohne Kenntnis dessen, was gemacht werden muss, kann auch beim Agilen Vorgehen nichts zielgerichtet erstellt werden – RE bleibt auch in agilen Kontexten eine wesentliche Disziplin



Aus Sicht des klassischen Requirements Engineers:

- Überprüfen Sie, welche Elemente des Agilen Requirements Engineerings Sie auch beim klassischen RE benötigen könnten (und umgekehrt)
- Bei kleinen Projekten reicht oftmals ein Product Backlog aus, um Requirements Engineering zu betreiben. Sie brauchen dann nicht aufwendige(re) klassische RE-Tools einsetzen
- Vielfach schreckt die Sprache des agilen Umfeldes die Anwender ab, da die Begriffe nur dort existieren. Bereiten Sie daher Ihr Umfeld auf Agilität vor, wenn Sie Agile Methoden und Elemente einsetzen möchten



Teil III

- Notwendigkeit der Unterstützung
- Organisationsformen für Requirements Engineering
- Vorgehen bei der Einführung
- Tipps zur Einführung von Requirements Engineering

Kapitel 11

Seite
175–179



Wenn in Unternehmen oder Requirements Engineering nicht verankert ist, so sollte zur Einführung ein Prozess verwendet werden, der die Bedürfnisse der Organisation stützt: Ein angepasstes Requirements Engineering muss das Ziel sein.

Ebert /Ebert14/ unterscheidet dabei drei Modelle des Requirements Engineerings, die eingeführt werden könnten (siehe nächste Folie):

- Zentrales Requirements Engineering: Hier werden alle RE-Tätigkeiten (vorab) durch eine zentrale Stelle gebündelt
- Hybridmodelle: Mischung aus dem zentralen und eingebetteten Requirements Engineering
- Eingebettetes Requirements Engineering: In dieser Variante wird das RE im Projekt selbst durchgeführt

Welche Variante (oder Mischform) für ein Unternehmen „die beste“ ist, muss im Einzelfall bestimmt werden.



	Zentrales RE	Hybridmodelle	Eingebettetes RE
Methodenkompetenz, Standardisierung	++	+	0
Anwendungsnahe	0	+	++
Ausbildung	Intensive RE-Ausbildung und Beispiele aus verschiedenen Anwendungsbereichen	Grundlagen des RE und kritische Techniken für die Anwendungsbereiche	Hoher Aufwand aufgrund der verschiedenen Rollen; benötigt begleitendes Coaching
Verfügbarkeit, Auslastungsspitzen	Auslastungsspitzen können nicht kompensiert werden	Auslastungsspitzen können gut kompensiert werden	Auslastungsspitzen im Projekt führen zu Engpässen
Nutzen und Sparpotenzial	Gering (wegen Reibungsverlusten)	Hoch (wegen Balance, Methodik und Projektnähe)	Hoch (wegen Projektnähe wenn gute Planung und Tools)
Risiko	Formalismus und geringe Akzeptanz in Projekten („Elfenbeinturm“)	Ping-Pong-Effekte durch unzureichendes Verständnis in Projekten	Unzureichende Methodik, RE fällt im Zeitdruck herunter
Bewertung	Schlecht, da hoher Bedarf an spezifischem Projektverständnis und Risiko für Ineffizienz	Empfehlenswert wegen Konsistenz, sichtbare Rolle bringt Effizienz und Nachhaltigkeit	Empfehlenswert bei großen Produkten und Projekten, braucht gutes Coaching



Die Einführung von Requirements Engineering in Unternehmen sollte selbst als Projekt (Organisationsprojekt mit starken Change-Management-Anteilen) durchgeführt werden.

Hierzu sollte vorab eine intensive Analyse der Bedürfnisse und der Möglichkeiten erfolgen. Die Umsetzung selbst kann dann schrittweise durchgeführt werden.

Lizenzfreie Version für
den privaten Gebrauch! © 2025



- Ziehen Sie bei der Einführung von Requirements Engineering einen Berater zur Planung hinzu
- Überlegen Sie vorab, welche RE-Elemente Sie benötigen (jetzt, demnächst, langfristig)
- Planen Sie Zeit für Einführung von Requirements Engineering ein

Lizenzfreie Version für
den privaten Gebrauch! © 2021



Teil III

- Das Schätzen von Aufwänden
- Das Einbetten des Requirements Engineering im Projektmanagement
- Die sieben Hauptrisiken beim Requirements Engineering nach Ebert
- Bezeichnungen von Anforderungen im RE-Lebenszyklus
- Die Weiterverwendung von Anforderungen

Kapitel 12

Seite
180–186



Das Schätzen der Aufwände, die für die Realisierung der einzelnen Anforderungen notwendig sind, gehört auch zum Requirements Engineering. Typischerweise wird das Schätzen in verschiedenen Stufen durchgeführt (Grob- und Feinschätzung).

Folgende Fragen müssen bei der Aufwandsschätzung beantwortet werden:

- Was wird geschätzt?
- Wann wird geschätzt?
- Mit welcher Genauigkeit soll geschätzt werden?
- Wer schätzt?
- Wie wird geschätzt?

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.



Anforderungen sind für Projekte von zentraler Bedeutung, denn sie bestimmen, wie das zu erstellende Produkt oder die Dienstleistung aussehen wird.

Daher kann das Requirements Engineering als Disziplin betrachtet werden, über die Projekte gesteuert werden. Diese Sichtweise ist jedoch nicht sonderlich verbreitet.

In der Literatur zum Projektmanagement wird das Requirements Engineering (ebenso wie die Business Analysis) eher am Rande gestreift. Statt von Requirements wird dann häufig von Scope gesprochen, oder Anforderungen werden mit Zielen gleich gesetzt.



Folgende Risiken, die beim Requirements Engineering häufig auftreten und dabei besonders starke Probleme bereiten, benennt Ebert /Ebert12/:

1. Kunden sind unzureichend repräsentiert
2. Kritische Anforderungen werden übersehen
3. Nur funktionale Anforderungen werden berücksichtigt
4. Unkontrollierte Änderungen von Anforderungen
5. Beschreibung von Anforderungen als Entwurf
6. Anforderungen werden nicht geprüft
7. Perfektionierung von Anforderungen und Spezifikationen

Aber auch:

„Das neue System soll alle Funktionen des alten Systems enthalten und zusätzlich ...“

Die Bezeichnungen von Anforderungen können sich im Lebenszyklus eines Produkts ändern – je später eine Anforderung hinzugenommen wird, desto aufwendiger wird die Umsetzung.

Spezielle Bezeichnung	Phase	Aufwand der Umsetzung / Bereinigung
Anforderungen	RE-Startphase	Gering
Änderungen	Implementierungsphase	Hoch
Fehler	Start- und Abnahmephase	Sehr hoch
Produkterweiterungen	Betrieb	Mittel bis hoch

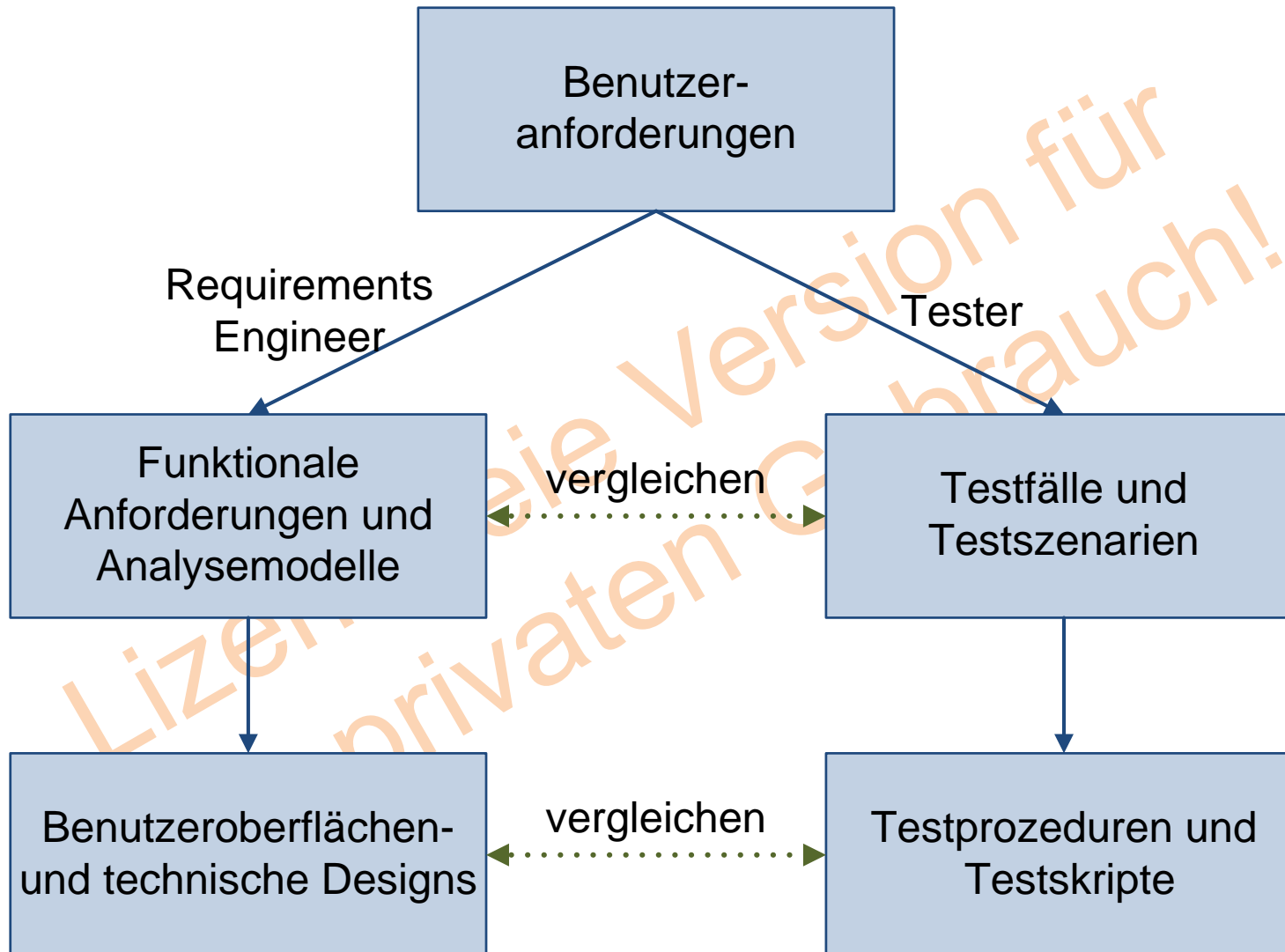


Die Anforderungen werden weiterverwendet, insbesondere ...

- beim Testen (falls sie umgesetzt wurden).
- im Service Management (Fehler im laufenden Betrieb).

Aus den Anforderungen sollten auch abgestimmte End-Anwender-Dokumente „generiert“ werden können, wie:

- Bedienungshandbücher
- Service-Handbücher
- Bedienungsanleitungen



nach /Wiegors05/



Teil III

- Fragen zum Teil III

Kapitel 13

Seite
187–188



- Welche Aufgaben können besonders gut mit einem RE-Tool erledigt werden?
- Warum ist die Nachverfolgbarkeit von Anforderungen wichtig? Wie wird diese in den Tools gewährleistet?
- Was zeichnet Agiles Requirements Engineering aus?
- Was ist eine User Story?

Lizenzfreie Version für
den privaten Gebrauch! © 2017



Teil IV

Teil IV

- Anhang A Literatur, Weblinks, Sprüche und Glossar
- Anhang B Normen, Standards und Zertifikate
- Anhang C Reifegradmodelle und techn. Dokumentation
- Anhang D Weitere Präsentationen, Kontakt zum Autor

Seite
189–248



Teil IV

- Literatur
- Anmerkungen zur Literatur: Empfehlungen
- Weblinks
- Sprüche
- Glossar – Top-Ten-Begriffe zum Requirements Engineering

Anhang A

Seite
190–220



- /ABPMP13/ ABPMP, EABPMP: BPM CBOK Version 3.0: Guide to the Business Process Management Common Body Of Knowledge, CreateSpace, Charleston, South Carolina 2013, ISBN 978-1-49051659-2
- /ABPMP14-d/ ABPMP, EABPM: BPM CBOK – Business Process Management BPM Common Body of Knowledge, Version 3.0, Leitfaden für das Prozessmanagement, Dr. Götz Schmidt, Wettenberg 2. Auflage 2014, ISBN 978-3-921313-91-6
- /Alexander02/ Ian Alexander, Richard Stevens: Writing Better Requirements, Addison-Wesley Professional, Boston, Massachusetts 2002, ISBN 978-0321-13163-8
- /Alexander09/ Ian Alexander, Ljerka Beus-Dukic: Discovering Requirements: How to Specify Products and Services, John Wiley & Sons, Hoboken, New Jersey 2009, ISBN 978-0-470-71240-5

Diese Literaturliste ist auf Basiswerke reduziert. Eine umfangreichere Liste (vom gleichen Autor) ist unter https://www.peterjohann-consulting.de/_pdf/peco-re-literatur.pdf frei herunterladbar.



- /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
- /Balzert11/ Helmut Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Spektrum Akademischer Verlag, 3. Auflage Heidelberg 2011, ISBN 978-3-8274-1706-6
- /BAPG15/ Project Management Institute: Business Analysis For Practitioners: A Practice Guide, Project Management Institute, Philadelphia, Pennsylvania 2015, ISBN 978-1-62825-069-5



- /BBG09/ IIBA: A Guide to the Business Analysis Body of Knowledge (BABOK Guide), International Institute of Business Analysis, Marietta, Georgia 2nd Edition 2009, ISBN 978-0-98112921-1
- /BBG12-d/ IIBA: Leitfaden zum Business Analysis Body of Knowledge: BABOK 2.0, Dr. Götz Schmidt, Wettenberg 2012, ISBN 978-3-921313-81-7
- /BBG15/ IIBA: A Guide to the Business Analysis Body of Knowledge (BABOK Guide), International Institute of Business Analysis, Marietta, Georgia 3rd Edition 2015, ISBN 978-1-927584-02-6
- /BBG17-d/ IIBA: BABOK v3: Leitfaden zur Business-Analyse BABOK Guide 3.0, Dr. Götz Schmidt, Wettenberg 2017, ISBN 978-3-945997-03-1
- /Beatty12/ Joy Beatty, Anthony Chen: Visual Models for Software Requirements, Microsoft Press, Redmond, Washington 2012, ISBN 978-0-7356-6772-3
- /Bergsmann14/ Johannes Bergsmann: Requirements Engineering für die agile Softwareentwicklung: Methoden, Techniken und Strategien zur erfolgreichen Umsetzung, dpunkt, Heidelberg 2014, ISBN 978-3-86490-149-2
- /Blais11/ Stephen Blais: Business Analysis: Best Practices for Success, John Wiley & Sons, Hoboken, New Jersey 2011, ISBN 978-1-118-07600-2



- /Cadle14a/ James Cadle, Malcolm Eva, Keith Hindle, Debra Paul: Business Analysis, BCS, The Chartered Institute for IT, London 3rd Edition 2014, ISBN 978-1-78017-277-4
- /Cadle14b/ James Cadle, Debra Paul, Paul Turner: Business Analysis Techniques: 99 Essential Tools for Success, BCS, The Chartered Institute for IT, London 2nd Edition 2014, ISBN 978-1-78017-273-6
- /Carkenord09/ Barbara A. Carkenord: Seven Steps to Mastering Business Analysis, J. Ross Publishing, Fort Lauderdale, Florida 4th Edition 2009, ISBN 978-1-60427-007-5
- /Carkenord16/ Barbara A. Carkenord: PMI-PBA Exam Prep, Premier Edition, RMC Publications, Burnsville, Minnesota 2nd Edition 2016, ISBN 978-1-943704-01-9
- /Cockburn00/ Alistair Cockburn: Writing Effective Use Cases, Addison-Wesley Longman, Amsterdam 2000, ISBN 978-0-201-70225-5
- /Cohn10/ Mike Cohn: User Stories: Für die agile Software-Entwicklung mit Scrum, XP u.a., mitp, Bonn 2010, ISBN 978-3-8266-5898-3
- /Chrissis11a/ Mary Chrissis, Mike Konrad, Sandy Shrum: CMMI for Development: Guidelines for Process Integration and Product Improvement, Addison-Wesley Professional, Old Tappan, New Jersey 3rd Edition 2011, ISBN 978-0-321-71150-2
- /Chrissis11b/ Mary Chrissis, Mike Konrad, Sandy Shrum: CMMI 1.3 für die Entwicklung: Richtlinien für Prozessintegration und Produktverbesserung, Addison-Wesley, München 2011, ISBN 978-3-8273-3065-9
- /Davis05/ Alan M. Davis: Just Enough Requirements Management: Where Software Development Meets Marketing, Dorset House Publishing, New York 2005, ISBN 978-0-932633-64-4



- /Ebert12/ Christof Ebert: Systematisches Requirements Engineering. Anforderungen ermitteln, spezifizieren, analysieren und verwalten, dpunkt, Heidelberg 4. Auflage 2012, ISBN 978-3-89864-812-7
- /Ebert14/ Christof Ebert: Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren, analysieren und verwalten, dpunkt, Heidelberg 5. Auflage 2014, ISBN 978-3-86490-139-3
- /Gause89/ Donald C. Gause, Gerald M. Weinberg: Exploring Requirements. Quality before Design, Dorset House Publishing, New York 1989, ISBN 978-0-932633-13-2
- /Gause93/ Donald C. Gause, Gerald M. Weinberg: Software Requirements. Anforderungen erkennen, verstehen und erfüllen, Hanser, München 1993, ISBN 978-3-446-17113-8
- /Gerstbach15/ Ingrid Gerstbach, Peter Gerstbach: Basiswissen Business-Analyse: Probleme lösen, Chancen nutzen, Redline, München 2015, ISBN 978-3-86881-574-0
- /Gilb05/ Tom Gilb: Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, Butterworth-Heinemann, Burlington, Massachusetts 2005, ISBN 978-0-7506-6507-0
- /Got02/ Ellen Gottesdiener: Requirements by Collaboration, Addison-Wesley Professional, Boston, Massachusetts 2002, ISBN 978-0201-78606-4
- /Got05/ Ellen Gottesdiener: The Software Requirements Memory Jogger: A Pocket Guide to Help Software and Business Teams Develop and Manage Requirements, Goal/QPC, Salem, New Hampshire 2005, ISBN 978-157681-060-6
- /Grady06/ Jeffrey O. Grady: System Requirements Analysis, Academic Press Inc., Burlington, Massachusetts 2006, ISBN 978-0-12-088514-5
- /Grady13/ Jeffrey O. Grady: System Requirements Analysis, Elsevier Science, London 2nd Edition 2013, ISBN 978-0-12-417107-7
- /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



- /Haber15/ Reinhard Haberfellner, Oliver L. de Weck, Ernst Fricke, Siegfried Vössner: Systems Engineering: Grundlagen und Anwendung, Orell Füssli, Zürich 13. Auflage 2015, ISBN 978-3-280-04068-3
- /Hammer13/ Ulrike Hammerschall, Gerd Beneken: Software Requirements, Pearson Studium, München 2013, ISBN 978-3-86894-151-7
- /Hanschke16/ Inge Hanschke, Gunnar Giesinger, Daniel Goetze: Business Analyse – einfach und effektiv: Geschäftsanforderungen verstehen und in IT-Lösungen umsetzen, Hanser, München 2. Auflage 2016, ISBN 978-3-446-44345-7
- /Herrmann12/ Andrea Herrmann, Eric Knauss, Rüdiger Weißbach: Requirements Engineering und Projektmanagement, Springer, Berlin 2012, ISBN 978-3-642-29431-0
- /Hood05/ Colin Hood, Rupert Wiebel: Optimieren von Requirements Management & Engineering mit dem HOOD Capability Model, Springer, Berlin 2005, ISBN 978-3-540-21178-5
- /Hood07/ Colin Hood, Chris Rupp, Susanne Mühlbauer, Gerhard Versteegen, Rupert Wiebel: ix-Studie Anforderungsmanagement, Heise, Hannover 2. Auflage 2007, ohne ISBN
- /Hossen07/ Rosemary Hossenlopp, Kathleen B. Hass: Unearthing Business Requirements: Elicitation Tools and Techniques, Management Concepts, Washington D.C. 2007, ISBN 978-1-56726-210-0
- /Hruschka14/ Peter Hruschka: Business Analysis und Requirements Engineering: Prozesse und Produkte nachhaltig verbessern, Hanser, München 2014, ISBN 978-3-446-43807-1
- /Hull10/ Elisabeth Hull, Jeremy Dick, Kenneth Jackson: Requirements Engineering, Springer, Berlin 3rd Edition 2010, ISBN 978-1-84996-404-3



/IREB15/ siehe /Pohl15a/

/Jonasson12/ Hans Jonasson: Determining Project Requirements. Mastering the BABOK and the CBAP Exam, Auerbach Publications, New York 2nd Edition 2012, ISBN 978-1-4398-9651-8

/Jonasson16/ Hans Jonasson: CBAP Certification and BABOK Study Guide, Taylor & Francis, London 2016, ISBN 978-1-4987-6725-5

/Kotonya98/ Gerald Kotonya, Ian Sommerville: Requirements Engineering, John Wiley & Sons, Hoboken, New Jersey 1998, ISBN 978-0-471-97208-2

/Kuper13/ Kupe Kupersmith, Paul Mulvey, Kate McGoey: Business Analysis For Dummies, John Wiley & Sons, Hoboken, New Jersey 2013, ISBN 978-1-118-51058-2

/Larson15/ Elizabeth Larson, Richard Larson: PMI-PBA Certification Study Guide, Watermark Learning, Minneapolis, Minnesota 2015, ISBN 978-0-578-15547-0

/Larson16/ Elizabeth Larson, Richard Larson: CBAP Certification Study Guide v3.0, Watermark Learning, Minneapolis, Minnesota 2016, ISBN 978-0-692-69145-8

/Leffing03/ Dean Leffingwell, Don Widrig: Managing Software Requirements. A Use Case Approach Addison-Wesley Professional, Boston, Massachusetts 2nd Edition 2003, ISBN 978-0-321-12247-6

/Leffing07/ Dean Leffingwell: Scaling Software Agility: Best Practices for Large Enterprises, Addison-Wesley Professional, Boston, Massachusetts 2007, ISBN 978-0-321-45819-3

/Leffing10/ Dean Leffingwell: Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise, Addison-Wesley Professional, Boston, Massachusetts 2010, ISBN 978-0-321-63584-8



- /Moore06/ James W. Moore: The Road Map to Software Engineering: A Standards-Based Guide, John Wiley & Sons, Hoboken, New Jersey 2006, ISBN 978-0-471-68362-9
- /Niebisch13/ Thomas Niebisch: Anforderungsmanagement in sieben Tagen: Der Weg vom Wunsch zur Konzeption, Springer, Berlin 2013, ISBN 978-3-642-34856-3
- /Nielsen16/ Klaus Nielsen: Achieve Business Analysis Certification: The Complete Guide to PMI-PBA, CBAP and CPRE Exam Success, J. Ross Publishing, Fort Lauderdale, Florida 2016, ISBN 978-1-60427-111-9
- /Partsch10/ Helmuth Partsch: Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme, Springer, Berlin 2. Auflage 2010, ISBN 978-3-642-05357-3
- /PBG12/ Project Management Institute: A Guide to the Project Management Body of Knowledge, Project Management Institute, Philadelphia, Pennsylvania Fifth Edition 2012, ISBN 978-1-935589-67-9
- /PBG12-d/ Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK Guide). Fünfte Ausgabe, Project Management Institute, Philadelphia, Pennsylvania 2012, ISBN 978-1-62825-003-9
- /PMG-BA17/ Project Management Institute: The PMI Guide to Business Analysis, Project Management Institute, Philadelphia, Pennsylvania 2017, ISBN 978-1-62825-198-2



- /Pohl08/ Klaus Pohl: Requirements Engineering. Grundlagen, Prinzipien, Techniken, dpunkt, Heidelberg 2. Auflage 2008, ISBN 978-3-89864-550-8
- /Pohl10/ Klaus Pohl: Requirements Engineering. Fundamentals, Principles, and Techniques, Springer, Berlin 2010, ISBN 978-3-642-12577-5
- /Pohl15a/ auch /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
- /Pohl15b/ Klaus Pohl, Chris Rupp: Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant, Rocky Nook, Santa Barbara, California 2nd Edition 2015, ISBN 978-1-937538-77-4



- /REPG16/ Project Management Institute: Requirements Management: A Practice Guide, Project Management Institute, Philadelphia, Pennsylvania 2016, ISBN 978-1-62825-089-3
- /Robertson04/ Suzanne Robertson, James Robertson: Requirements-Led Project Management. Discovering David's Slingshot, Addison-Wesley Professional, Boston, Massachusetts 2004, ISBN 978-0-321-18062-9
- /Robertson06/ Suzanne Robertson, James Robertson: Mastering the Requirements Process, Addison-Wesley Professional, Boston, Massachusetts 2nd Edition 2006, ISBN 978-0-321-41949-1
- /Robertson12/ Suzanne Robertson, James Robertson: Mastering the Requirements Process, Addison-Wesley Professional, Boston, Massachusetts 3rd Edition 2012, ISBN 978-0-321-81574-3



- /Rupp12/ Chris Rupp, Stefan Queins: UML 2 glasklar: Praxiswissen für die UML-Modellierung, Hanser, München 4. Auflage 2012, ISBN 978-3-446-43057-0
- /Rupp13/ Chris Rupp: Systemanalyse kompakt, Springer Vieweg, Berlin 3. Auflage 2013, ISBN 978-3-642-35445-8
- /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
- /Rupp16/ Chris Rupp (und die Sophisten): Requirements-Engineering: Die kleine RE-Fibel, Sophist, Nürnberg, 3. Auflage 2016, keine ISBN
- /Rüping13/ Andreas Rüping: Dokumentation in agilen Projekten: Lösungsmuster für ein bedarfsgerechtes Vorgehen, dpunkt, Heidelberg 2013, ISBN 978-3-86490-040-2



- /Schienmann01/ Bruno Schienmann: Anforderungsmanagement: Prozesse – Techniken – Werkzeuge, Addison-Wesley, München 2001, ISBN 978-3-8273-1787-2
- /Schmidt14/ Götz Schmidt: Organisation und Business Analysis – Methoden und Techniken, Dr. Götz Schmidt, Wettenberg 15. Auflage 2014, ISBN 978-3-921313-93-0
- /Schmied08/ Jürgen Schmied, Paul-Roux Wentzel, Michael Gerdorn: Mit CMMI Prozesse verbessern! Umsetzungsstrategien am Beispiel Requirements Engineering, dpunkt, Heidelberg 2008, ISBN 978-3-89864-538-6
- /Schwinn11/ Hans Schwinn: Requirements Engineering: Modellierung von Anwendungssystemen, Oldenbourg Wissenschaftsverlag, München 2011, ISBN 978-3-486-58893-4
- /Sommerville97/ Ian Sommerville, Pete Sawyer: Requirements Engineering. A Good Practice Guide, John Wiley & Sons, Hoboken, New Jersey 1997, ISBN 978-0-471-97444-4



- /Weilkins14/ Tim Weilkins: Systems Engineering mit SysML/UML: Anforderungen, Analyse, Architektur, dpunkt, Heidelberg 3. Auflage 2014, ISBN 978-3-86490-091-4
- /Wiegers03/ Karl E. Wiegers: Software Requirements, Microsoft Press, Redmond, Washington 2nd Edition 2003, ISBN 978-0-7356-1879-4
- /Wiegers05/ Karl E. Wiegers: Software-Requirements, Microsoft Press, München Deutsche Ausgabe der 2nd Edition 2005, ISBN 978-3-860-63594-0
- /Wiegers06/ Karl E. Wiegers: More About Software Requirements: Thorny Issues and Practical Advice, Microsoft Press, Redmond, Washington 2006, ISBN 978-0-7356-2267-8
- /Wiegers13/ Karl E. Wiegers, Joy Beatty: Software Requirements, Microsoft Press, Redmond, Washington 3rd Edition 2013, ISBN 978-0-7356-7966-5
- /Withall07/ Stephen J. Withall: Software Requirement Patterns. Best Practices, Microsoft Press, Redmond, Washington 2007, ISBN 978-0-7356-2398-9



- /Young01/ Ralph R. Young: Effective Requirements Practices, Addison-Wesley Professional, Boston, Massachusetts 2nd Edition 2001, ISBN 978-0-201-70912-4
- /Young03/ Ralph R. Young: The Requirements Engineering Handbook, Artech House Publishers, Norwood, Massachusetts 2003, ISBN 978-1-5805-3266-2
- /Young06/ Ralph R. Young: Project Requirements: A Guide to Best Practices, Management Concepts, Washington D.C. 2006, ISBN 978-1-56726-169-8

Lizenzfreie Verwendung
den privaten Gebrauch



Folgende Bücher zum „reinen“ Requirements Engineering werden besonders häufig zitiert und können daher als (minimaler) Standard angesehen werden:

- /Ebert14/: Umfassendes RE-Buch (deutsch), inzwischen in der 5. Auflage erschienen
- /Got05/: Englischsprachiger Klassiker
- /Pohl15a/ (= /IREB15/): Einstiegsbuch, welches aufzeigt, welche Inhalte vom IREB (International Requirements Engineering Board) als wesentlich betrachtet werden; 4. Auflage
- /Rupp14/: RE-Buch (deutsch), welches die praktischen Aspekte in den Vordergrund stellt; 6. Auflage
- /Robertson12/: Englischsprachiger Klassiker, ursprünglich aus dem Jahr 1999
- /Wiegers13/: Englischsprachiger Klassiker, inzwischen in der 3. Auflage; 1. Auflage aus dem Jahr 1999



Auf den folgenden Seiten sind Weblinks aufgeführt, die zur Beschreibung oder zur Einarbeitung in das Requirements Engineering hilfreich sein können. Diese Weblinks wurden zum Teil in dieser Präsentation verwendet. Eine Bewertung der Websites und deren Inhalte wird hier nicht vorgenommen, kann aber vom Autor abgefragt werden.

Legende für die nachfolgenden Folien – so werden die Weblinks klassifiziert:

// Verweis auf Website generell

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

/#/ Verweis auf einzelnes Thema auf einer Website

/#V/ Verweis auf ein Video (auf einer Website) mit Minutenangabe und Sprache



- /AK-Req/ Arbeitskreis Requirements der GI-Regionalgruppe München: <https://www.gi-muc-ak-req.de>; eingesehen am 02.03.2017
- /Easy-RE/ Ralf Baumann: Blog zum Requirements Engineering (deutsch): <https://easyrequirement.blogspot.de>; eingesehen am 02.03.2017
- /GfSE/ Gesellschaft für Systems Engineering e.V., German Chapter of INCOSE
/INCOSE/: <https://www.gfse.de/>; eingesehen am 02.03.2017
- /GI-FG-RE/ Fachgruppe Requirements der Gesellschaft für Informatik (2.1.6):
<https://fg-re.gi.de/>; eingesehen am 02.03.2017
- /Glinz-RE-I-Vorles-16/ Martin Glinz: Vorlesung „Requirements Engineering I“ aus dem Jahr 2016 (englisch): <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs16/re-i.html>; eingesehen am 02.03.2017
- /Glinz-RE-II-Vorles-16/ Martin Glinz: Vorlesung „Requirements Engineering II“ aus dem Jahr 2016 (englisch): <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs16/re-ii.html>; eingesehen am 02.03.2017
- /INCOSE/ International Council of Systems Engineering (INCOSE):
<https://www.incose.org/>; eingesehen am 02.03.2017



- /Meetup-ARE/ Meetup-Gruppe Agiles Requirements Engineering München:
<https://www.meetup.com/Munchen-Agile-Requirements-Engineering-Meetup/>;
eingesehen am 02.03.2017
- /Modern-Analyst/ Modern Analyst: <https://www.modernanalyst.com>; eingesehen am
02.03.2017
- /OBJSpek-16/ OBJEKTSpektrum 02/2016, Titelthema „Modernes Requirements“:
<https://www.sigs-datacom.de/fachzeitschriften/objektspektrum/archiv/artikelansicht/ausgabebetitel/modernes-requirements-management.html>; eingesehen am 02.03.2017
- /OBJSpek-RE-15/ OBJEKTSpektrum, Onlinethemenspecial RE/2015, Titelthema:
Requirements Engineering <https://www.sigs-datacom.de/fachzeitschriften/objektspektrum/archiv/artikelansicht/ausgabebetitel/requirements-engineering-3.html>; eingesehen am 02.03.2017
- /OBJSpek-RE-16/ OBJEKTSpektrum, Onlinethemenspecial RE/2016, Titelthema:
Requirements Engineering: <https://www.sigs-datacom.de/fachzeitschriften/objektspektrum/archiv/artikelansicht/ausgabebetitel/requirements-engineering-2.html>; eingesehen am 02.03.2017



- /RE-Bib/ Requirements Bibliography – umfangreiche Literaturliste von Alan M. Davis:
<http://www.reqbib.com/adavis/>; eingesehen am 02.03.2017
- /#RE-Kompass-13/ Ergebnisbericht RE-Kompass 2013 des Fraunhofer IESE (deutsch, 38 Seiten):
https://www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche_studien/Ergebnisbericht_RE-Kompass_2013.pdf; eingesehen am 02.03.2017
- /RE-Magazine/ Requirements Engineering Magazine (englisch): <https://re-magazine.ireb.org/>; eingesehen am 02.03.2017
- /#RE-Software-Tools/ Liste mit Software-Tools für das Requirements Engineering von Andreas Birk und Gerald Heller (englisch):
<https://makingofsoftware.com/resources/list-of-rm-tools>; eingesehen am 02.03.2017



- /RE-Wissen/ Portal für Anforderungsmanagement in kleinen und mittleren Unternehmen: <http://www.re-wissen.de>; eingesehen am 02.03.2017
- /OMG-ReqIF/ ReqIF – Austauschformat, Version 1.2 vom 01.07.2016, 104 Seiten, englisch: <https://www.omg.org/spec/ReqIF/>; eingesehen am 02.03.2017
- /SysGuild/ The Atlantic System Guild: Zusammenschluss einiger namhafter RE-Autoren (deMarco, Hruschka, Robertson, Lister, ...): <http://www.systemsguild.com>; eingesehen am 02.03.2017
- /Sophist/ SOPHIST GmbH, Nürnberg: Autoren der Bücher /Rupp13, Rupp14, Rupp15 Rupp16/: <https://www.sophist.de>; eingesehen am 02.03.2017



- /Volere/ Website mit dem (inzwischen nicht mehr kostenlosen) Volere-Template (aus /Robertson12/), aktuelle Version 18 aus 2016 (Preis: \$55) – letzte kostenfreie Version: 13 von 2008: <https://www.volere.co.uk>; eingesehen am 02.03.2017
- /Wiki-d/ Deutsche Wikipedia: <https://de.wikipedia.org>; eingesehen am 02.03.2017
- /Wiki-e/ Englische Wikipedia: <https://en.wikipedia.org>; eingesehen am 02.03.2017
- /XING-RE/ XING-Gruppe zum Requirements Engineering:
<https://www.xing.com/communities/groups/requirements-engineering-1068749>;
eingesehen am 02.03.2017

Lizenzfreie
den privaten Gebrauch



- /*Gerstbach-Podcasts/ Podcasts zum Thema Business Analysis, von den Autoren von /Gerstbach15/: <http://gerstbach.at/podcast/>; eingesehen am 02.03.2017
- /*Got02/ Website zum Buch von Ellen Gottesdiener /Got02, Got05/: <https://www.ebgconsulting.com/>; eingesehen am 02.03.2017
- /*Pohl08/ Website zum Buch von Klaus Pohl /Pohl08/ und /Pohl15a, Pohl15b/: <https://re-buch.de/buecher/>; eingesehen am 02.03.2017
- /*Robertson12/ Website zum Buch von Suzanne und James Robertson /Robertson12/: <https://www.volere.co.uk>; eingesehen am 02.03.2017
- /*Wiegers13/ Website zum Buch von Karl E. Wiegers /Wiegers13/: <https://www.processimpact.com/>; eingesehen am 02.03.2017



/#Capri-13/ Stichpunkte zum RE, Mitschrift eines CPRE-Kurses (deutsch)

<http://www.capri-soft.de/blog/?p=836>; eingesehen am 02.03.2017

/#pmag-Konflikte-im-RE-10/ projektmagazin 08/2010 (21.04.2010): „Wenn jeder etwas anderes will – Konflikte im Anforderungsmanagement lösen“, Autorin: Chris Rupp:

https://www.projektmagazin.de/artikel/konflikte-im-anforderungsmanagement-loesen_7266; eingesehen am 02.03.2017

Lizenzfreie Verwendung
den privaten Gebrauch



- /#Stöckel-14/ Frank Stöckel und Jens Donig (Hood GmbH): „Die 7 Irrtümer bei der Einführung von Requirements Engineering“, 24.03.2014:
https://www.slideshare.net/HOOD_Requirements_Experts/7-irrtmer-bei-der-einfhrung-von-requirements-engineering-andere-titelfolie; eingesehen am 02.03.2017
- /#Wiki-AnfMgmt/ Anforderungsmanagement / Requirements Management in der deutschen Wikipedia: <https://de.wikipedia.org/wiki/Anforderungsmanagement>;
eingesehen am 02.03.2017
- /#Wiki-BA/ Business Analyse in der deutschen Wikipedia:
<https://de.wikipedia.org/wiki/Business-Analyse>; eingesehen am 02.03.2017
- /#Wiki-K-AnfMgmt/ Kategorie Anforderungsmanagement in der deutschen Wikipedia:
<https://de.wikipedia.org/wiki/Kategorie:Anforderungsmanagement>; eingesehen am 02.03.2017
- /#Wiki-Kano-Modell/ Kano-Modell in der deutschen Wikipedia:
<https://de.wikipedia.org/wiki/Kano-Modell>; eingesehen am 02.03.2017
- /#Wiki-RE-Ana-e/ Requirements Analysis in der englischen Wikipedia:
https://en.wikipedia.org/wiki/Requirements_analysis; eingesehen am 02.03.2017
- /#Wiki-UML/ UML (Unified Modeling Language) in der deutschen Wikipedia:
<https://de.wikipedia.org/wiki/UML>; eingesehen am 02.03.2017



- /Berliner-RE/ Berliner Requirements Engineering Symposium: <https://www.berliner-re-symposium.de>; eingesehen am 02.03.2017
- /RE-Conf/ Requirements Conference, München: <https://www.hood-group.com/reconf/>; eingesehen am 02.03.2017
- /RE-Int-Conf/ Requirements Conference, jährliche, an wechselnden Orten weltweite Wissenschaftskonferenz: <http://requirements-engineering.org/>; eingesehen am 02.03.2017
- /RE-Int-Work-Conf/ International Working Conference on Requirements Engineering <https://refsq.org>; eingesehen am 02.03.2017

Lizenzfreie Version für
den privaten Gebrauch! © 2025



- /IREB/ Die Website des IREB (International Requirements Engineering Board e.V., deutsch) generell: <https://www.ireb.org/de/>; eingesehen am 02.03.2017
- /IREB-Glossar/ Das Glossar (mehrsprachig) mit Erklärungen auf Englisch und Begriffe in den jeweilige Sprachen; 130 Seiten, davon 16 Glossary (englisch) und 10 Seiten deutsch <-> englische Begriffe: <https://www.ireb.org/de/downloads/#cpre-glossary>; eingesehen am 02.03.2017
- /IREB-Lehrpläne/ Die Lehrpläne zum CPRE Foundation Level (mehrere Sprachen) mit Prüfungsordnung und Übungsprüfungen: <https://www.ireb.org/de/exams/foundation/>; eingesehen am 02.03.2017
- /IREB-17/ Lehrplan zum CPRE Foundation Level, deutsche Version 2.2.1 vom 24.07.2017: https://www.ireb.org/content/downloads/2-syllabus-foundation-level/ireb_cpre_syllabus_fl_de_v221.pdf; eingesehen am 24.07.2017
- /IREB-Literatur/ Einige Basisbücher aus Sicht des IREB: <https://www.ireb.org/de/service/literature/>; eingesehen am 02.03.2017
- /IREB-Zertzahlen/ Die Zertifizierungszahlen zum CPRE (weltweit): <https://www.ireb.org/de/service/statistics/>; eingesehen am 02.03.2017



/IIBA/ International Institute of Business Analysis (IIBA): <https://www.iiba.org>;
eingesehen am 02.03.2017

/IIBA-GC/ International Institute Business Analysis (IIBA), German Chapter:
<https://germany.iiba.org/>; eingesehen am 02.03.2017

/PMI/ Project Management Institute (PMI): <https://www.pmi.org>; eingesehen am
02.03.2017

/PMI-SGC/ Project Management Institute (PMI), Southern Germany Chapter:
<https://pmi-sgc.de/>; eingesehen am 02.03.2017

/PMI-BA/ Informationen vom PMI zur Business Analysis:
<https://www.pmi.org/learning/featured-topics/business-analysis>; eingesehen am
02.03.2017

/PMI-PBA/ Informationen zum PMI-PBA (Professional in Business Analysis):
<https://www.pmi.org/learning/featured-topics/requirements>; eingesehen am
02.03.2017

/#V-PMI-PBA/ Video auf Youtube zu „PMI-PBA: From ‘What Is It?’ To ‘I Did It!’“ von
Richard Larson, 07.11.2016, 59:35 Minuten, englisch,
https://www.youtube.com/watch?v=m77ni_zkNqYe; eingesehen am 02.03.2017



- „Ich habe keine Zeit, die Axt zu schärfen – ich muss Bäume fällen.“ /Hood05/
„Wenn ich acht Stunden hätte um einen Baum zu fällen, würde ich davon sechs verwenden, die Axt zu schärfen.“ (Abraham Lincoln)
„Ohne Änderungen ist ein Projekt kein Projekt.“ (unbekannt)
„Das beste RE-Tool ist das Whiteboard.“ /Robertson06/
„Der Kunde bekommt nicht das, was er braucht, sondern das, was er will!“ /Ebert14/
„Gut ist, wenn der Kunde wiederkommt und nicht das Produkt.“ (unbekannt)
„Sehr gut ist, wenn der Kunde wiederkommt und seinen Freund mitbringt.“ (unbekannt)
„Wenn jemand nur einen Hammer hat, sieht er überall nur Nägel.“ (Paul Watzlawick)
„Wenn ich meine Kunden nach ihren Wünschen gefragt hätte, dann hätten sie mir gesagt, dass sie gern stärkere Pferde für ihre Kutschen hätten.“ (Henry Ford)
„Ein Projekt kennt entweder nur Gewinner oder nur Verlierer.“ (unbekannt)
„The perfect is the enemy of the good.“ (unbekannt)
„Die Niederlage zu verstehen ist der erste Schritt zum Sieg.“ (Mao)
„It isn't that they can't see the solution. It is that they can't see the problem.“ (Gilbert K. Chesterton)



Begriff	Beschreibung	Quelle
Anforderung	Beschaffenheit, Fähigkeit oder Leistung, die ein Produkt, Prozess oder die am Prozess beteiligte Person erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere formell vorgegebene Dokumente zu erfüllen	DIN 69901-5: 2009-01
Entwickler (Designer)	Derjenige, der für die Umsetzung der Spezifikation sorgt	selbst
Kunde (Benutzer, Anwender)	Derjenige, der das Produkt oder die Dienstleistung nutzen soll – er macht die Vorgaben	selbst
Produkt	Arbeitsergebnis, das aus Prozessen resultiert	selbst
Prozess	Abfolge zusammengehöriger Tätigkeiten, die der Erreichung eines Ziels dient	selbst



Begriff	Beschreibung	Quelle
Requirements Engineering	RE ist der systematische Ansatz, Anforderungen des Systems zu erheben, zu organisieren und zu dokumentieren und einen Prozess, der eine Übereinkunft zwischen dem Kunden und dem Projektteam liefert, zu etablieren	offen
Requirements Engineer	Derjenige, der das zu erstellende Produkt, System oder die Dienstleistung beschreibt. (Andere Bezeichnungen: Requirements-Ingenieur, Business Analyst, Anforderungsingenieur, Systemanalytiker, Systems Analyst, Requirements Analyst)	selbst
Stakeholder	Interesseninhaber oder -vertreter. Dies können sein: Der Fachanwender oder die Fachabteilung, der Kunde, das Marketing, der Vertrieb, das Management, der Projektmanager (Projektleiter), der Projektmitarbeiter, die Schnittstellen (!), ...	selbst
Ziel	Bezeichnet einen in der Zukunft liegenden, gegenüber dem Gegenwärtigen im Allgemeinen veränderten, erstrebenswerten und angestrebten Zustand (Zielvorgabe)	selbst



Teil IV

Anhang B

- Normen und Standards
- Zertifizierungen für Personen
- Das IREB (Kurzdarstellung, CPRE-Kurzdarstellung, Die CPRE-Ausbildungslevel, Die Struktur)
- Das IIBA (Kurzdarstellung, CBAP-Kurzdarstellung, Die sechs Wissensgebiete (Knowledge Areas))
- Das PMI (Kurzdarstellung, PBA-Kurzdarstellung, Die fünf Domänen (Domains), Anmerkungen)

Seite
221–236



Zum Requirements Engineering gibt es eine Reihe von Normen, die direkt oder indirekt das Requirements Engineering betreffen – diese sind auf der nächsten Folie gelistet.

Seit 2011 gibt es die **ISO/IEC/IEEE 29148:2011 “Systems and software engineering – Life cycle processes – Requirements engineering”**, welche die drei folgenden ISO/IEC/IEEE-Normen ersetzt:

- 830-1998: “IEEE Recommended Practice for Software Requirements Specification”
- 1233-1998: “IEEE Guide for Developing System Requirements Specifications”
- 1362-1998: “IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document“

Sie ergänzt zudem die Normen ISO/IEC 12207 und ISO/IEC 15288.



- IEEE 610-1990: Standard Glossary of Software Engineering Terminology (1990)
- ~~IEEE 830-1998: Recommended Practice for Software Requirements Specifications (1998)~~
- IEEE 1220-1998: Standard for Application and Management of the System Engineering Process (1998)
- ~~IEEE 1233-1998: Guide for Developing of System Requirements Specifications (1998)~~
- ~~IEEE 1362-1998: Guide for Information Technology – System Definition (1998)~~
- ISO/IEC 9126:2001: Software Engineering – Product Quality (2001)
- ISO/IEC 12207:1997: Software Life Cycle Processes (1997)
- ISO/IEC 15288:2002: System Life Cycle Processes (2002)
- **ISO/IEC/IEEE 29148:2011: “Systems and software engineering – Life cycle processes – Requirements engineering” (2011)**
- ISO/IEC 61508:200x: Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme
- VDI-Richtlinie 2519 Blatt 1: Vorgehensweise bei der Erstellung von Lasten-/Pflichtenheften (2001)



Seit einigen Jahren wetteifern zwei international agierende Organisationen um die Vereinheitlichung des REs und der zu vergebenden Personenzertifikate:

- In 2007 etablierte sich in Deutschland das „International Requirements Engineering Board“ (**IREB**) mit dem Personenzertifikat „Certified Professional for Requirements Engineering“ (**CPRE**)
- Seit 2003 existiert das US-amerikanische (aber in Kanada gegründete) „International Institute of Business Analysis“ (**IIBA** – in den USA wird Requirements Engineering zum Teil der Business Analysis gleichgesetzt), welches sich stark an dem erfolgreichen Modell des PMI (Project Management Institute) orientiert und das Personenzertifikat „Certified Business Analysis Professional“ (**CBAP**) vergibt

Im Jahr 2014 startete das **PMI** (Project Management Institute) mit einem eigenen Zertifikat, dem **PMI-PBA** – „Professional in Business Analysis“:

- Der PMI-PBA verlangt ähnliche Voraussetzungen wie der CBAP. Durch die Kopplung mit weltweit erfolgreichen PMP-Zertifikat (PMP=Project Management Professional) ist der PBA insbesondere für PMP-Zertifikatsinhaber interessant

Die drei Organisationen mit ihren Zertifikaten werden hier kurz vorgestellt.

IREB e.V. – International Requirements Engineering Board

- Bekanntester (deutscher) RE-Fachverband, internationaler Anspruch
- Gegründet 2006 in Erlangen
- Sitz in Nürnberg
- Die Mitgliedschaft im IREB ist personenbezogen, jedoch nicht offen; im Board sind 82 Mitglieder vertreten, die als unabhängige und international anerkannte Experten aus Industrie, Beratung, Forschung und Lehre berufen wurden; der Mitgliedsbeitrag ist unbekannt; Stand: 03/2017
- Website: <https://www.ireb.org/de>
- Basisliteratur des IREB:
/IREB15/ = /Pohl15a/ Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt, Hannover 4. Auflage 2015, ISBN 978-3-86490-283-3
- Zertifizierungen:
 - a) **CPRE – Certified Professional for Requirements Engineering, 3 Levels**

CPRE – Certified Professional for Requirements Engineering (Foundation Level)

- Einstieg in ein dreistufiges, international anerkanntes Zertifizierungsangebot
- Starke Praxisorientierung, Sprache: Deutsch oder englisch
- Existiert seit 2007
- Voraussetzungen: Keine, auch kein Schulungskurs
- Anmeldeverfahren: Anmeldung beim Trainingsprovider oder bei der Zertifizierungsstelle (in Dtl. beispielsweise ISQI)
- Aufwand: Etwa 3 Tage Schulung + Eigenvorbereitung
- Kosten: Etwa 250 € für die Prüfung und 1.500 € für einen optionalen Schulungskurs (3 Tage) in Deutschland; in anderen Ländern abweichend
- Prüfungsrahmen: 75 Minuten, 45 Fragen (Multiple Choice), 70 % der Punktzahl zum Bestehen notwendig, Durchfallquote etwa 20-25 %
- Rezertifizierung nicht erforderlich
- Über 29.000 Foundation-Level-Zertifizierte weltweit (Stand: 03/2017), davon etwa 13.000 in Deutschland, etwa 3.000 in Österreich und 5.500 in der Schweiz (siehe <https://www.ireb.org/de/service/statistics/>)

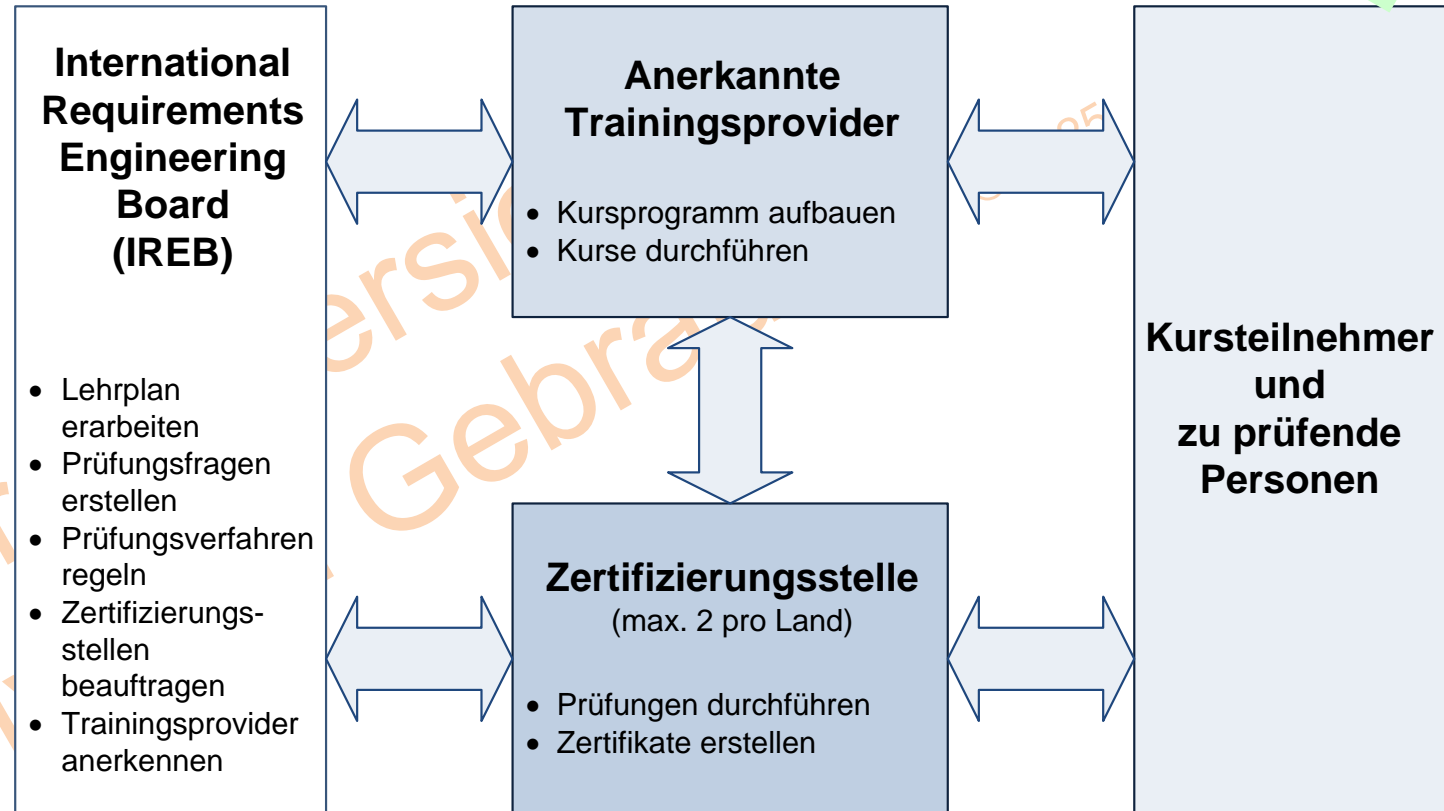
Das IREB kennt drei CPRE-Ausbildungslevel, die aufeinander aufbauen, derzeit jedoch noch nicht komplett verfügbar sind:

Kürzel	Ausbildungslevel	Charakterisierung	Verfügbar
FL	Foundation	<ul style="list-style-type: none"> • Grundlegendes Verständnis methodischer Ansätze 	2007
AL	Advanced	<ul style="list-style-type: none"> • Spezialisierung auf ein oder mehrere Module* • Anwendung der Techniken und Methoden 	2011
EL	Expert	<ul style="list-style-type: none"> • Bei drei oder mehreren CPRE-AL-Zertifikaten 	in Planung

* Das Advanced-Zertifikat wird für ein Spezialgebiet (Modul) vergeben – zurzeit (Stand: 03/2017) sind verfügbar: Requirements Modeling, Requirements Management, Elicitation and Consolidation. Geplant sind folgende weitere Module: Product Management und Bidding. Man kann daher mehrere CPRE-AL-Zertifikate erwerben: Bei mehr als drei hat man die Expert-Zertifizierung erreicht.

Das Foundation Level ist mit Abstand am stärksten verbreitet.

Beim IREB sind die Trainingsprovider und die Zertifizierungsstellen (= Prüfungsstellen) voneinander getrennt. Es können daher auch Trainingsanbieter CPRE-Schulungen abhalten, die nicht im Board selbst vertreten sind.



Das IREB folgt mit der CPRE-Zertifizierung der DIN EN ISO/IEC 17024:2003: „Konformitätsbewertung – Allgemeine Anforderungen an Stellen, die Personen zertifizieren.“



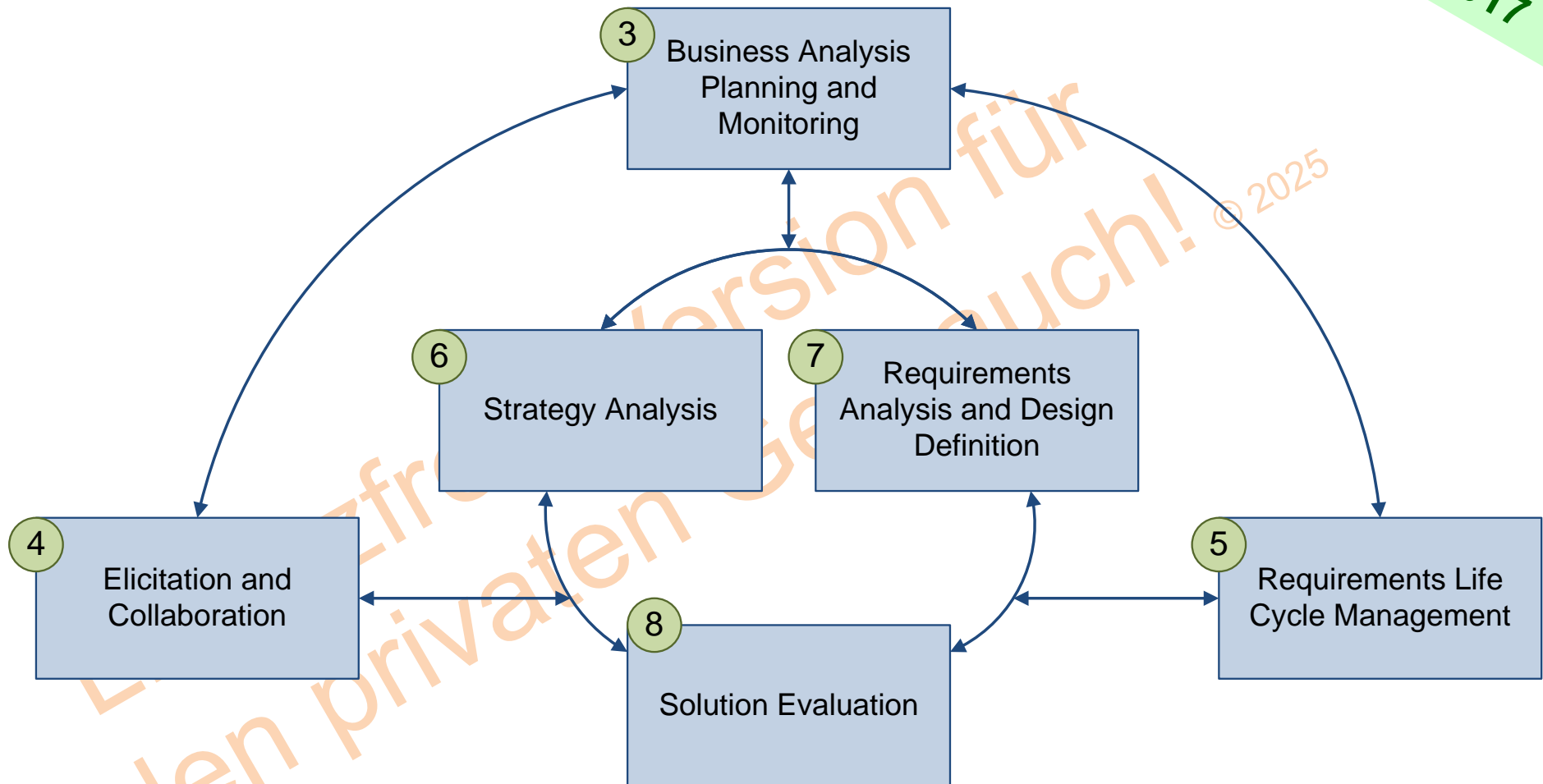
IIBA – International Institute of Business Analysis

- Über 30.000 Mitglieder (Stand: 03/2017), vorwiegend in den USA; über 300 Unternehmensmitglieder (Corporate Members)
- Gegründet Oktober 2003 in Kanada; Sitz in Toronto, Kanada
- Über 120 Landesgesellschaften (Chapter) in über 60 Ländern
- Mitgliedschaft direkt (über das IIBA in den USA) – Beitrag etwa \$125 (110 €/Jahr + 20 € German-Chapter-Mitgliedschaft)
- Website: <https://www.iiba.org>
- Das deutsche Chapter existiert seit 2010: <https://germany.iiba.org/>
- Basisliteratur des IIBA – der BABOK Guide:
/BBG15/ IIBA: A Guide to the Business Analysis Body of Knowledge (BABOK Guide), International Institute of Business Analysis, Marietta, Georgia 3rd Edition 2015, ISBN 978-1-927584-02-6
- Wesentliche Zertifizierungen:
 - a) **CBAP** („**Certified Business Analysis Professional**“)
 - b) **CCBA** („**Certification of Competency in Business Analysis**“) (seit 2011)

CBAP – Certified Business Analysis Professional

- Voraussetzungen: Nachweis über 7.500 h BA-Tätigkeiten (davon jeweils mindestens 900 in jedem Wissensgebiet) gemäß BABOK innerhalb der letzten 10 Jahre, 2 Referenzen, Schulabschluss („High-School-Degree“),
- Anmeldeverfahren: Via Internet ähnlich PMI
- 3,5-stündige Multiple-Choice-Prüfung am Computer (bei Prometric) in englisch (auch in deutsch möglich, dann aber deutlich teurer)
- 150 Fragen, die mit je einem Punkt bewertet werden, 150 Punkte erreichbar, ??* Punkte zum Bestehen notwendig (?? %)
- Rezertifizierung alle drei Jahre; hierzu müssen so genannte 60 CDUs (Continuing Development Units) über Weiterbildung o.Ä. erworben werden; Faustregel: 1 CDU = 1h Weiterbildung
- Aufwand: Für den CBAP werden minimal drei Tage Schulung (21 h) angesetzt und einige Stunden Selbstlernaufwand
- Kosten: Ab 1.000 € für die Schulung; \$325 / \$450 für die Prüfung (IIBA-Mitglieder / Nicht-Mitglieder); \$125 für die Anmeldung
- CBAP-Zertifizierung existiert seit 2003; etwa 7.800 CBAPs (weltweit, Stand: 03/2017); davon etwa 70 in Dtd., internationale Anerkennung

* Eine Bestehenspunktzahl wird nicht veröffentlicht!



Die Nummerierung der Wissensgebiete (3 bis 8) entspricht den Buchkapiteln des BABOK Guides /BBG15/.

Ch	Knowledge Area	Definition/Description
3	Business Analysis Planning and Monitoring	describes the tasks that business analysts perform to organize and coordinate the efforts of business analysts and stakeholders. These tasks produce outputs that are used as key inputs and guidelines for the other tasks throughout the BABOK guide.
4	Elicitation and Collaboration	describes the tasks that business analysts perform to prepare for and conduct elicitation activities and confirm the results obtained. It also describes the communication with stakeholders once the business analysis information is assembled and the ongoing collaboration with them throughout the business activities.
5	Requirements Life Cycle Management	describes the tasks that business analysts perform in order to manage and maintain requirements and design information from inception to retirement. These tasks describe establishing meaningful relationships between related requirements and designs, and assessing, analyzing and gaining consensus on proposed changes to requirements and designs.
6	Strategy Analysis	describes the business analysis work that must be performed to collaborate with stakeholders in order to identify a need of strategic or tactical importance (The business need), enable the enterprise to address that need, and align the resulting strategy for the change with higher - and lower - level strategies.
7	Requirements Analysis	describes the tasks that business analysts perform to structure and organize requirements discovered during elicitation activities, specify and model requirements and designs, validate and verify information , identify solution options that meet business needs, and estimate the potential value that could be realized for each solution option. This knowledge area covers the incremental and iterative activities ranging from the initial concept and exploration of the need through the transformation of those needs into a particular recommended solution.
8	Solution Evaluation	describes the tasks that business analysts perform to assess the performance of an value delivered by a solution in use by the enterprise, and to recommend removal of barriers or constraints that prevent the full realization of the value.

/BBG15/



PMI – Project Management Institute

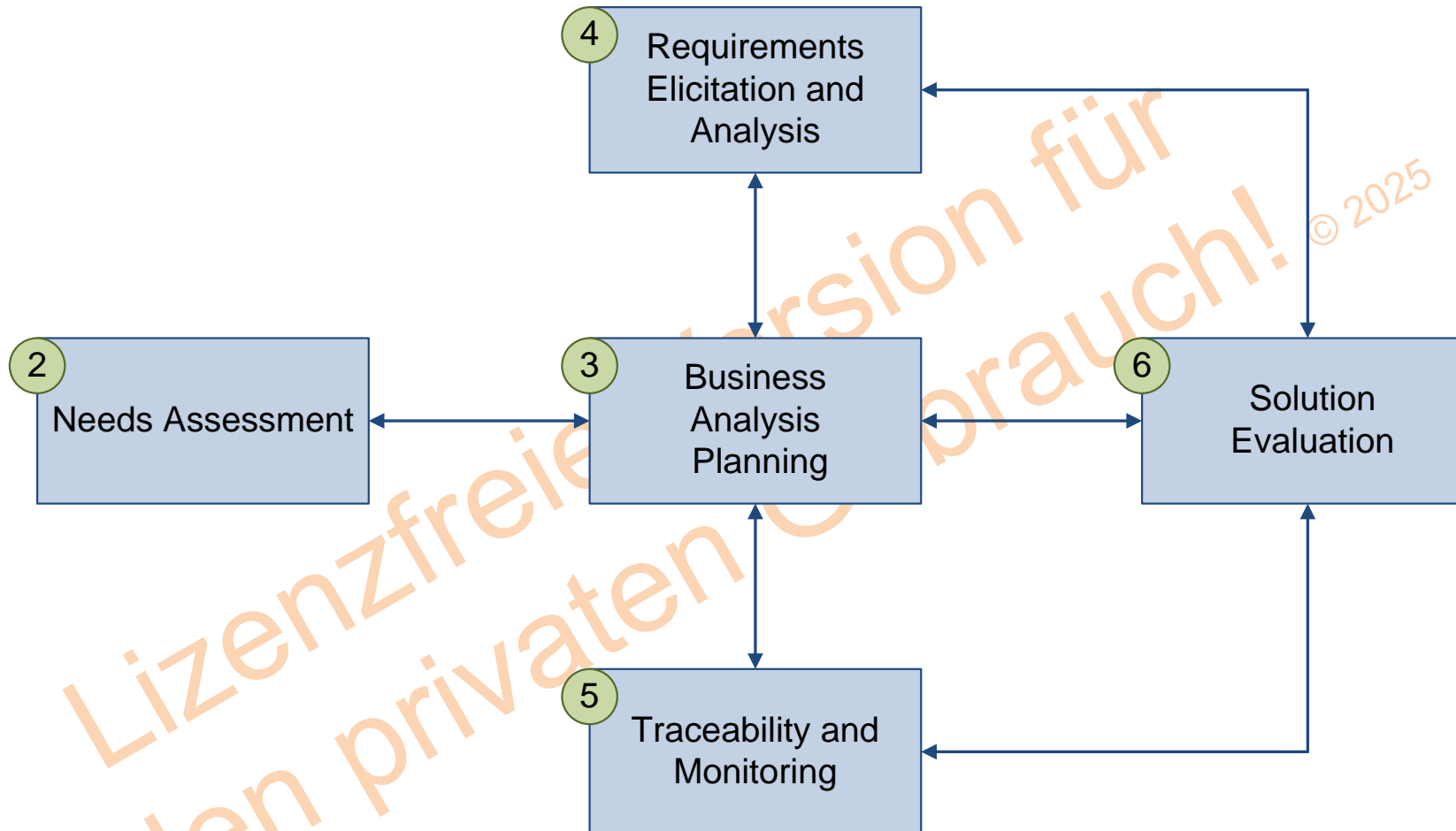
- Etwa 470.000 Mitglieder (Stand: 03/2017), vorwiegend in den USA und China sowie etwa 3.500 in Dtl.; nur persönliche Mitglieder
- Gegründet 1969; Sitz in Philadelphia, Pennsylvania, USA
- Über 280 Landesgesellschaften in 67 Ländern
- Schwerpunkt: Projektmanagement-Zertifizierung – der PMP (Project Management Professional) mit über 750.000 Zertifizierten weltweit
- Mitgliedschaft direkt (über das PMI in den USA) – Beitrag etwa 100 €/Jahr (+ Chapter-Beiträge, etwa 20-30 €/Jahr)
- Website: <https://www.pmi.org>; vier deutsche Landesgesellschaften (Chapter), die größte ist das Chapter Southern Germany: <https://pmi-sgc.de/>
- Basisliteratur des PMI für Business Analysis – der BA Practice Guide: **/BAPG15/** Project Management Institute: Business Analysis For Practitioners: A Practice Guide, Project Management Institute, Philadelphia, Pennsylvania 2015, ISBN 978-1-62825-069-5
(herunterladbar unter <https://www.pmi.org/pmbok-guide-standards/practice-guides/business-analysis>)
- Zertifizierung für Business Analysis (BA) seit Ende 2014 möglich:
PMI-PBA („Professional in Business Analysis“)

PMI-PBA – Professional in Business Analysis

- Verfahren generell wie beim PMP
- Voraussetzungen (ähnlich dem PMP): Nachweis entweder über
 - 7.500 h BA-Tätigkeiten innerhalb der letzten 8 Jahre Schulabschluss („High-School-Degree“)
 - 4.500 h BA-Tätigkeiten innerhalb der letzten 8 Jahre, Hochschulabschluss („Bachelor-Degree“)
- Vierstündige Multiple-Choice-Prüfung am Computer (bei Prometric), Basissprache: englisch, aber deutsch ist ebenfalls möglich
- 200 Fragen, 175 werden mit je einem Punkt bewertet, 175 Punkte erreichbar, 106 Punkte notwendig (60,6 %)
- Rezertifizierung alle 3 Jahre; hierzu müssen so genannte 60 PDUs (Personal Development Units) in drei Jahren (über Weiterbildung o.Ä.) erworben werden; Faustregel: 1 PDU = 1h Weiterbildung; Renewal Fee: \$150 non-PMI members / \$60 PMI members
- Aufwand: Für den PBA werden minimal fünf Tage Schulung (35 h) angesetzt und einige hundert Stunden Selbstlernaufwand
- Kosten: Ab 2.000 € für die Schulung; für die Prüfung: \$555 for non-PMI members, \$405 PMI members (im EURO-Raum: 465 € Nicht-PMI-Mitglieder, 340 € PMI-Mitglieder)
- PBA-Zertifizierung existiert seit 2014, auch international; 1.127 Zertifizierte weltweit Ende 2016 insgesamt, davon 558 neu in 2016

Das PMI (3/4): Die fünf Domänen (Domains)

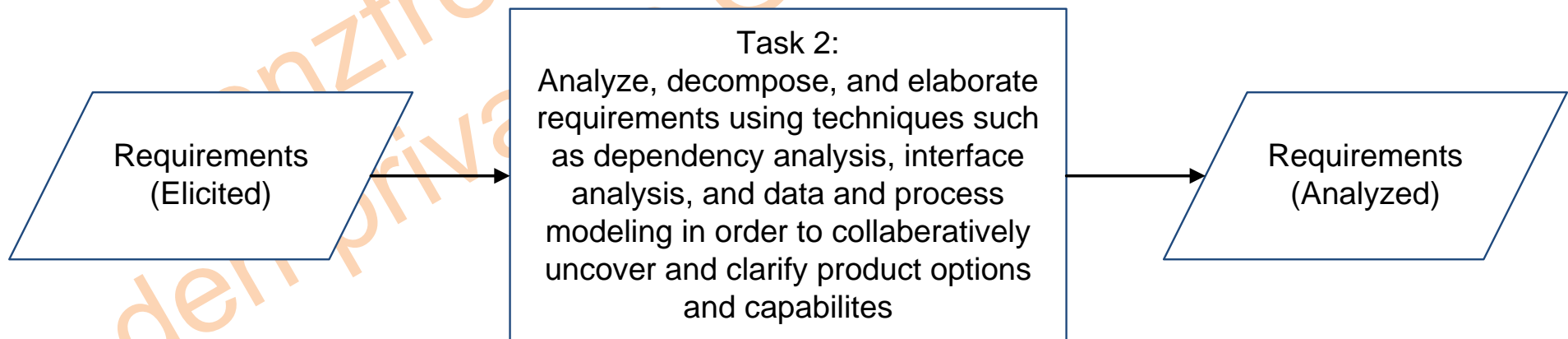
Stand:
03/2017



Die Nummerierung der Domänen (2 bis 6) entspricht den Buchkapiteln des Business Analysis Practical Guides /BAPG15/.

Beim PMI wird die Business Analysis über Domänen definiert, denen wiederum Tasks zugeordnet sind – dies gleicht dem Vorgehen beim IIBA. Den Tasks werden jeweils Werkzeuge und Methoden (*Tools and Techniques*) zugeordnet, wobei einzelne W&M bei verschiedenen Tasks zugeordnet werden können.

Nach Larson /Larson15/ wird jede der 28 Tasks in der Form ITO (Input, Task, Output) folgendermaßen dargestellt (hier das Beispiel „Analyze, decompose, and elaborate requirements“):





Teil IV

Anhang C

- Reifegradmodelle
- Das CMMI und Requirements Engineering
- Requirements Management (REQM) im CMMI: Ziele und Praktiken
- Requirements Development (RD) im CMMI: Ziele und Praktiken
- Das Reifegradmodell nach Hood
- Technische Dokumentation
- Die Guides: PMBOK, BABOK und SWEBOK

Seite
237–244



Reifegradmodelle dienen zur Bestimmung der Prozessfähigkeit (auf einem bestimmten Gebiet) von Unternehmen. Hierzu werden typischerweise die benötigten Fähigkeiten auf einer (drei- bis sechsstufigen) Skala untergebracht, bei der die Skalenwerte benannt werden. Reifegradmodelle verlangen keine konkrete Vorgehensweise (das „Wie“), sondern benennen Prozesse, die aus bewährten Praktiken hervorgehen (das „Was“).

Die Unternehmen werden dann anhand der Skalen bewertet und streben oftmals anschließend einen höheren Reifegrad an. Hierzu müssen sie die geforderten Prozesse gezielt einführen oder verbessern.

Das wohl bekannteste Reifegradmodell ist das CMMI (Capability Maturity Model Integration) des SEI (Software Engineering Institute) der Carnegie Mellon University (Pittsburgh), welches sich mit den Prozessen der Produktentwicklung, den Produkteinkauf und die Serviceerbringung beschäftigt /Wiki-d/.

Für das Requirements Engineering hat Hood /Hood05/ ein Reifegradmodell entwickelt, welches die Qualitätsattribute nach IEEE in drei Stufen umsetzt. Dieses Modell wird auf einer der nächsten Folien vorgestellt.



Im CMMI (Capability Maturity Model Integration) /Chrissis11a, Chrissis11b/ werden zwei Requirements-Prozessgebiete betrachtet:
Dies ist das Management (REQM) und das Development (RD). Dabei muss das Management bereits im Reifegrad 2 gewährleistet sein, während das Ausarbeiten für Stufe 3 vorgesehen ist.

Prozessgebiet	Kürzel	Deutsch	Kategorie	Rfgr
Requirements Management	REQM	Anforderungsmanagement	Engineering	2
Requirements Development	RD	Anforderungsentwicklung	Engineering	3

Für Unternehmen bedeutet dies: Der im Allgemeinen minimal angestrebte Reifegrad 2 verlangt das Verwalten von Anforderungen; wird Reifegradstufe 3 angestrebt, so muss man sich gezielt mit der Entwicklung von Anforderungen auseinandersetzen.

Pohl /Pohl08/ widmet dem Thema RE und CMMI ein komplettes Kapitel.



Das CMMI benennt fünf spezifische Praktiken für das Requirements Management (REQM).

SG 1 Manage Requirements		SG 1 Anforderungen verwalten	
SP 1.1	Understand Requirements	SP 1.1	Anforderungen verstehen
SP 1.2	Obtain Commitment of Requirements	SP 1.2	Zusagen zu Anforderungen einholen
SP 1.3	Manage Requirements Changes	SP 1.3	Anforderungsänderungen verwalten
SP 1.4	Maintain Bidirectional Traceability of Requirements	SP 1.4	Bidirektionale Nachverfolgbarkeit von Anforderungen aufrecht erhalten
SP 1.5	Ensure Alignment Between Project Work and Requirements	SP 1.5	Abstimmung zwischen Projektarbeit und Anforderungen sicherstellen



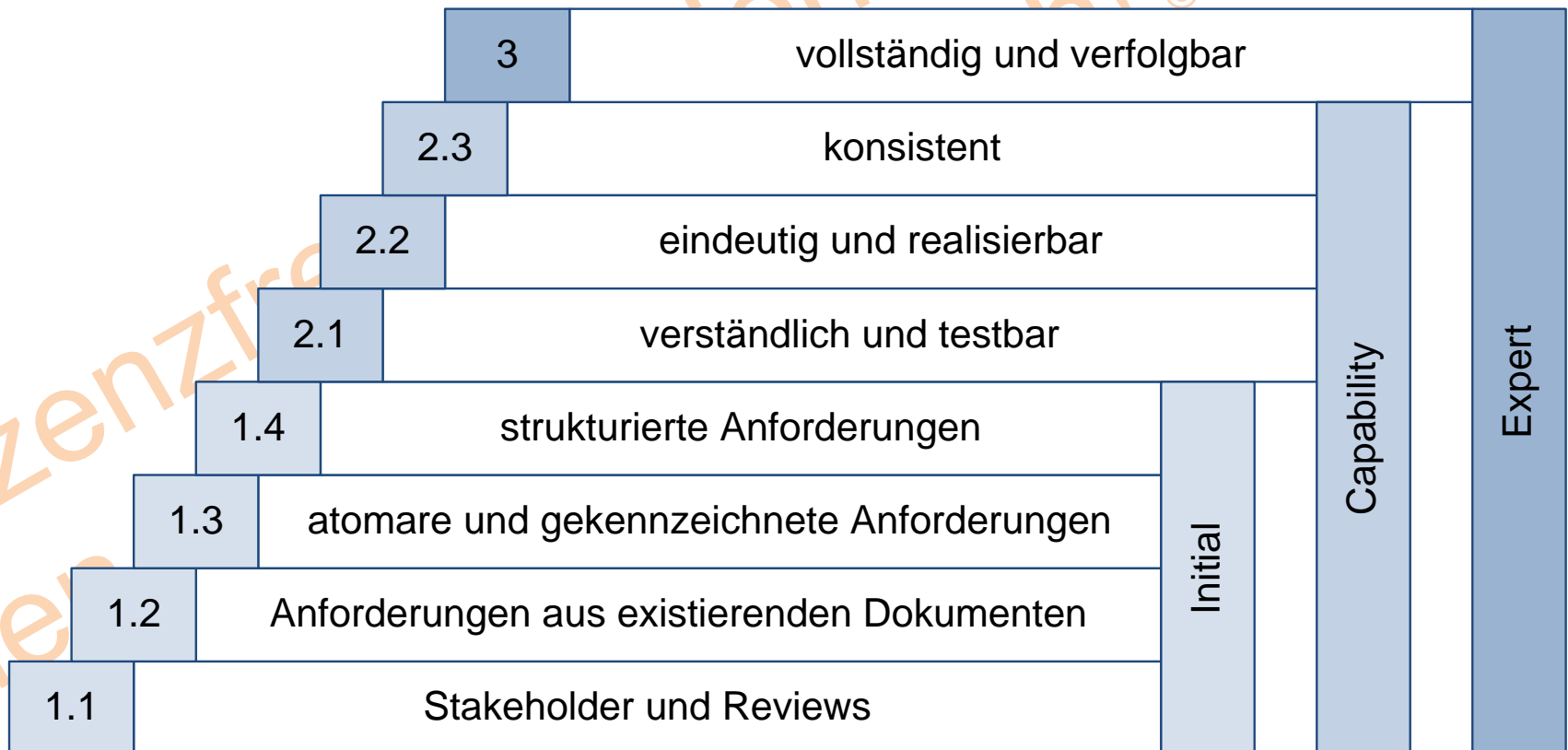
Das CMMI benennt drei spezifische Ziele und zehn spezifische Praktiken für das Requirements Development (RD).

SG 1 Develop Customer Requirements	
SP 1.1	Elicit Needs
SP 1.2	Transform Stakeholder Needs into Customer Requirements
SG 2 Develop Product Requirements	
SP 2.1	Establish Product and Product Component Requirements
SP 2.2	Allocate Product Component Requirements
SP 2.3	Identify Interface Requirements
SG 3 Analyze and Validate Requirements	
SP 3.1	Establish Operational Concepts and Scenarios
SP 3.2	Establish a Definition of Required Functionality
SP 3.3	Analyze Requirements
SP 3.4	Analyze Requirements to Achieve Balance
SP 3.5	Validate Requirements

SG 1 Kundenanforderungen entwickeln	
SP 1.1	Bedürfnisse herausfinden
SP 1.2	Bedürfnisse der Stakeholder in Kundenanforderungen überführen
SG 2 Produkthanforderungen entwickeln	
SP 2.1	Anforderungen an Produkte und Produktbestandteile etablieren
SP 2.2	Anforderungen an Produktbestandteile zuweisen
SP 2.3	Schnittstellenanforderungen identifizieren
SG 3 Anforderungen analysieren und validieren	
SP 3.1	Betriebskonzepte und Anwendungsszenarien etablieren
SP 3.2	Definitionen erforderlicher Funktionalität etablieren
SP 3.3	Anforderungen analysieren
SP 3.4	Anforderungen analysieren und abgleichen
SP 3.5	Anforderungen validieren



Das Reifegradmodell von Hood /Hood05/ ist ein reines Modell für das RE. Dabei wird zwischen drei Basisstufen (Initial, Capability, Expert) unterschieden, die wiederum Teilstufen kennen. Die erforderlichen Fähigkeiten orientieren sich an den Vorgaben für Anforderungen nach IEEE 830.





Die technische Dokumentation beschäftigt sich mit der Beschreibung von technischen Produkten (für Anwender), beispielsweise mit der Erstellung von Benutzerdokumentationen (wie Handbücher). Ein Großteil der Anforderungen sollte sich später im Bedienungshandbuch wiederfinden.

Die technische Dokumentation ist eine dem RE verwandte Disziplin – Tools könnten (und sollten) gemeinsam eingesetzt werden.

Schlagworte: DITA – dies ist ein XML-basierter Ansatz zur Klassifizierung von Dokumenten

Tools: Z.B. AuthorIT – <https://www.author-it.com/>

Links: <http://www.doku.info>; <https://www.tanner.de>

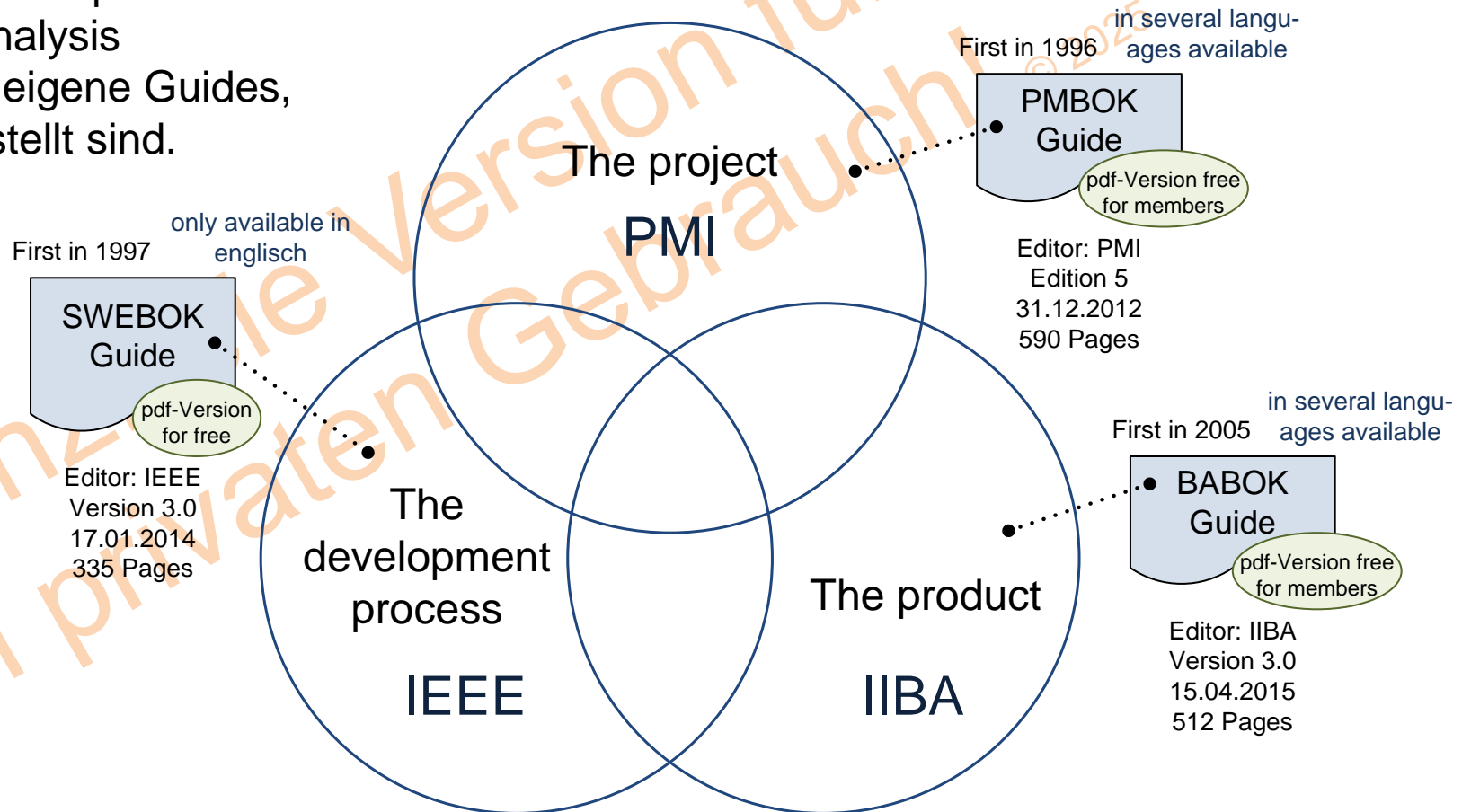
Literatur hierzu:

/Juhl15/ Dietrich Juhl: Technische Dokumentation: Praktische Anleitung und Beispiele, Springer, Berlin 3. Auflage 2015, ISBN 978-3-662-46864-7

Zu den Disziplinen

- Project Management
- Software Development
- Business Analysis

gibt es jeweils eigene Guides,
die hier dargestellt sind.





Teil IV

- Weitere öffentliche Präsentationen des Autors
- Meine Dienstleistungen – Das kann ich für Sie tun
- Kontakt zum Autor

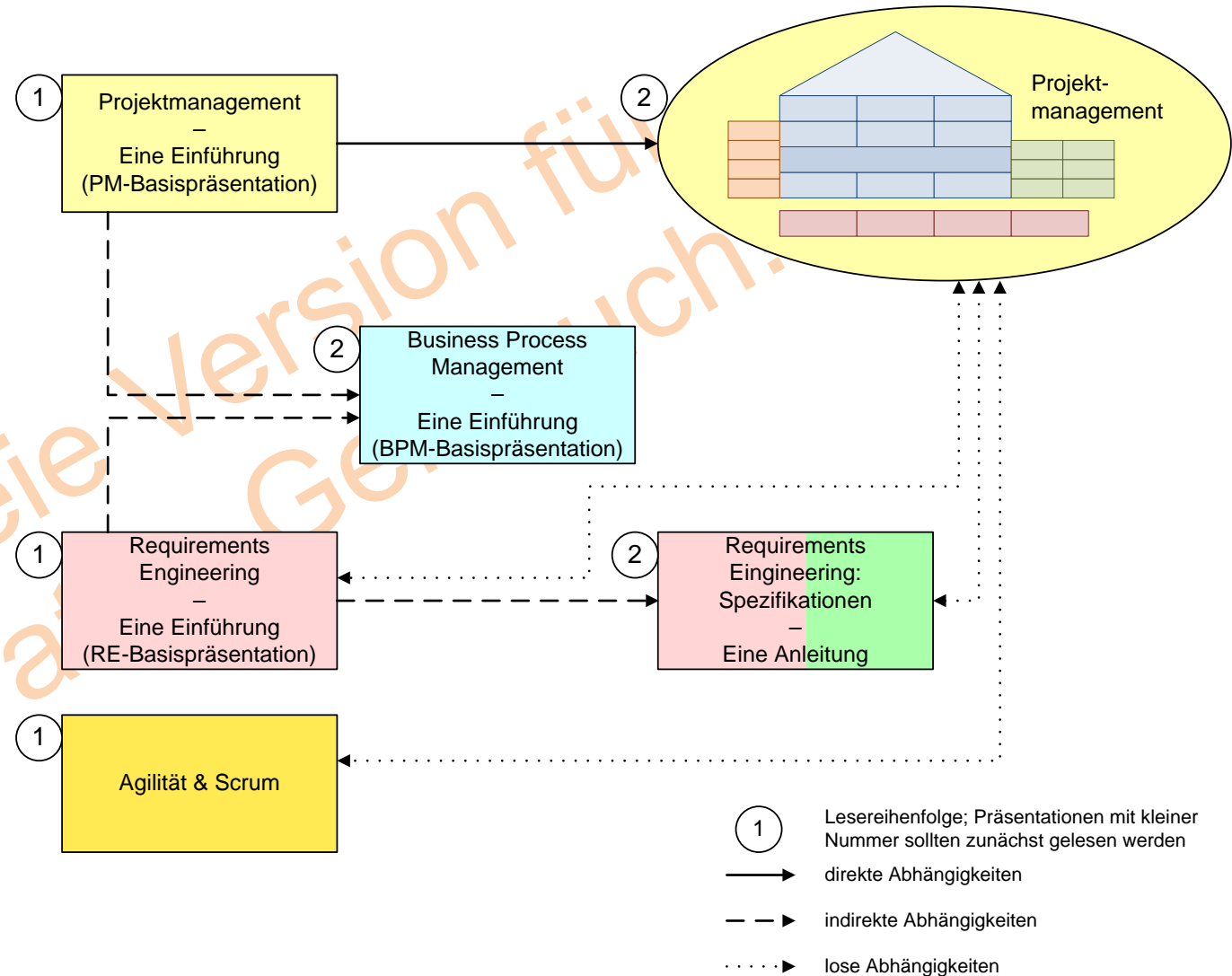
Anhang D

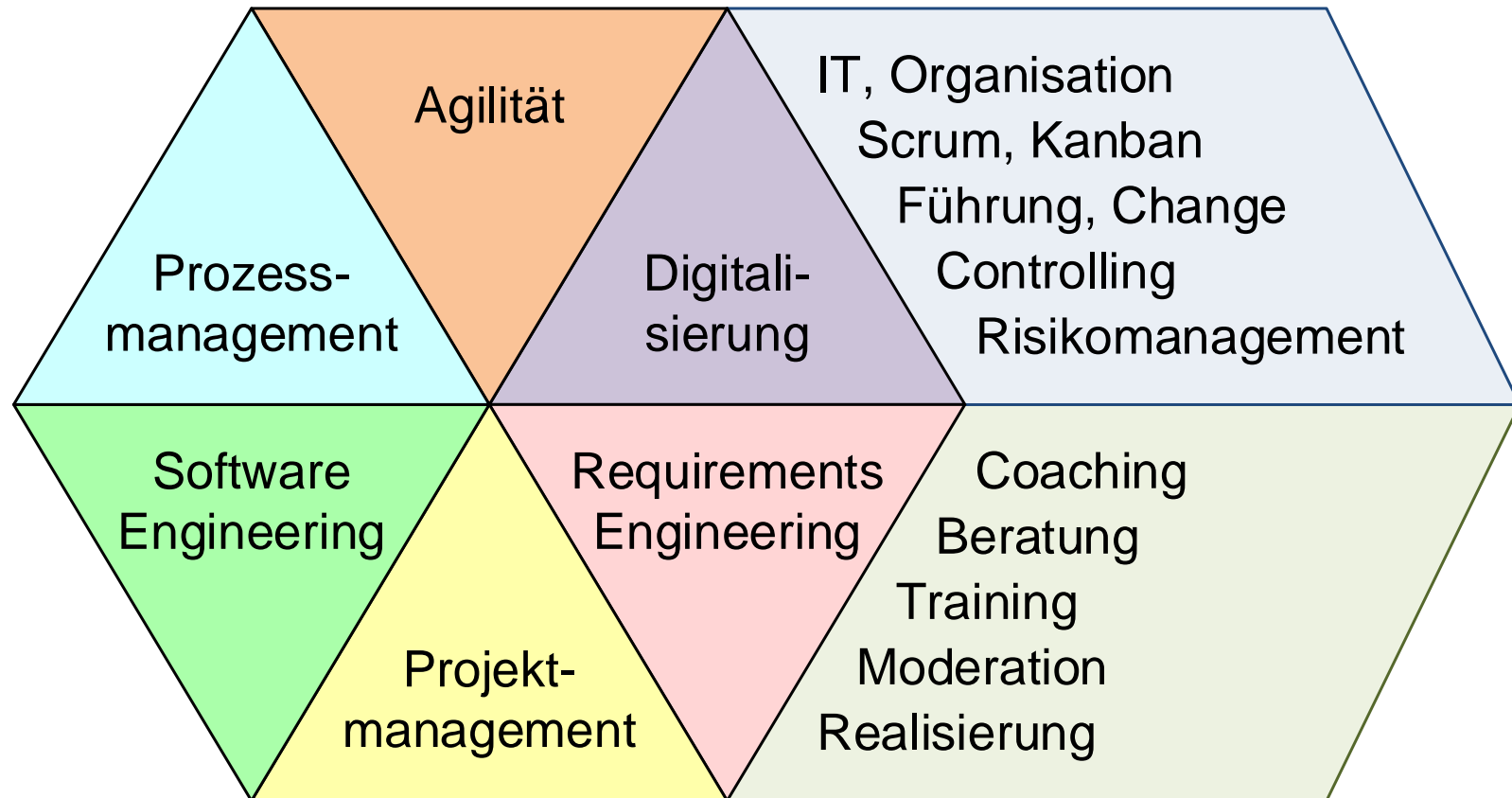
Seite
245–248



Auf folgende Präsentationen des Autors wurde in dieser Ausarbeitung hingewiesen, die auch auf der Website frei verfügbar sind:

- Requirements Engineering: Spezifikationen
- Projektmanagement: Eine Einführung
- Projektmanagement: Zertifizierungen & Normen





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>