

IREB Foundation Level

DI (FH) Elisabeth Wölfel

IREB-FL Struktur

Tag	Inhalt
Tag 1	LE 1, LE 2, LE 3.1, LE 3.5 bis LE 3.8, LE 4.1 bis LE 4.2
Tag 2	LE 3.2 bis LE 3.4
Tag 3	LE 4.3 bis LE 4.4, LE 5, LE 6, LE 7
Tag 4	Zertifizierungsprüfung

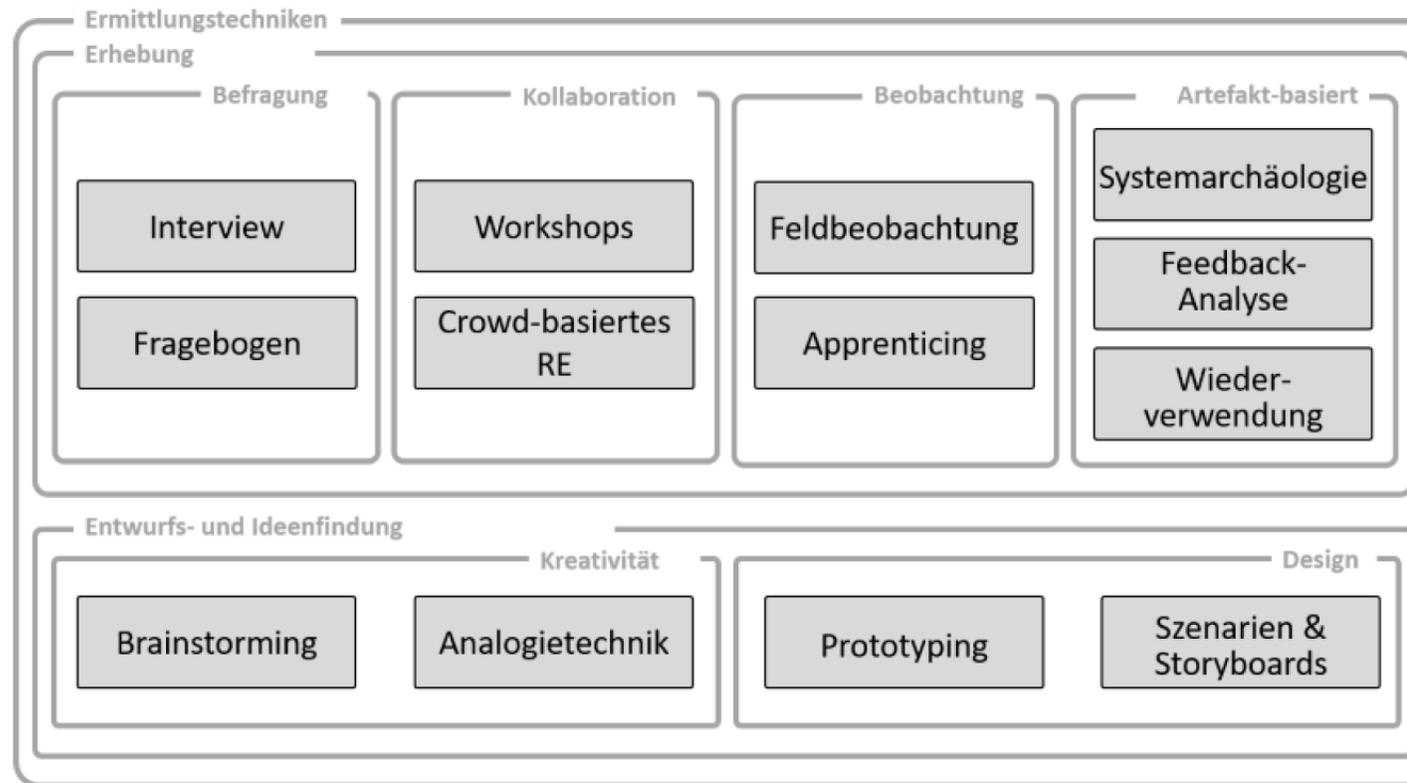
IREB-FL Tagesablauf

Tag	Inhalt
9:00 – 10:30	LE 4.3
10:30 – 10:45	Pause
10:45 – 12:15	LE 4.4, LE 5.1 - LE 5.3
12:15 – 13:15	Mittagspause
13:15 – 14:45	LE 6.1 – LE 6.8
14:45 – 15:00	Pause
15:00 – 17:00	LE 7.1 & LE 7.2

4.3 Auflösen von Konflikten bezüglich der Anforderungen

Welche Ermittlungstechniken kennen sie bereits?

Techniken der Anforderungsermittlung



Erhobene Anforderungen können sein:

- Inkonsistent
- Inkompatibel
- Widersprüchlich

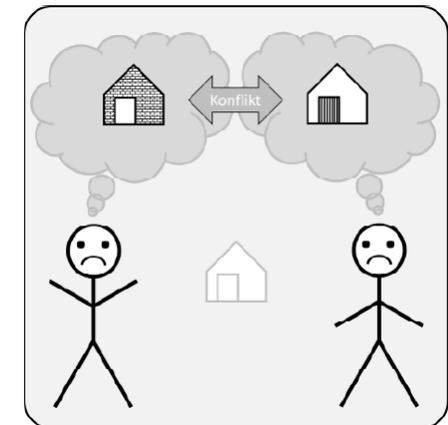
- Aufgabe des Requirements Engineers: die Stakeholder*innen zu einem **gemeinsamen Verständnis** des gesamten Anforderungssatzes bringen
- Überprüfung auf Widersprüche und Abstimmung muss über das gesamte RE hinweg erfolgen

Konflikt

„Eine Interaktion zwischen Akteur*innen (Individuen, Gruppen, Organisationen usw.), wobei **wenigstens ein/e Akteur*in eine Differenz bzw. Unvereinbarkeiten** im Wahrnehmen und Denken bzw. Vorstellen und im Fühlen und im Wollen mit dem/ der anderen Akteur *in in der Art erlebt, dass beim Verwirklichen dessen, was der/ die Akteur *in denkt, fühlt oder will **eine Beeinträchtigung durch eine/n andere/n Akteur*in** erfolge.“ [Glas1999]

Anforderungskonflikte:

- Widersprüchliche Meinungen zu Anforderungen
- Unaufgelöste Konflikte gefährden die Akzeptanz des Systems



Handbuch des CPRE Certified Professional for Requirements Engineering Foundation Level - Version 1.0.0, S 96



Schritte der Konfliktlösung

1. Konfliktidentifizierung
2. Konfliktanalyse
3. Konfliktlösung
4. Dokumentation der Konfliktlösung

1. Konfliktidentifizierung

- Konfliktidentifizierung – Zeichen in der ...
 - Kommunikation:
 - Verleugnung, Gleichgültigkeit, Pedanterie, ständiges Nachfragen nach mehr Details, absichtliche Falschinterpretationen, Verheimlichung oder Delegation
 - Dokumentation:
 - widersprüchliche Aussagen von Stakeholder*innen, konträre Ergebnisse aus der Analyse von Dokumenten oder Systemen, Inkonsistenzen auf verschiedenen Detailebenen und eine uneinheitliche Verwendung von Begriffen

2. Konfliktanalyse

- Anforderungskonflikt → Requirements Engineer
- Andere Konflikte → Projektleitung, Teamleitung, ...

- Aspekte im Zuge der Konfliktanalyse
 - Sachverhalt – das eigentliche Problem dahinter
 - Betroffene Anforderungen
 - Beteiligte Stakeholder*innen
 - Konfliktpositionen - um das Problem noch besser zu verstehen
 - Ursache des Konflikts
 - Historie des Konflikts
 - Folgen der Lösung oder Nichtlösung – geschätzte Kosten und Risiken
 - Projekttrandbedingungen – persönliche, organisatorische, inhalts- oder domänenspezifische Randbedingungen um den Lösungsraum zu definieren

3. Konfliktlösung

- Konfliktlösungstechnik wählen
 - Wird Technik akzeptiert
 - Auswahl neutral und objektiv durch Requirements Engineer

4. Dokumentation der Konfliktlösung

- Nachvollziehbarkeit hinsichtlich
 - Annahmen in Bezug auf den Konflikt und seine Auflösung
 - in Betracht gezogene Alternativen
 - Randbedingungen, die die gewählte Technik und/oder Auflösung beeinflussen
 - Die Art und Weise, wie der Konflikt gelöst wurde, inklusive der Gründe, die zur Wahl der Konfliktlösungstechnik geführt haben
 - Entscheider*innen und weitere Beteiligte
- Um Entscheidungen zu verstehen und falsche Interpretation zu vermeiden
- Um wiederholte Behandlung von Konflikten zu vermeiden
- Um einen Konflikt erneut untersuchen/auflösen zu können, falls sich die Auflösung als falsch oder nicht geeignet herausstellt
- Dokumentation mittels Konfliktsteckbrief

Keine Angst vor Konflikten

- Es gibt sie immer
- Wenn sie keine haben, haben sie sie noch nicht gefunden
- Je später ein Problem entdeckt wird, desto teurer wird es, es zu lösen



Konflikttypen im Rahmen der Konfliktanalyse (Moore 2014)

- Sachkonflikt
- Datenkonflikt
- Interessenkonflikt
- Wertekonflikt
- Beziehungskonflikt
- Strukturkonflikt

Sachkonflikt

- Ursachen:
 - Unterschiedliche Bedürfnisse
 - Unterschiedliche Benutzung des Systems
 - Z.B. unterschiedliche Rechtslage

- Mögliche Lösung:
 - Fakten analysieren, dokumentieren -> Einigung, Kompromisse, Varianten, Abstimmungen

Datenkonflikt

- Ursachen:
 - Inkonsistente Daten
 - Unterschiedliche Interpretation gleicher Daten
 - Z.B. schlechte Kommunikation, fehlende Hintergrunddaten, kulturelle Unterschiede, Annahmen & Schätzungen etc.

- Mögliche Lösung:
 - Analyse der unterschiedlichen Interpretationen durch Einfühlungsvermögen und Kommunikation

Interessenkonflikt

- Ursachen:
 - Unterschiedliche Positionen der Konfliktparteien
 - Z.B. durch unterschiedliche persönliche Ziele, Ziele einer Gruppe, Ziele einer Position etc.
- Mögliche Lösung:
 - Verständnis für unterschiedliche Interessen zeigen und Lösung durch Identifikation und Stärkung gemeinsamer Interessen

Wertekonflikt

- Ursachen:
 - Unterschiedliche Werte und Prinzipien der Stakeholder*innen
 - Im Unterschied zum Interessenkonflikt ist Wertekonflikt individueller und langfristiger
- Mögliche Lösung:
 - Suche nach höheren Werten die die Parteien vereinen, aber meist schwer zu lösen
 - Gegenseitiges Verständnis und Anerkennung meist das Beste was erreicht werden kann

Beziehungskonflikt

- Ursachen:
 - Negative Erfahrungen in der Vergangenheit in vergleichbaren Situationen
- Mögliche Lösung:
 - Suche nach höheren Werten die die Parteien vereinen, aber meist schwer zu lösen
 - Meist hilft es das Problem auf höhere Autoritätsebene zu eskalieren und Auswechseln von Personen

Strukturkonflikt

- Ursachen:
 - Ungleichheit in Macht, Wettbewerb um Ressourcen oder strukturelle Abhängigkeit von Personen
- Mögliche Lösung:
 - Auch hier hilft es meist Probleme auf höhere Autoritätsebene zu eskalieren

Übung 1: Konflikttypen

- Siehe Übungsdokument „Schulung IREB FL_Tag 3_Übungen“



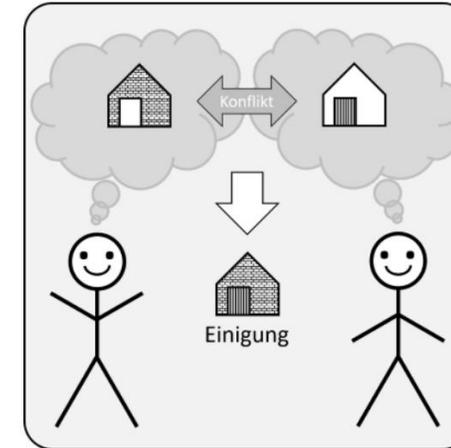
Konfliktlöstechniken (1)

- Einigung
- Kompromiss
- Abstimmung
- Ober-sticht-Unter
- Variantenbildung
- Unterstützende Techniken

Konfliktlöstechniken (2)

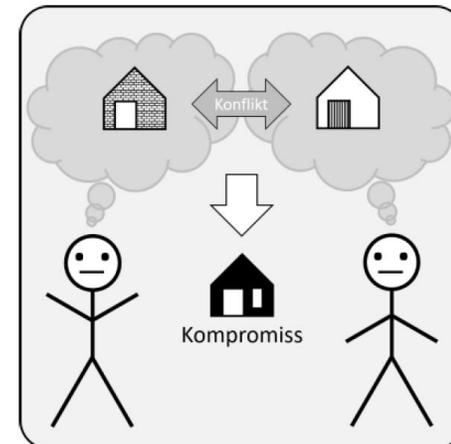
Einigung

- Ergebnis einer Diskussion bis alle Stakeholder*innen einer bevorzugten Option zustimmen
- Meist bei Datenkonflikten einsetzbar



Kompromiss

- Ergebnis ist eine Option die nicht bevorzugt ist, aber als vorteilhaft angesehen wird
- Meist bei Sachkonflikten, oft auch bei Interessen- oder Strukturkonflikten einsetzbar



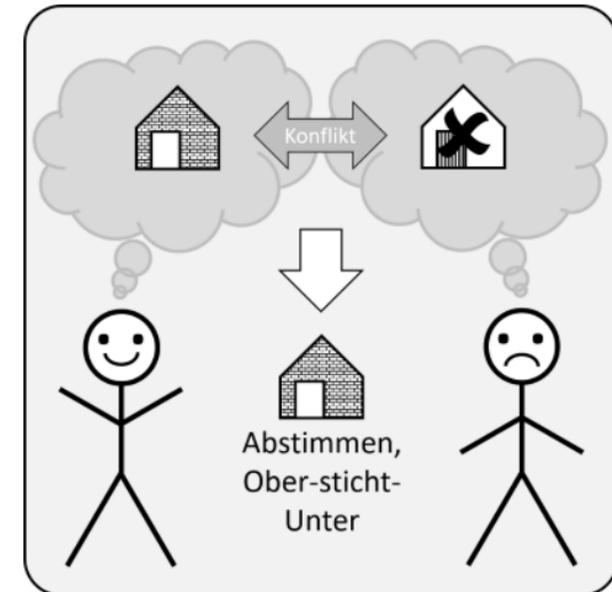
Konfliktlöstechniken (3)

Abstimmung

- Einfach, da Auswahl zwischen widersprüchlichen Anforderungen
- Durch anonyme Stimmabgabe bei Sach- und Interessenkonflikten

Ober-sticht-Unter

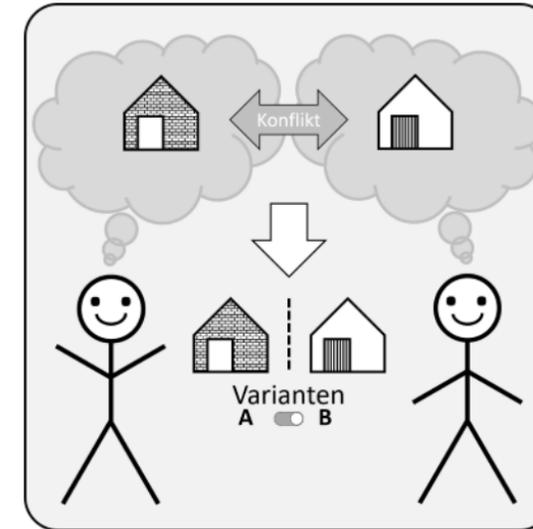
- Wenn keine Einigung durch Kompromiss erzielt wird, entscheidet übergeordnete Autorität
- Bei Interessens- und Strukturkonflikten geeignet



Konfliktlöstechniken (4)

Variantenbildung

- Separate Lösungen für alle widersprüchlichen Anforderungen
- Bei Sach-, Interessens- und Wertekonflikten
- Achtung:
 - Viel Zeit erforderlich und Komplexität wächst bei Entwicklung, Betrieb und Wartung



Konfliktlöstechniken (5)

Unterstützende Techniken zu vorangegangenen Techniken

- Consider-All-Facts (CAF) – Lösungssuche basierend auf Kriterien wie Kosten, Zeit, Risiken etc.
- Plus-Minus-Interesting (PMI) – Plus- und Minuspunkte einer Lösung werden gegenübergestellt
- Entscheidungsmatrix

Kriterium	Gewichtung	Alternative 1: nur iPhone		Alternative 2: Android und iPhone	
		Bewertung	Gewichtet	Bewertung	Gewichtet
Kundenbestand	2	3	6	4	8
Entwicklungskosten	1	3	3	2	2
Time-to-Market	3	4	12	2	6
Ansehen	2	2	4	4	8
Benutzererlebnis	1	5	5	3	3
Summe			30		27

Beispiel einer Entscheidungsmatrix

Übungen 2-6: CAF, PMI und Entscheidungsmatrix

- Siehe Übungsdokument „Schulung IREB FL_Tag 3_Übungen“

4.4 Validierung von Anforderungen

Validierung von Anforderungen

- Qualität von Anforderungen frühzeitig sicher stellen, um unnötigen Aufwand in Zukunft zu vermeiden
 - Sind die Bedürfnisse der Stakeholder*innen durch unsere Dokumentation abgedeckt?
 - Stimmen alle Stakeholder*innen mit den Anforderungen übereinstimmen? und
 - Wie wahrscheinlich sind unsere Annahmen über den Systemkontext?
- Validierung fügt dem Projekt Zeit und Kosten hinzu → Effizienz und Effektivität Anliegen des RE
- Fehler entdecken, überwachen und analysieren und nach Verbesserungsmöglichkeiten suchen
- Beispiele für Fehler: Mehrdeutigkeit, Unvollständigkeit, Widersprüche
- Validierung zur Absicherung des rechtlichen Risikos
 - Vertrag wird auf Basis der Anforderungen erstellt

Qualitätsaspekte von Anforderungen

- Inhalt einer Anforderung
 - Inhaltliche Fehler wenn Entwicklungsaktivitäten auf falschen Informationen basieren
 - Vollständigkeit aller Anforderungen und einzelner Anforderungen, Verfolgbarkeit, Korrektheit, Konsistenz, keine vorzeitigen Entwurfsentscheidungen, Überprüfbarkeit, Notwendigkeit
- Dokumentation einer Anforderung
 - Überprüfung auf Mängel in der Dokumentation oder Verstöße gegen Dokumentationsvorschriften
- Abgestimmtheit einer Anforderung
 - Überprüfung von Anforderungen auf Abgestimmtheit zwischen Stakeholdern

Wichtige Aspekte zur Validierung von Anforderungen (1)

- Die richtigen Stakeholder*innen einbeziehen
 - Geringes Maß an Unabhängigkeit (Stakeholder*innen aus Ermittlung) kostengünstig, aber ev. widersprüchlich, fehlerhafte Annahmen etc.
 - Hohes Maß an Unabhängigkeit (externe Prüfer*innen bzw. Auditor*innen) teurer, aber kann effektiver sein, wenn diese mehrere und schwerwiegendere Mängel finden sollen
- Trennung von Fehlererkennung und Fehlerkorrektur
 - Zuerst alle Anforderungen für die Validierung auswählen und prüfen und erst danach entscheiden, ob bestimmte festgestellte Mängel behoben werden sollen oder nicht
- Validierung aus unterschiedlichen Perspektiven

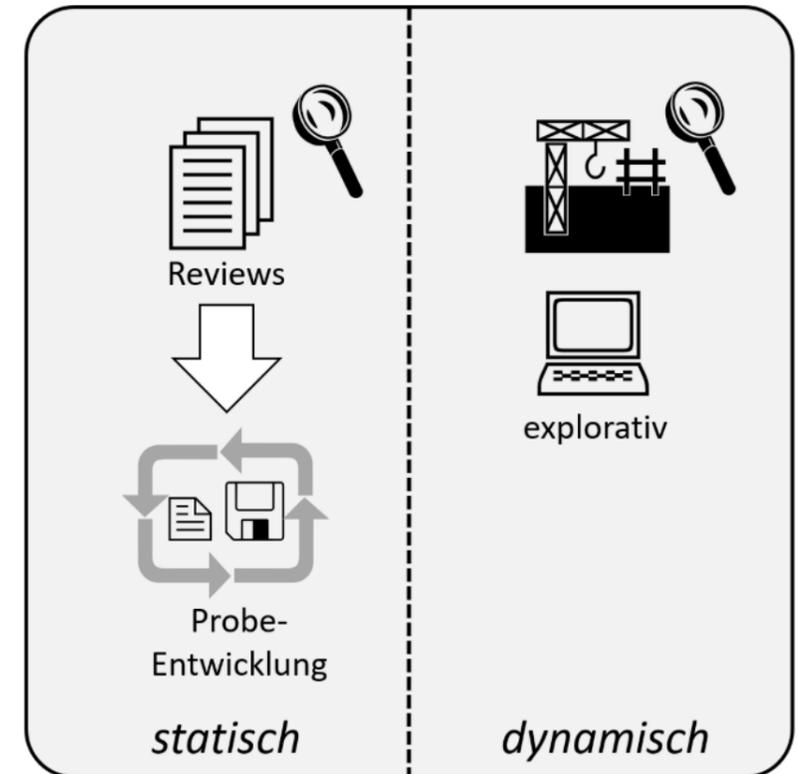
Wichtige Aspekte zur Validierung von Anforderungen (2)

- Korrekte Validierung ist immer eine Gruppenleistung
 - bei iterativen Projekten ist es das agile Team und ev. zusätzliche Gutachter*innen
 - bei sequenziellen Projekten ev. für jeden Validierungsvorgang spezielles Team zusammenstellen (Unternehmen, Anwender*innen, Entwickler*innen, Tester*innen, Betreiber*innen, Fachexpert*innen oder Spezialist*innen)
- Wiederholte Validierung
 - Auch bei sequenziellen Projekten nicht ausschließlich Anforderungen zu Beginn validieren
 - Rituale iterativer Projekte beinhalten viele Validierungsmaßnahmen wie bspw. Sprint-Planung, Sprint-Reviews etc. bieten Möglichkeit, einzelne Anforderungen zu validieren und zu verbessern - jedoch soll auch das gesamte Produkt-Backlog zu Beginn eines Projekts / Inkrements validiert werden



Validierungstechniken (1)

- Meist mehrere Validierungstechniken in verschiedenen Projektphasen eingesetzt
- 3 Kategorien von Validierungstechniken:
 - Review-Techniken
 - Explorationstechniken
 - Probe-Entwicklung



Validierungstechniken (2)

Reviews

- Informelle Reviews
 - Autor-Reviewer-Zyklus, bei dem der/ die Autor*in das Arbeitsprodukt an die Gruppe sendet, danach Reviews erhält und diese einarbeitet
 - Günstige Technik, aber meist nur in frühen Entwicklungsphasen eingesetzt
- Formale Reviews
 - Walkthroughs
 - Autor erklärt Arbeitsprodukt dem Publikum in einer interaktiven Sitzung und das Publikum kann Kommentare abgeben, Mängel aufzeigen und Alternativen vorschlagen
 - Inspektionen
 - Meist in Form eines Meetings mit Moderator*in, Autor*in und Inspektor*in/ Moderator durchgeführt
 - Inspektor *innen prüfen Arbeitsprodukt auf Basis deren Fachwissen und Autor*in erklärt ev. Unklarheiten

Exkurs: Inspektion

- Formellste Review-Technik
- Inspektor*innen prüfen Arbeitsprodukt auf Basis deren Fachwissen
- Prüfen des Arbeitsprodukts auf Übereinstimmung mit geltenden Standards, Normen und Vorschriften
- Autor*in: Erläutert den Inspektor *innen die Anforderungen (Übersichtsphase) und ist später für Behebung der Fehler zuständig
- Rollen: Organisator, Moderator, Autor, Vorleser, Inspektoren, Protokollant



Validierungstechniken (3)

Probeentwicklung

- Anforderungen werden als Input für Entwickler*innen bereit gestellt und diese versuchen Zwischenarbeitsprodukte (z.B. Designs, Code, Testfälle, Handbücher) zu erstellen
- Währenddessen können Entwickler*innen Fehler wie z.B. Unklarheiten, Auslassungen und Inkonsistenzen entdecken
- Quantität und Schwere festgestellter Mängel deutet auf Qualität der Anforderungen hin – ist diese nicht ausreichend, sind weitere Validierungen nötig, wie bspw. Alpha- Beta-Tests etc.
- Ähnliche Validierung kann vom RE selbst durch die Erstellung von Anforderungsmodellen durchgeführt werden

Validierungstechniken (4)

Explorationstechniken

- Prototyping
 - Hierbei wird eine frühe Systemversion einer Gruppe von Stakeholder*innen zur Evaluierung übergeben
 - Ein Prototyp sieht von außen wie das beabsichtigte System aus und ermöglicht Stakeholder*innen praktische Erfahrungen zu sammeln – intern ist der Prototyp aber meist noch unvollständig
 - Prototyping und Storyboarding können auch als Ermittlungstechniken eingesetzt werden
- Alpha- und Beta-Tests
 - Endbenutzer*innen wird eine voll funktionsfähige Vorproduktionsversion übergeben
 - Bei Alpha-Tests wird die Systeminteraktion der Benutzer*innen in simulierter Umgebung beobachtet, z.B. im Usability-Labor
 - Bei Beta-Tests sollen Benutzer*innen das Aussehen und Verhalten des Systems in realer Umgebung validieren
- A/B-Tests
 - System wird Benutzer*innengruppen in zwei Varianten mit unterschiedlichem Design und/ oder Funktion übergeben und die Reaktion beider Gruppen wird gemessen und verglichen
- Minimum Viable Product (MVP)
 - Sammlung von Benutzerfeedback zu einem frühen Entwicklungszeitpunkt; in agiler Entwicklung häufig eingesetzt

Wissenswiederholung

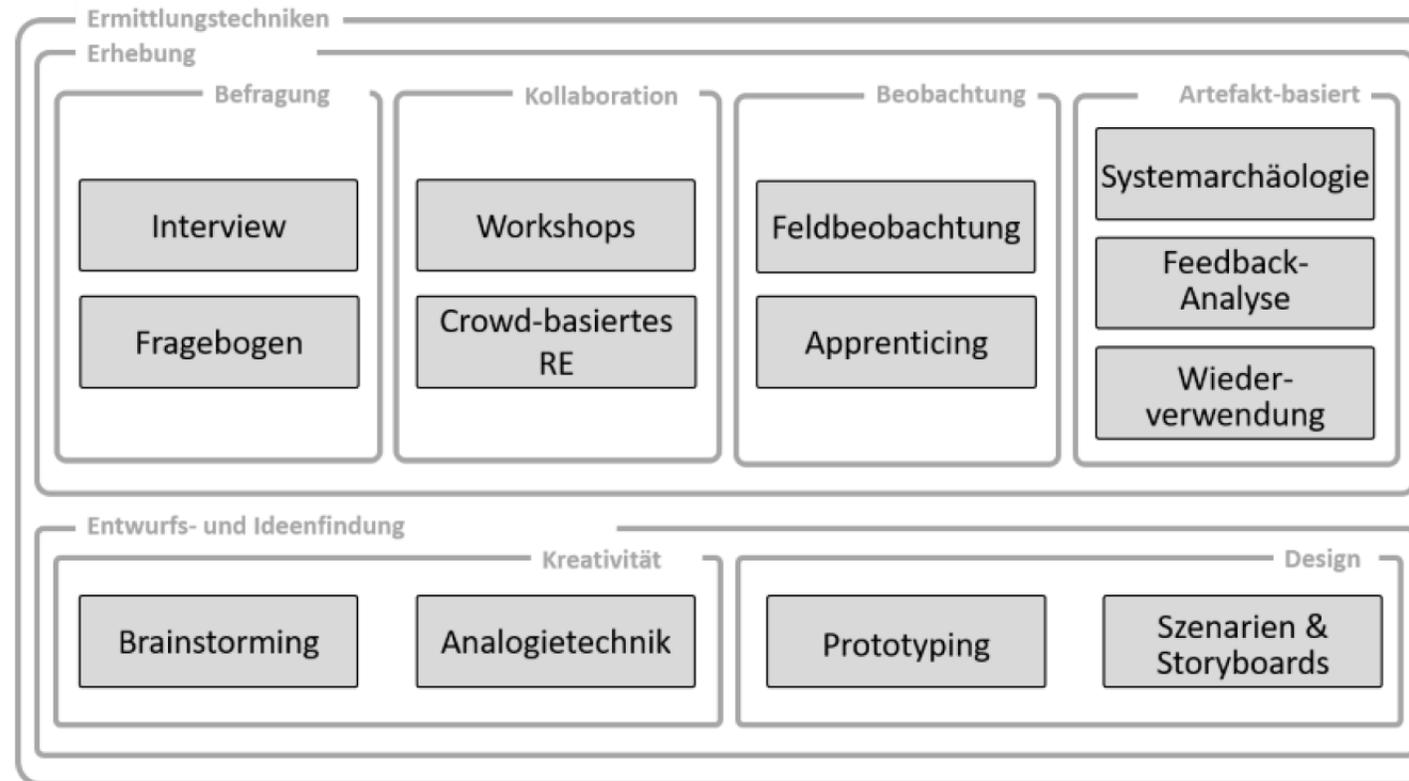
LE 4.3 & 4.4

Wissenswiederholung (1)

- Welche Techniken der Anforderungsermittlung kennen Sie?

! Wissenswiederholung (1)

- Welche Techniken der Anforderungsermittlung kennen Sie?



Wissenswiederholung (2)

- Wie lauten die 4 Schritte zur Konfliktlösung im RE?

Wissenswiederholung (2)

- Wie lauten die 4 Schritte zur Konfliktlösung im RE?
 1. Konfliktidentifizierung
 2. Konfliktanalyse
 3. Konfliktlösung
 4. Dokumentation der Konfliktlösung

Wissenswiederholung (3)

- Welche Konflikttypen kennen Sie?

Wissenswiederholung (3)

- Welche Konflikttypen kennen Sie?
- Sachkonflikt
- Datenkonflikt
- Interessenkonflikt
- Wertekonflikt
- Beziehungskonflikt
- Strukturkonflikt

Wissenswiederholung (4)

- Welche Konfliktlösungstechniken kennen Sie?

Wissenswiederholung (4)

- Welche Konfliktlöstechniken kennen Sie?
 - Einigung
 - Kompromiss
 - Abstimmung
 - Ober-sticht-Unter
 - Variantenbildung
 - Unterstützende Techniken

Wissenswiederholung (5)

- Was sind wichtige Aspekte zur Validierung von Anforderungen?

Wissenswiederholung (5)

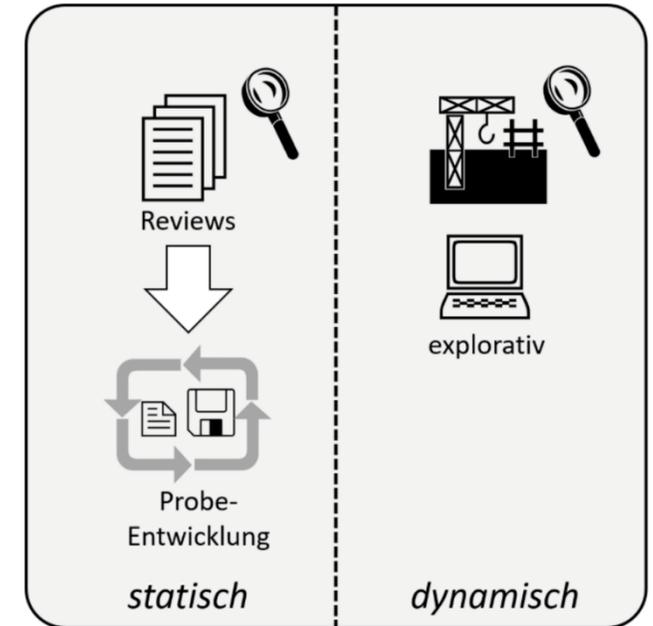
- Was sind wichtige Aspekte zur Validierung von Anforderungen?
 - Die richtigen Stakeholder*innen einbeziehen
 - Trennung Fehlererkennung und Fehlerkorrektur
 - Validierung aus unterschiedlichen Perspektiven
 - Korrekte Validierung ist immer eine Gruppenleistung
 - Wiederholte Validierung zur Qualitätssicherung

Wissenswiederholung (6)

- Welche Validierungstechniken kennen Sie?

! Wissenswiederholung (6)

- Welche Validierungstechniken kennen Sie?
 - Reviews
 - Informelle Reviews
 - Formale Reviews
 - Walkthroughs
 - Inspektionen
 - Probeentwicklung
 - Explorationstechniken
 - Prototyping
 - Alpha- und Beta-Tests
 - A/B-Tests
 - MVP



5 Prozess und Arbeitsstruktur

Der Requirements Engineering Prozess

- Prozesse dienen der systematischen Arbeit, um die Arbeitsweise und die Erzeugung von Arbeitsprodukten zu gestalten und zu strukturieren
- Ein Requirements Engineering (RE)-Prozess organisiert, wie RE-Aufgaben unter Verwendung geeigneter **Praktiken** durchgeführt und die erforderlichen **Arbeitsprodukte** erzeugt werden
- Requirements Engineers müssen immer einen **maßgeschneiderten** RE-Prozess konfigurieren
- Der RE-Prozess gestaltet den **Informationsfluss** und das **Kommunikationsmodell** zwischen den RE Beteiligten wie Kund*innen, Benutzer*innen, Requirements Engineers, Entwickler und Tester und definiert die zu erstellenden **RE-Arbeitsprodukte**
- Ein guter RE-Prozess bietet den **Rahmen**, in dem Requirements Engineers Anforderungen ermitteln, dokumentieren, abstimmen und verwalten

5.1 Einflussfaktoren



Wichtige Faktoren bei der Auswahl von Requirements Engineering Prozessen

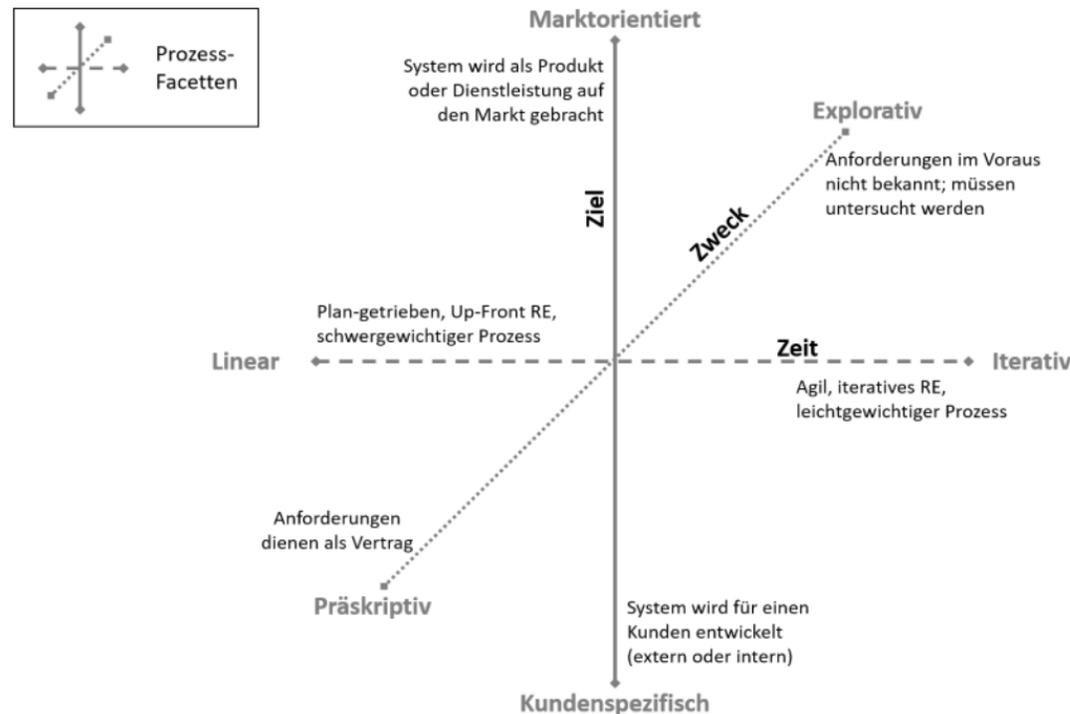
- Eignung im Gesamtprozess
- Entwicklungskontext
 - Kund*innen en-Lieferant *innen-Benutzer *innen-Beziehung (Auftraggeber-Auftragnehmer-Beziehung)
 - Entwicklungstyp
 - Vertrag
 - Vertrauen
- Verfügbarkeit und Fähigkeit der Stakeholder*innen
- Gemeinsames Verständnis
- Komplexität und Kritikalität
- Randbedingungen
- Verfügbare Zeit und Budget
- Volatilität von Anforderungen
- Erfahrung der Requirements Engineers

5.2 Facetten von RE Prozessen



Facetten und Ausprägungen von RE-Prozessen (1)

- Aufwandsverschwendung für jedes RE-Vorhaben den RE-Prozess von Grund auf neu zu definieren
- Daher RE-Prozess aus bereits vorhandenen Elementen konfigurieren
- Folgende Grafik zeigt die 3 Facetten mit jeweils 2 Ausprägungen
- Für jede Ausprägung einer Facette gibt es Kriterien für ihre Auswahl - nicht alle Kriterien müssen erfüllt sein, um eine Ausprägung einer Facette auszuwählen



Facetten und Ausprägungen von RE-Prozessen (2)

1. Zeitfacette: Linear versus Iterativ

Linearer Prozess

- Im linearen RE Prozess werden Anforderungen vorab in einer eigenen Prozessphase spezifiziert
- Es wird vorab eine umfassende Anforderungsspezifikation erstellt die kaum Anpassungen während der System-Konzeption und –Implementierung erfordert
- Kriterien für die Wahl eines linearen Prozesses:
 - Entwicklungsprozess ist planbasiert und weitgehend linear
 - Stakeholder*innen sind verfügbar, kennen ihre Anforderungen und können sie spezifizieren
 - Anforderungsspezifikation ist als vertragliche Grundlage der Systemrealisierung erforderlich
 - Regulierungsbehörden verlangen frühzeitig eine umfassende Anforderungsspezifikation

Facetten und Ausprägungen von RE-Prozessen (3)

1. Zeitfacette: Linear versus Iterativ

Iterativer Prozess

- Im iterativen RE Prozess werden Anforderungen schrittweise spezifiziert
- Die Idee besteht darin, die Spezifikation der Anforderungen mit der Konzeption und der Implementierung des Systems zu verknüpfen
- Kriterien für die Wahl eines iterativen Prozesses:
 - Entwicklungsprozess für das System ist iterativ und agil
 - Viele Anforderungen sind nicht im Voraus bekannt, sondern werden erst im Laufe der Entwicklung des Systems entstehen und sich entwickeln
 - Stakeholder*innen stehen für kurze Rückkopplungsschleifen zur Verfügung
 - Dauer der Entwicklung lässt mehr als nur ein oder zwei Iterationen zu

Facetten und Ausprägungen von RE-Prozessen (4)

2. Zweckfacette: Präskriptiv versus Explorativ

Präskriptiver Prozess

- Beim präskriptiven RE-Prozess stellt die Anforderungsspezifikation einen Vertrag dar
- Alle Anforderungen sind verbindlich und müssen umgesetzt werden
- Idee ist eine Anforderungsspezifikation mit wenig Interaktion zw. Stakeholder *innen und Entwickler*innen
- Kriterien für die Wahl eines präskriptiven Prozesses:
 - Kund*in benötigt einen Vertrag für die Systementwicklung mit fixierter Funktionalität, fixiertem Umfang, Preis und Endtermin
 - Funktionalität und Umfang haben Vorrang vor Kosten und Terminen
 - Entwicklung des spezifizierten Systems kann ausgeschrieben oder ausgelagert werden

Facetten und Ausprägungen von RE-Prozessen (5)

2. Zweckfacette: Präskriptiv versus Explorativ

Explorativer Prozess

- Im explorativen RE-Prozess sind nur Ziele im Vorfeld bekannt, während konkrete Anforderungen ermittelt werden müssen
- Anforderungen sind häufig nicht a priori bekannt sind, sondern müssen erst ermittelt werden
- Kriterien für die Wahl eines explorativen Prozesses:
 - Stakeholder*innen haben zunächst nur eine vage Vorstellung ihrer Anforderungen
 - Stakeholder*innen sind stark involviert und liefern kontinuierliches Feedback
 - Termine und Kosten haben Vorrang vor Funktionalität und Umfang
 - Kund*in ist mit einem Rahmenvertrag über Ziele, Ressourcen und den zu zahlenden Preis für einen bestimmten Zeitraum oder eine bestimmte Anzahl von Iterationen einverstanden
 - Umsetzung der Anforderungen ist noch nicht konkret definiert

Facetten und Ausprägungen von RE-Prozessen (6)

3. Zielfacette: Kund*innenspezifisch versus Marktorientiert

Kund*innenspezifisch

- Im kund*innenspezifischen RE-Prozess wird das System vom Kunden individuell bestellt und von Lieferant*innen entwickelt
- Lieferant*in und Kund*in können hierbei Teil der gleichen Organisation sein
- Kriterien für die Wahl eines kund*innenspezifischen Prozesses:
 - System wird hauptsächlich von derjenigen Organisation genutzt, die das System bestellt hat und für seine Entwicklung bezahlt
 - Die wichtigen Stakeholder*innen sind hauptsächlich mit der Organisation der Kund*innen verbunden
 - Kund*in wünscht eine Anforderungsspezifikation, die als Vertrag dienen kann

Facetten und Ausprägungen von RE-Prozessen (7)

3. Zielfacette: Kundenspezifisch versus Marktorientiert

Marktorientiert

- Im marktorientierten RE-Prozess wird das System als Produkt oder Service für einen Markt entwickelt, ausgerichtet auf bestimmte Nutzer*innensegmente
- Idee ist, dass die Organisation, die das System entwickelt, auch den RE-Prozess steuert
- Kriterien für die Wahl eines marktorientierten Prozesses:
 - Die entwickelnde Organisation oder einer ihrer Kund*innen beabsichtigt, das System in einem bestimmten Marktsegment als Produkt oder Dienstleistung zu verkaufen
 - Künftige Benutzer*innen sind nicht eindeutig identifizierbar
 - Der/ die RE muss die Systemanforderungen an den Bedürfnissen der Zielnutzer*innen orientieren
 - Produktmanager*in; Marketingpersonal und ggf Systemarchitekten sind primäre Stakeholder

Facetten und Ausprägungen von RE-Prozessen (8)

Hinweise und Warnungen

- Zuvor genannte Kriterien zur Prozessauswahl sind Heuristiken, keine Regeln
- Einige miteinzubeziehende Faktoren hierzu:
 - Lineare RE-Prozesse funktionieren nur, wenn ein durchdachtes Verfahren für sich **ändernde Anforderungen** vorhanden ist
 - Lineare RE-Prozesse implizieren lange **Rückkopplungsschleifen** – vom Anforderung-Schreiben bis zur Implementierung können Monate oder sogar Jahre vergehen
 - In einem marktorientierten Prozess ist das **Feedback potenzieller Benutzer*innen** das einzige Mittel, um zu validieren
 - In einem agilen Umfeld passt ein **iterativer** und **explorativer** RE-Prozess am besten. Iterationen haben eine feste Dauer (in der Regel 2-6 Wochen)

Facetten und Ausprägungen von RE-Prozessen (9)

Gegenseitige Beeinflussung

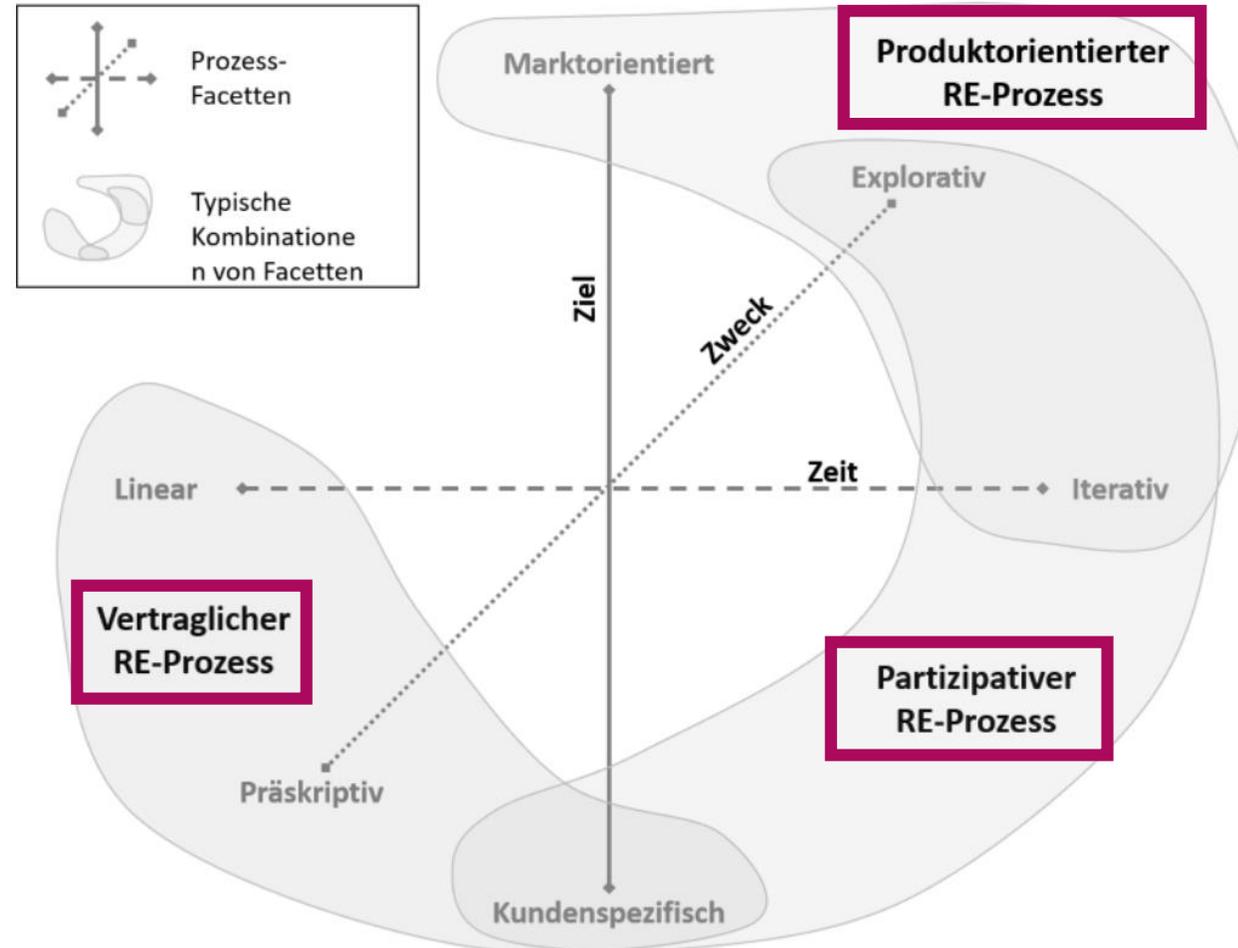
- Wahl einer Facette hat oft Einfluss auf die Auswahl in einer weiteren Facette
- Einige Beispiele hierzu:
 - Linear und präskriptiv werden häufig zusammen gewählt, d.h. wenn Requirements Engineers sich für einen linearen RE-Prozess entscheiden, entscheiden sie sich in der Regel für einen Prozess, der sowohl linear als auch präskriptiv ist
 - Explorative RE-Prozesse sind typischerweise auch iterativ (und umgekehrt)
 - Ein marktorientierter RE-Prozess lässt sich kaum mit linearen & präskriptiven Prozessen kombinieren

5.3 Konfigurieren eines RE Prozesses



Typische Kombinationen von Facetten (1)

3 typische RE-Prozesskonfigurationen und ihre Beziehung zu den 3 Facetten



Typische Kombinationen von Facetten (2)

Partizipativer RE-Prozess: iterativ & explorativ & kundenspezifisch

- Hauptanwendungsfall: Lieferant*innen & Kund*innen arbeiten eng zusammen; Stakeholder sind im RE- und im Entwicklungsprozess eingebunden
- Typische Arbeitsprodukte: Produkt-Backlog mit User Storys und/oder Aufgabenbeschreibungen, Prototypen
- Typischer Informationsfluss: Kontinuierliche Interaktion zw. Stakeholder*innen, Product Owner*innen, Requirements Engineers & Entwickler*innen

Vertraglicher RE-Prozess: linear & präskriptiv & kundenspezifisch

- Hauptanwendungsfall: Die Anforderungsspezifikation stellt die vertragliche Grundlage für die Systementwicklung durch Personen dar, die nicht an der Spezifikation beteiligt sind, und mit nur wenig Interaktion mit den Stakeholder*innen nach der Anforderungsphase.
- Typische Arbeitsprodukte: Klassische System-Anforderungsspezifikation (Lasten-/Pflichtenheft), bestehend aus natürlichsprachigen Anforderungen und Modellen
- Typischer Informationsfluss: Hauptsächlich von den Stakeholder*innen zu den Requirements Engineers

Produktorientierter RE-Prozess: Iterativ & explorativ & marktorientiert

- Hauptanwendungsfall: Eine Organisation spezifiziert und entwickelt Software, um sie als Produkt oder Service zu verkaufen oder zu vertreiben
- Typische Arbeitsprodukte: Produkt-Backlog, Prototypen
- Typischer Informationsfluss: Interaktion zwischen Product Owner*innen, Marketing, Requirements Engineers, Digital Designer*innen, Entwickler*innen und (vielleicht) schnellem Feedback von Kund*innen/Benutzer*innen

Vorgehen zur Konfiguration des RE-Prozesses

1. Analyse Einflussfaktoren
2. Beurteilen der Facettenkriterien
 - Liste der Auswahlkriterien für Facetten durchgehen und bewerten (-,--,0,+,++)
3. Konfigurieren d. Prozesses
 - Idealerweise eine der 3 genannten Konfigurationen eindeutig erkennbar, sonst Mischformen erstellen
4. Arbeitsprodukte bestimmen
5. Geeignete Praktiken auswählen

Wissenswiederholung

LE 5.1 - 5.3

Wissenswiederholung (1)

- Was sind wichtige Faktoren bei der Auswahl von RE Prozessen?

Wissenswiederholung (1)

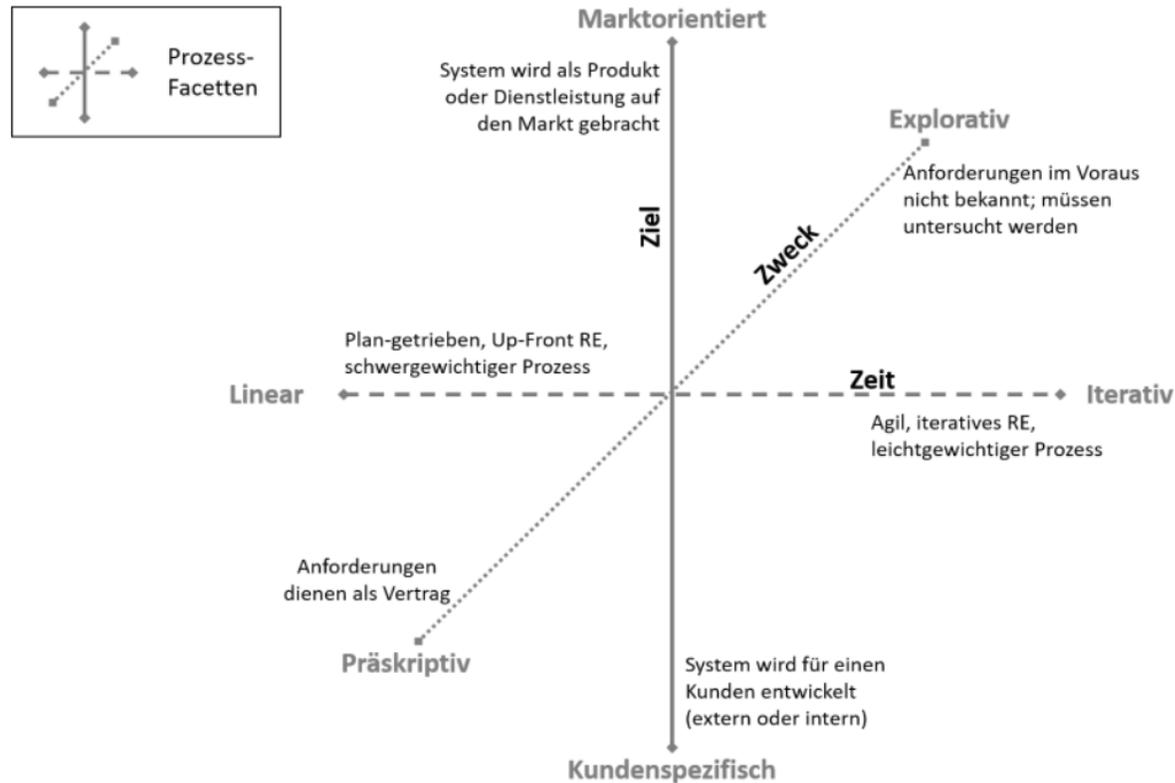
- Was sind wichtige Faktoren bei der Auswahl von RE Prozessen?
 - Eignung im Gesamtprozess
 - Entwicklungskontext
 - Verfügbarkeit und Fähigkeit der Stakeholder*innen
 - Gemeinsames Verständnis
 - Komplexität und Kritikalität
 - Randbedingungen
 - Verfügbare Zeit und Budget
 - Volatilität von Anforderungen
 - Erfahrung der Requirements Engineers

Wissenswiederholung (2)

- Skizzieren Sie die 3 Facetten mit den jeweils 2 Ausprägungen und die zugehörigen Auswahlkriterien von RE Prozessen.

! Wissenswiederholung (2)

- Skizzieren Sie die 3 Facetten mit den jeweils 2 Ausprägungen und die zugehörigen Auswahlkriterien von RE Prozessen.



Wissenswiederholung (3)

- Was ist jeweils der **Hauptunterschied** zwischen...
 - einem **linearen** und einem **iterativen Prozess**?
 - einem **präskriptiven** und einem **explorativen Prozess**?
 - einem **kundenspezifischen** und einem **marktorientierten Prozess**?

Wissenswiederholung (3)

- Was ist jeweils der **Hauptunterschied zwischen...**
 - einem **linearen und einem iterativen Prozess?**
 - Linear: Anforderungen vorab in einer eigenen Prozessphase spezifiziert
 - Iterativ: Anforderungen werden schrittweise spezifiziert
 - einem **präskriptiven und einem explorativen Prozess?**
 - Präskriptiv: Anforderungen stellen Vertragsinhalt dar und müssen umgesetzt werden
 - Explorativ: hier sind nur Ziele im Vorfeld bekannt - konkrete Anforderungen müssen ermittelt werden
 - einem **kundenspezifischen und einem marktorientierten Prozess?**
 - Kund*innenspezifisch: System wird von Kund*innen individuell bestellt und für sie individuell entwickelt
 - Marktorientiert: System wird für einen Markt entwickelt und auf Nutzer*innensegmente ausgerichtet

6 Praktiken für das Requirements Management

6.1 Was ist Requirements Management?



Was ist Requirements Management?

Definition Requirements Management

- Die Norm ISO/IEC/IEEE 29148:2018 [ISO29148] definiert Requirements Management als:
„Aktivitäten, die Anforderungen während des gesamten Lebenszyklus eines Systems, Produkts oder einer Dienstleistung identifizieren, dokumentieren, aufrechterhalten, kommunizieren, nachverfolgen und verfolgen.“
- Im CPRE Glossar [Glin2020], wird Requirements Management definiert als *„Der Prozess der Verwaltung bestehender Anforderungen und anforderungsbezogener Arbeitsprodukte, einschließlich der Speicherung, Änderung und Verfolgung von Anforderungen.“*
- Requirements Management soll bereits zu Projektbeginn durchgeführt werden um die Ressourcen und Prozesse für das RM mit Blick auf die am Ende zu erwartenden Anforderungen einzurichten



Ebenen des Requirements Management

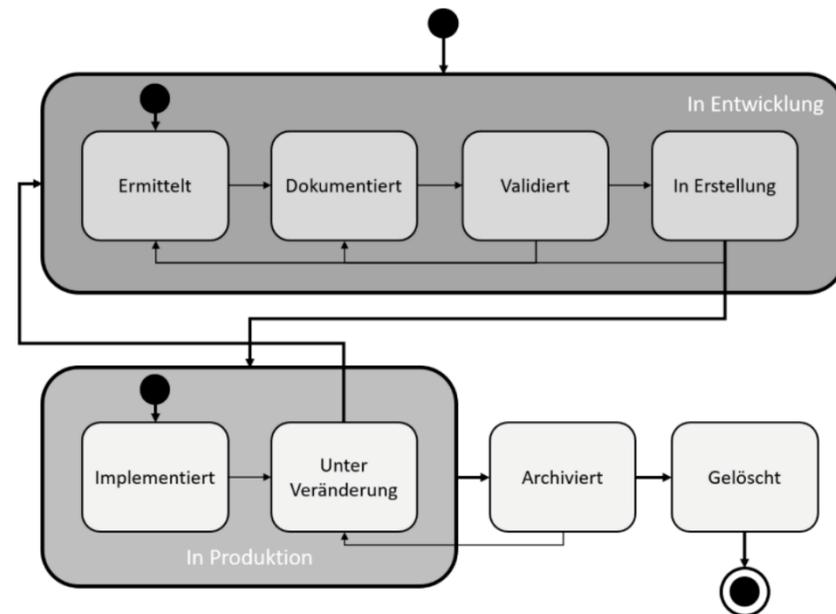
- Requirements Management kann auf verschiedenen Ebenen erfolgen:
 - den einzelnen Anforderungen (z.B. einzelne User Story in einem agilen Projekt)
 - den Arbeitsprodukten, die eine bis mehrere Anforderungen (z.B. Use Case Diagramm) enthalten können
 - dem System (z.B. gesamtes Anforderungsspezifikationsdokument einer Software) hinsichtlich seiner Arbeitsprodukte und der darin enthaltenen Anforderungen

6.2 Verwaltung des Lebenszyklus

Verwaltung des Lebenszyklus (1)

Management des Lebenszyklus von Anforderungen und Arbeitsprodukten

- Anforderungen werden erstellt, ausgearbeitet, validiert, konsolidiert, implementiert, verwendet, geändert, gepflegt, überarbeitet, umgestaltet, zurückgezogen, archiviert und/oder gelöscht
- Während ihrer Lebensdauer kann eine Anforderung eine begrenzte Anzahl von Zuständen und Zustandsübergängen annehmen, die auf Ereignissen beruhen
- Folgendes Modell zeigt ein Zustandsdiagramm für den Lebenszyklus einer einzelnen Anforderung



Verwaltung des Lebenszyklus (2)

Aktives Betreiben des Management Lebenszyklus

- Definieren von Lebenszyklusmodellen für Arbeitsprodukte und die enthaltenen Anforderungen mit
 - Den **Zuständen**, die ein Arbeitsprodukt oder eine Anforderung annehmen kann
 - Den erlaubten **Übergängen** zwischen diesen Zuständen
 - Den **Ereignissen**, die den Übergang von einem Zustand in einen anderen auslösen
- Sicherstellen, dass nur explizit erlaubte Übergänge stattfinden
- Aufzeichnen der tatsächlichen Zustände, die die Arbeitsprodukte und Anforderungen einnehmen
- Aufzeichnen der tatsächlich auftretenden Übergänge
- Berichterstaten über diese Zustände und Übergänge

6.3 Versionskontrolle



Versionskontrolle (1)

2 Gründe für die Versionskontrolle von Arbeitsprodukten

1. Wenn Änderungen schief gehen oder deren Nutzen nicht erreicht wird, kann man eine neue Änderung einführen oder zu einer früheren Version zurückgehen und dort fortfahren
2. Wissen über die Historie des Arbeitsprodukts kann helfen um über zukünftige Änderungen zu entscheiden



Versionskontrolle (2)

Versionsstrategie besteht mindestens aus:

1. Versionsnummer: Identifizierung jeder Version eines Arbeitsprodukts durch Versionsnummer & -datum
2. Änderungshistorie: Klare Änderungsbeschreibung - eindeutig mit der Versionsnummer verknüpft
3. Strikte Regelung für die Speicherung von Versionen, um alte Versionen abzurufen

Versionsnummern bestehen aus zumindest folgenden beiden Teilen:

- Version – beginnt bei Null, solange sich Arbeitsprodukt in Entstehung befindet. Wenn es formell genehmigt oder freigegeben ist, wird ihm die Version Eins zugewiesen – danach wird Version nur für größere, substantielle Aktualisierungen erhöht
- Inkrement – beginnt meist bei eins oder null und wird mit jeder (äußerlich sichtbaren) Änderung, sei es inhaltlich oder oft nur textuell oder redaktionell, erhöht

Versionskontrolle – Änderungshistorie

1.1. Änderungshistorie

Version	Datum	Änderungen	Autor
0.1	03.08.2021	Interne Arbeitsversion	E. Wölfel
0.2	17.08.2021	Anforderungserhebung	E. Wölfel
0.3	04.10.2021	Technologiepotenzial	E. Wölfel
0.4	16.11.2021	Technologiepotenzial	E. Wölfel, A.

Version	Datum	Änderungen	Autor	Version	Datum	Kommentar
0.5	07.12.2021	Konzept & Prototyp		1.0	2007	Erste Version
0.6	11.2.2022	Evaluierung		2.1	November 2010	Erstes großes Update
0.7	21.3.2022	Evaluierung & Ausblick		2.2	1. März 2015	Kleines Update (Inkonsistenzen beseitigt, Rechtschreib- und Grammatikfehler behoben)
1.0	22.04.2022	Finale Version		3.0.0	1. Oktober 2020	Zweites großes Update, das den Stand des heutigen RE widerspiegelt und sowohl planbasierte als auch agile Ansätze zur Spezifizierung und Verwaltung von Anforderungen umfasst.
				3.0.1	7. Oktober 2020	Rechtschreibfehler behoben und LE 4.3 umbenannt.

IREB Certified Professional for Requirements Engineering

6.4 Konfigurationen und Basislinien

Konfigurationen und Basislinien (1)

Was ist eine Konfiguration?

- Für einen Requirements Engineer ist eine Konfiguration ein konsistenter Satz von logisch zusammenhängenden Arbeitsprodukten, die Anforderungen enthalten
- Dieser Satz wird mit dem Zweck ausgewählt, um deutlich zu machen, welche Anforderungen in einer bestimmten Situation gültig sind oder waren



Konfigurationen und Basislinien (2)

Eigenschaften von Konfigurationen

- Logisch Verbunden - Anforderungskatalog gehört im Hinblick auf ein bestimmtes Ziel zusammen
- Konsistent - Anforderungskatalog hat keine internen Konflikte und ist in ein System integrierbar
- Eindeutig - sowohl Konfiguration als auch zugehörige Anforderungen sind eindeutig gekennzeichnet
- Unveränderlich - Konfiguration setzt sich aus ausgewählten Anforderungen mit jeweils einer bestimmten Version zusammen, die in dieser Konfiguration nie geändert wird
- Grundlage für das Zurücksetzen - Konfiguration ermöglicht ein Zurücksetzen auf eine frühere Konfiguration, wenn unerwünschte Änderungen aufgetreten sind

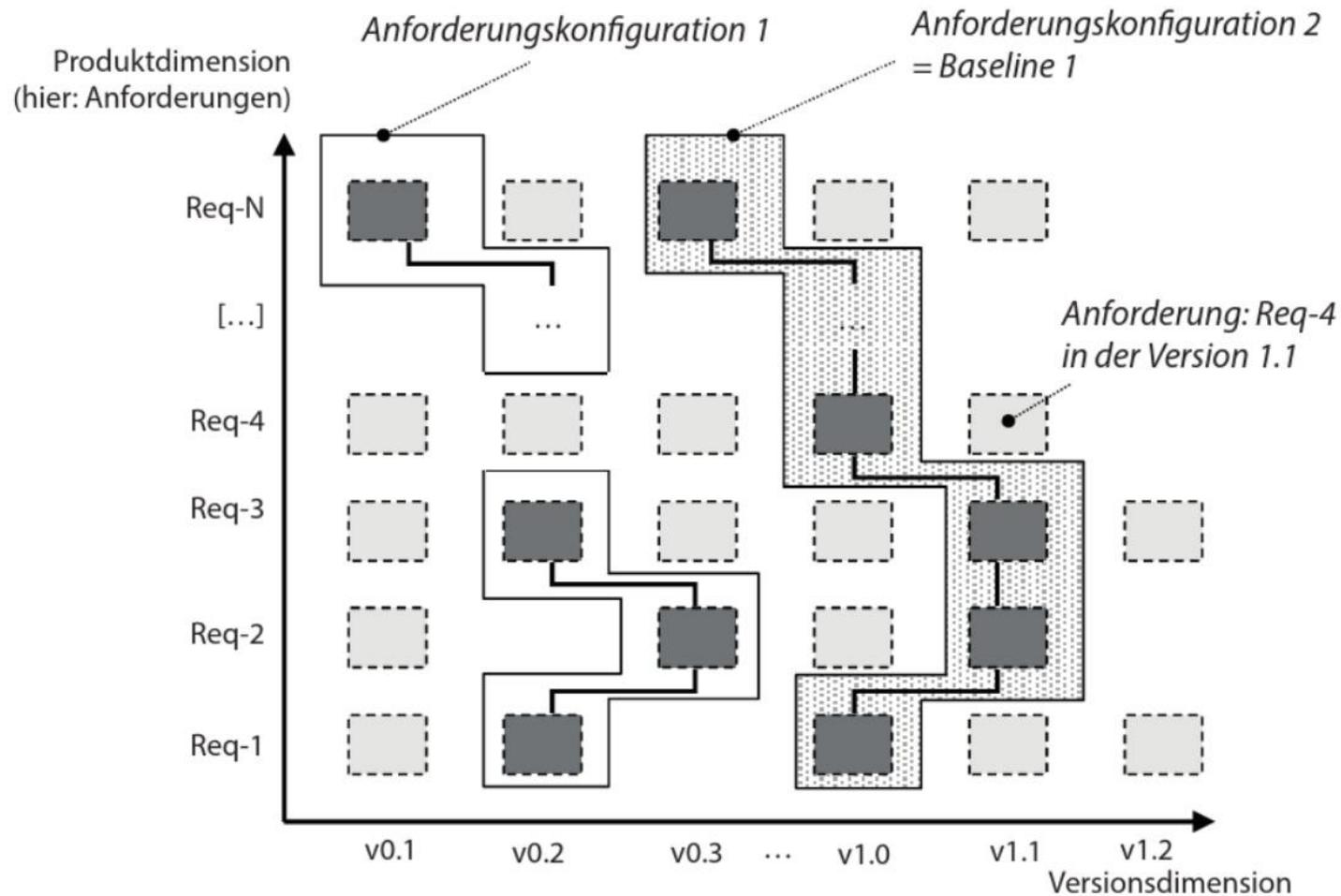


Konfigurationen und Basislinien (3)

2 Dimensionen von Konfigurationen

1. Produktdimension – gibt an, **welche Anforderungen** in dieser spezifischen Konfiguration enthalten sind. In der Regel enthält sie eine bestimmte Anforderungsauswahl, wie bspw. z.B. alle Anforderungen der französischen Systemversion, wohingegen die britische Systemversion eine andere Konfiguration haben kann
2. Versions Dimension - In einer bestimmten Konfiguration ist jede ausgewählte Anforderung in **genau einer Version** vorhanden. Es kann die neueste Version oder eine frühere Version sein, je nach dem Zweck der Konfiguration selbst – Bspw. ein neues Softwarerelease mit einigen Anforderungen in einer höheren Version, das dann eine andere Konfiguration hat

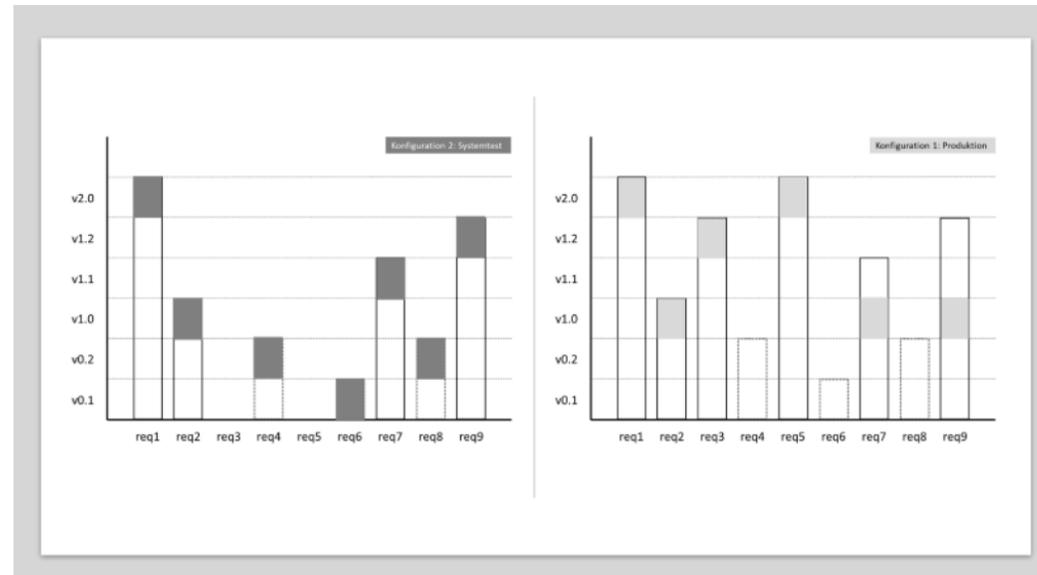
Konfiguration und Basislinien (Beispiel)



Konfigurationen und Basislinien (4)

Beispiel für Konfigurationen

- Die Abbildung zeigt ein Beispiel verschiedener Konfigurationen eines bestimmten Systems mit neun Anforderungen - einige befinden sich noch im Anfangsstadium der Entstehung, wie bspw. Anforderung 6 mit der Version v0.1, andere Anforderungen wurden bereits fertig gestellt und haben bereits ein größeres Update erfahren
- Das rechte Bild zeigt die Konfiguration, die sich derzeit in Produktion / Entstehung befindet
- Das linke Bild zeigt die Konfiguration, die gleichzeitig in der Systemtestumgebung vorhanden ist





Konfigurationen und Basislinien (5)

Basislinie / Baseline

- Eine Baseline ist eine **stabile, validierte** und **änderungskontrollierte** Konfiguration, die einen Meilenstein oder eine andere Art Haltepunkt im Projekt markiert
- Beispiel Baseline: Konfiguration am Ende der Entwurfsphase, kurz vor Beginn der Implementierungsphase, oder Konfiguration, die beim Go-Live eines bestimmten Releases gültig ist
- Sprint-Backlog dient in agilen Projekten als Baseline zu Start der nächsten Iteration – sollte im Projekt etwas schief laufen, kann das Team einen Rollback auf die Baseline durchführen und von dort aus neu starten

6.5 Attribute und Sichten



Attribute und Sichten (1)

Attribute von Anforderungen

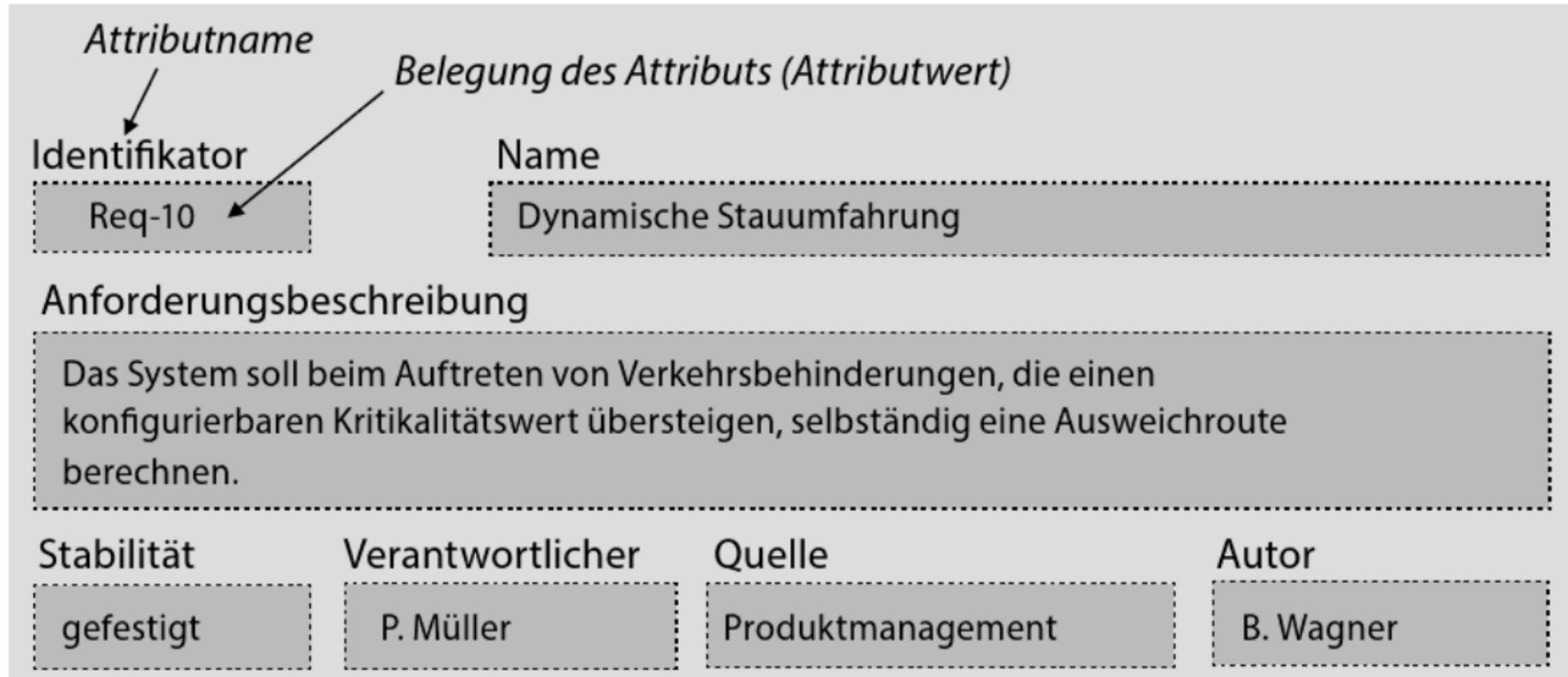
Festlegung der Attribute hängt vom Informationsbedarf der Stakeholder im Projekt ab.

Beispiele (siehe auch ISO29148):

- Identifizierung – eindeutige Nummer, einen Namen etc. für jede Anforderung
- Priorität für den Stakeholder - die (vereinbarte) Priorität der Anforderung aus der Sicht der Stakeholder.
- Abhängigkeit - manchmal gibt es eine Abhängigkeit zwischen den Anforderungen
- Risiko - Potential, dass die Umsetzung der Anforderung zu Problemen führt, wie z.B. Schäden, zusätzliche Kosten, Verzögerungen, Rechtsansprüche
- Quelle – zeigt den Ursprung der Anforderung und woher sie kam – diese Information kann manchmal zur Validierung, Konfliktlösung, Änderung oder Löschung benötigt werden
- Begründung – gibt Grund für die Notwendigkeit der Anforderung und die Ziele der Stakeholder an
- Schwierigkeit - Schätzung des Aufwands, der zur Umsetzung der Anforderung erforderlich ist
- Art - zeigt, ob es sich um eine funktionale Anforderung, Qualitätsanforderung oder Randbedingung handelt

Attribute und Sichten

Attributierungsschemata





Attribute und Sichten (2)

3 Arten von Sichten

- **Selektive Sichten** geben Auskunft über eine gezielte Auswahl von Anforderungen anstatt aller Anforderungen. Zum Beispiel eine Sicht nur auf die neuesten Versionen der Anforderungen, oder auf die Anforderungen mit hoher Priorität für die Stakeholder
- **Projektive Sichten** zeigen eine Auswahl der verfügbaren Daten (Attribute) der Anforderungen, z.B. nur die Identifikation, die Versionsnummer und den Namen
- **Verdichtende (aggregierte) Sichten** finden sich in Zusammenfassungen, Summen oder Durchschnittswerte, die aus einer Reihe von Anforderungen berechnet werden. Ein Beispiel wäre die Gesamtzahl der Anforderungen pro Abteilung

Attribute und Sichten

Beispiel einer projektiven Sicht auf 4 Attribute

Projektive Sicht (nur 4 Attribute)						
Selektive Sicht (nur Sales)	ID	Version	Name	Art	Quelle	Schwierigkeit (1..5)
	1	2,0	Berechnen	Funktional	Sales	3
	2	1.0	Response	Qualität	Sales	2
	3	1.2	VAT	Randbedingung	Sales	1
	4	0.2	Foreign VAT	Randbedingung	Sales	4
	5	2.0	Delivery date	Funktional	Logistics	2
	6	0.1	Track & trace	Funktional	Logistics	5
	7	1.1	Courier	Funktional	Logistics	1
	8	0.2	Routing	Funktional	Logistics	4
	9	1.2	Accessibility	Qualität	Logistics	3
Aggregierende Sicht	Funktional	5	Qualität	2	Randbedingung	2

6.6 Verfolgbarkeit



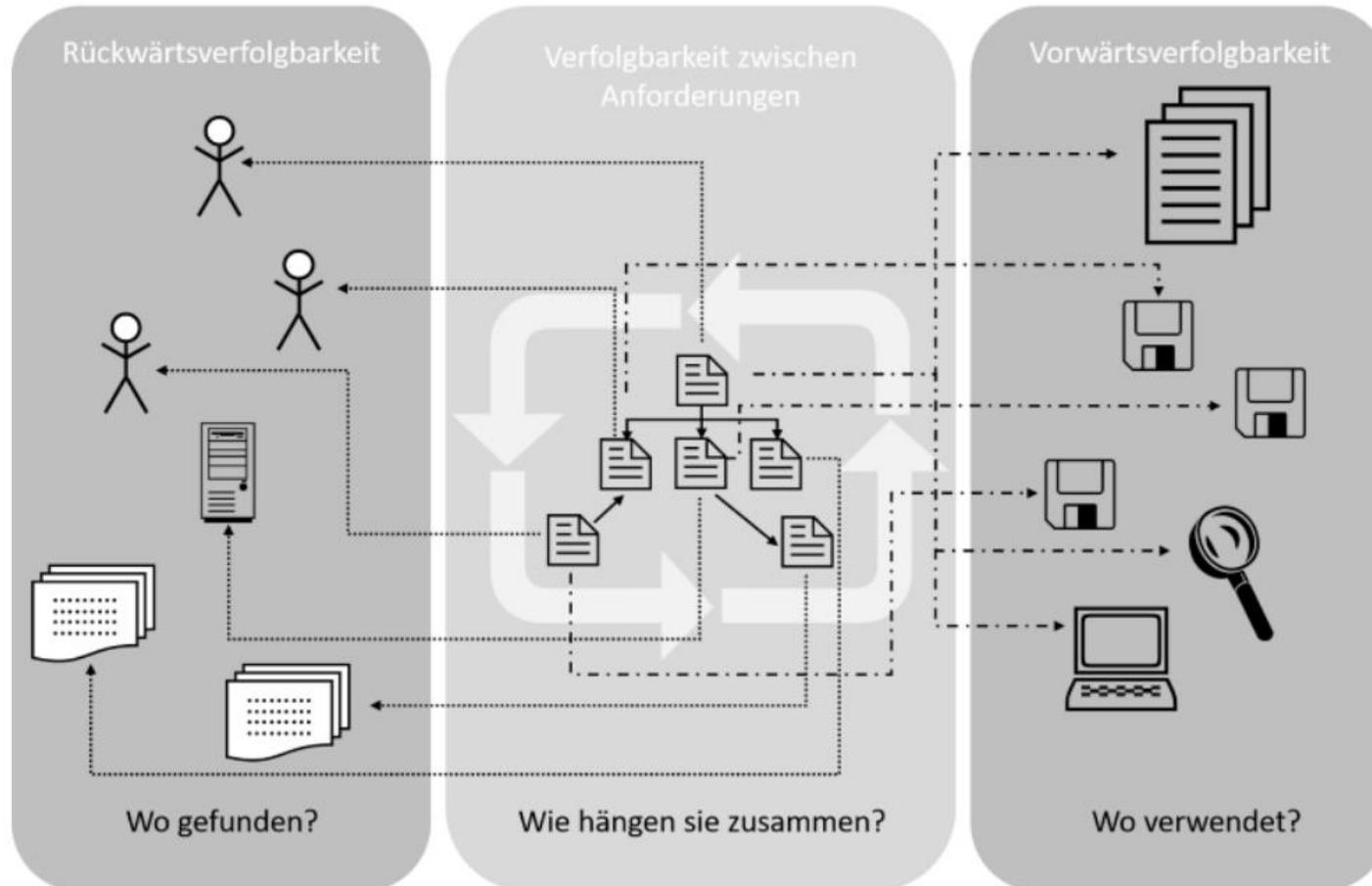
Verfolgbarkeit (1)

Die 3 Arten der Verfolgbarkeit

- **Rückwärtsverfolgbarkeit:**
 - Was war der Ursprung einer bestimmten Anforderung?
 - Wo wurde es ermittelt?
 - Welche Quellen (Stakeholder, Dokumente, andere Systeme) wurden bei der Ermittlung analysiert?
 - Die Rückwärtsverfolgbarkeit heißt auch PreRequirements-Specification-Traceability
- **Vorwärtsverfolgbarkeit:**
 - Wo wird diese Anforderung verwendet?
 - Welche Arbeitsergebnisse (Module, Testfälle, Prozeduren, Handbücher) basieren darauf?
 - Die Vorwärtsverfolgbarkeit heißt auch PostRequirements-Specification Traceability
- **Verfolgbarkeit zwischen Anforderungen:**
 - Hängen andere Anforderungen von dieser Anforderung ab oder umgekehrt?
 - Handelt es sich bei der Anforderung um eine Verfeinerung einer übergeordneten Anforderung?
 - Wie hängen sie zusammen?

Verfolgbarkeit (2)

Die 3 Arten der Verfolgbarkeit



Verfolgbarkeit (3)

Dokumentation der Beziehung zwischen Arbeitsprodukten

- Implizite Dokumentation durch Dokumentenstrukturen, Standardvorlagen und/ oder Namenskonventionen
- Explizite Dokumentation auf Basis der eindeutigen Identifizierung – wie bspw.
 - Durch die Verwendung spezifischer Attribute wie z.B. Quellen
 - In Dokumenten, durch Hinzufügen von Verweisen auf Vorgängerdokumente, andere Arbeitsprodukte oder individuelle Anforderungen
 - Durch Entwicklung einer Verfolgbarkeitsmatrix in einer Tabellenkalkulation oder einer Datenbanktabelle
In textueller Dokumentation, unter Verwendung von Hyperlinks im Wiki-Stil Visualisierung von Verfolgbarkeitsbeziehungen in einem Verfolgbarkeitsdiagramm
 - In vielen Fällen bietet ein Anforderungs- oder Konfigurationsmanagement Tool Funktionen zur Unterstützung der Verfolgbarkeit

Verfolgbarkeit (4)

Dokumentation der Beziehung zwischen Arbeitsprodukten

Beispiel einer Verfolgbarkeitsmatrix

Quelle	A1	A2	A3	A4	A5	A6	A7
<i>Interview Frau Schmidt 8. Juni</i>	X	X			X		
<i>Zusammenfassender Fragebogen 12. Mai</i>	X			X		X	X
<i>Bericht der Feldbeobachtung 3.7.</i>			X	X	X		
<i>Betriebsvorschrift 17.a.02</i>			X			X	X
<i>API-Dokumentation HR-System v3.0.2.a</i>	X			X	X		

Übungen 7: Verfolgbarkeit

- Siehe Übungsdokument „Schulung IREB FL_Tag 3_Übungen“

6.7 Umgang mit Änderungen

Umgang mit Änderungen (1)

- Änderungen sollen nicht als Bedrohung, sondern als Chance gesehen werden
- Unabhängig davon ist eine Änderung jedoch auch ein Risiko für Kostenerhöhungen, zeitliche Verschiebungen etc., daher müssen sie sorgfältig gehandhabt werden
- Ursachen von Änderungen: Fehler, Wandel von Nutzungswünschen, Gesetzesänderungen, neue Technologien,...
- Änderungen können sich auf einzelne Anforderungen oder die Anforderungsbasis beziehen

Umgang mit Änderungen (2)

- Entscheidung, Planung und Implementierung von Änderungen sind abhängig vom Zeitpunkt und Entwicklungsvorgehen
- Lineares Vorgehen: Dokumentation in einem Änderungsantrag (Change Request), Prüfung und Freigabe durch Change Control Board (in Projekten) oder Change Advisory Board (im Betrieb)
- Iteratives Vorgehen: Änderung wird durch Product Owner ins Product Backlog aufnehmen

6.8 Priorisierung

Priorisierung (1)

Anforderungen priorisieren

- Aufgrund begrenzter Ressourcen (Zeit, Geld etc.) können nicht immer alle Anforderungen umgesetzt werden und müssen daher priorisiert werden
- Zuerst Ziel der Priorisierung definieren und danach die Bewertungskriterien zur Priorisierung definieren





Priorisierung (2)

Die 6 Schritte zur Priorisierung von Anforderungen

1. Festlegen der wichtigsten Ziele und Randbedingungen der Priorisierung
2. Definieren gewünschter Bewertungskriterien
3. Festlegen, welche Stakeholder einbezogen werden
4. Festlegen, welche Anforderungen priorisiert werden
5. Auswahl der Priorisierungstechnik (siehe nächste Folie)
6. Durchführung der Priorisierung

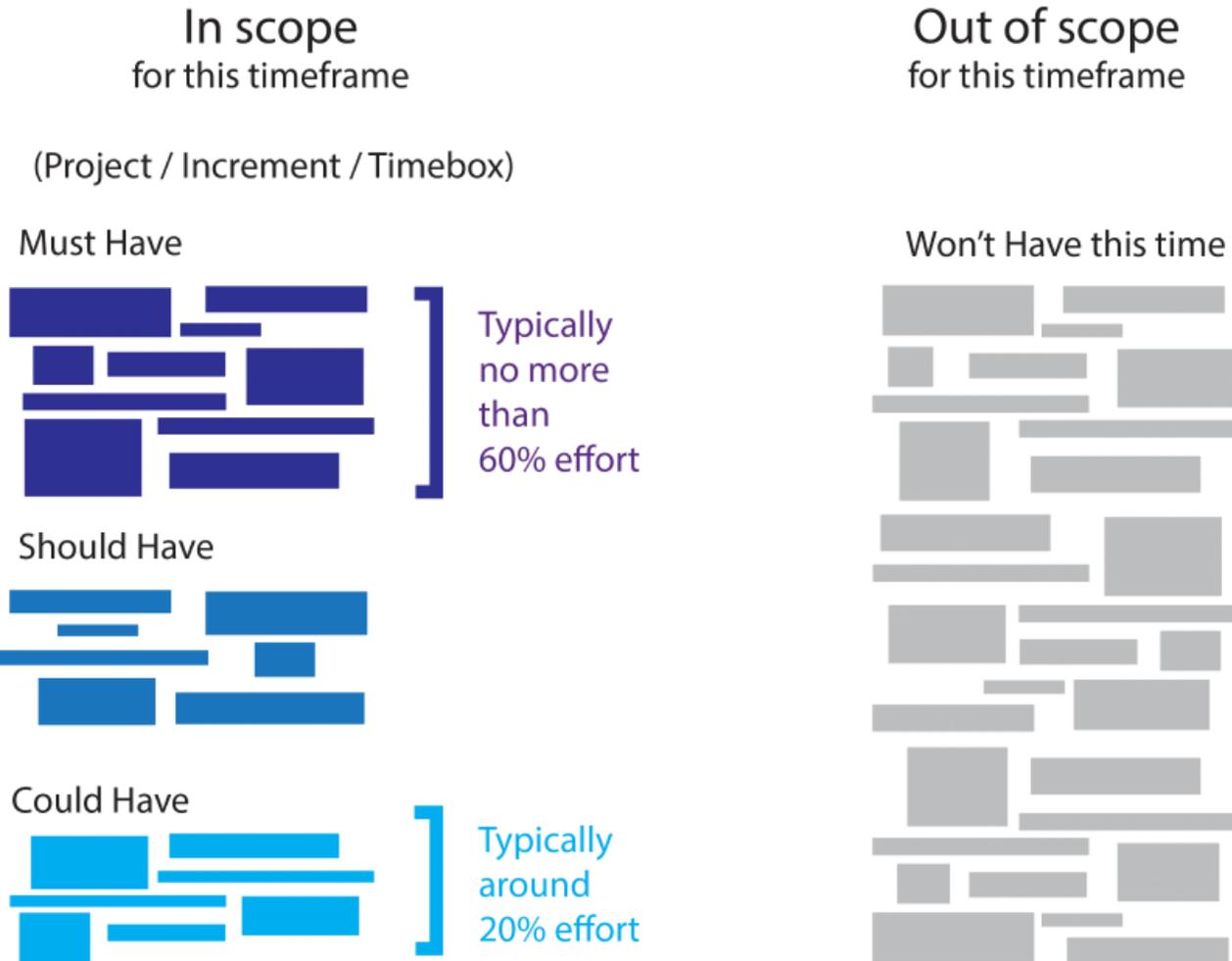


Priorisierung (3)

Kategorien von Priorisierungstechniken

- Ad-hoc-Techniken
 - Expert*innen weisen definierten Anforderungen Prioritäten zu
 - Günstig, aber eher subjektiv
 - Techniken wie Top-10-Rangliste, MoSCoW-Priorisierung, Kano-Analyse
- Analytische Techniken
 - Systematischer Prozess zur Zuweisung von Prioritäten
 - Expert*innen weisen mehreren Beurteilungskriterien Gewichtungen zu
 - Kostenintensiver, aber klarer Einblick in die Faktoren der Priorisierung
- Aufwand für Priorisierungen sollten immer in Relation zum Nutzen gesetzt werden

Priorisierung (4) – Beispiel MoSCoW



Wissenswiederholung

LE 6.1 - 6.8

Wissenswiederholung (1)

- Nennen Sie die 3 Ebenen auf welchen Requirements Management erfolgen kann.

Wissenswiederholung (1)

- Nennen Sie die 3 Ebenen auf welchen Requirements Management erfolgen kann.
 1. Auf den einzelnen Anforderungen (z.B. einzelne User Story in einem agilen Projekt)
 2. Auf den Arbeitsprodukten, die eine bis mehrere Anforderungen (z.B. Use Case Diagramm) enthalten können
 3. Auf dem System (z.B. gesamtes Anforderungsspezifikationsdokument einer Software) hinsichtlich seiner Arbeitsprodukte und der darin enthaltenen Anforderungen

Wissenswiederholung (2)

- Nennen Sie die 2 Gründe und die 3 Aspekte der Versionsstrategie bei RE Prozessen

Wissenswiederholung (2)

- Nennen Sie die 2 Gründe und die 3 Aspekte der Versionsstrategie bei RE Prozessen.

Gründe für die Versionskontrolle:

1. Wenn Änderung schief geht kann neue Änderung eingeführt oder bei früherer Version fortgesetzt werden
2. Wissen über Historie des Arbeitsprodukts hilft, um über Änderungen zu entscheiden

Versionsstrategie:

1. Versionsnummer: Identifizierung jeder Version eines Arbeitsprodukts durch Versionsnummer & -datum
2. Änderungshistorie: Klare Änderungsbeschreibung - eindeutig mit der Versionsnummer verknüpft
3. Strikte Regelung für die Speicherung von Versionen, um alte Versionen abzurufen

Wissenswiederholung (3)

- Was ist eine Konfiguration im RE?

Wissenswiederholung (3)

- Was ist eine Konfiguration im RE?
 - Ein Satz logisch zusammenhängender Arbeitsprodukte die Anforderungen enthalten, um zu zeigen, welche Anforderungen in einer bestimmten Situation gültig sind oder waren

Wissenswiederholung (4)

- Welche Eigenschaften hat eine Konfiguration und welche 2 Dimensionen können Konfigurationen annehmen?

Wissenswiederholung (4)

- Welche Eigenschaften hat eine Konfiguration und welche 2 Dimensionen können Konfigurationen annehmen?

Eigenschaften:

- Logisch Verbunden, Konsistent, Eindeutig, Unveränderlich und Grundlage für das Zurücksetzen

2 Dimensionen einer Konfiguration:

1. Produktdimension: enthält Anforderungsauswahl, wie bspw. z.B. alle Anforderungen der französischen Systemversion (britische Systemversion wäre andere Konfiguration)
2. Versions Dimension: In bestimmter Konfiguration ist jede Anforderung genau in einer Version vorhanden. Bspw. bei Softwarerelease

Wissenswiederholung (5)

- Was ist eine Basislinie/ Baseline im Requirements Engineering?

Wissenswiederholung (5)

- Was ist eine Basislinie/ Baseline im Requirements Engineering?
 - Eine stabile, validierte und änderungskontrollierte Konfiguration, die einen Meilenstein oder eine andere Art Haltepunkt im Projekt markiert – bspw. Konfiguration am Ende der Entwurfsphase, bei agilen Projekten dient Sprint-Backlog als Baseline

Wissenswiederholung (6)

- Welche Attribute von Anforderungen kennen Sie?

Wissenswiederholung (6)

- Welche Attribute von Anforderungen kennen Sie?
 - Identifizierung, Priorität für den Stakeholder, Abhängigkeit zw. Anforderungen, Risiko, Quelle, Begründung, Schwierigkeit (Aufwandsschätzung), Art der Anforderung

Wissenswiederholung (7)

- Welche 3 Arten der Sichten kennen Sie?

Wissenswiederholung (7)

- Welche 3 Arten der Sichten kennen Sie?
 1. Selektive Sichten
 2. Projektive Sichten
 3. Verdichtende (aggregierte) Sichten

Wissenswiederholung (8)

- Wie lauten die drei Arten der Verfolgbarkeit von Anforderungen?

Wissenswiederholung (8)

- Wie lauten die drei Arten der Verfolgbarkeit von Anforderungen?
 1. Rückwärtsverfolgbarkeit (Ursprung einer bestimmten Anforderung)
 2. Vorwärtsverfolgbarkeit (Wo wird die Anforderung verwendet)
 3. Verfolgbarkeit zwischen Anforderungen (Zusammenhang zw. Anforderungen)

Wissenswiederholung (9)

- Nennen Sie die 6 Schritte zur Priorisierung von Anforderungen.

Wissenswiederholung (9)

- Nennen Sie die 6 Schritte zur Priorisierung von Anforderungen.
 1. Festlegen der wichtigsten Ziele und Randbedingungen der Priorisierung
 2. Definieren gewünschter Beurteilungskriterien
 3. Festlegen, welche Stakeholder einbezogen werden
 4. Festlegen, welche Anforderungen priorisiert werden
 5. Auswahl der Priorisierungstechnik (siehe nächste Folie)
 6. Durchführung der Priorisierung

7 Werkzeugunterstützung

7.1 Werkzeuge im Requirements Engineering

Werkzeuge im Requirements Engineering (1)

Arten von Werkzeugen und deren unterstützte Aspekte

- Zur Verwaltung von Anforderungen
 - Definition und Speichern von **Anforderungsattributen** zur Identifizierung und Sammlung von Daten über Arbeitsprodukte und Anforderungen
 - Erleichterung und Dokumentation der **Priorisierung** von Anforderungen
 - **Lebenszyklusmanagement**
 - **Versionskontrolle, Konfigurationen** und Baselines
 - **Nachverfolgung** und **Rückverfolgung** von Anforderungen sowie von Defekten in den Anforderungen und Arbeitsprodukten
 - **Änderungsmanagement** für Anforderungen

Werkzeuge im Requirements Engineering (2)

Arten von Werkzeugen und deren unterstützte Aspekte

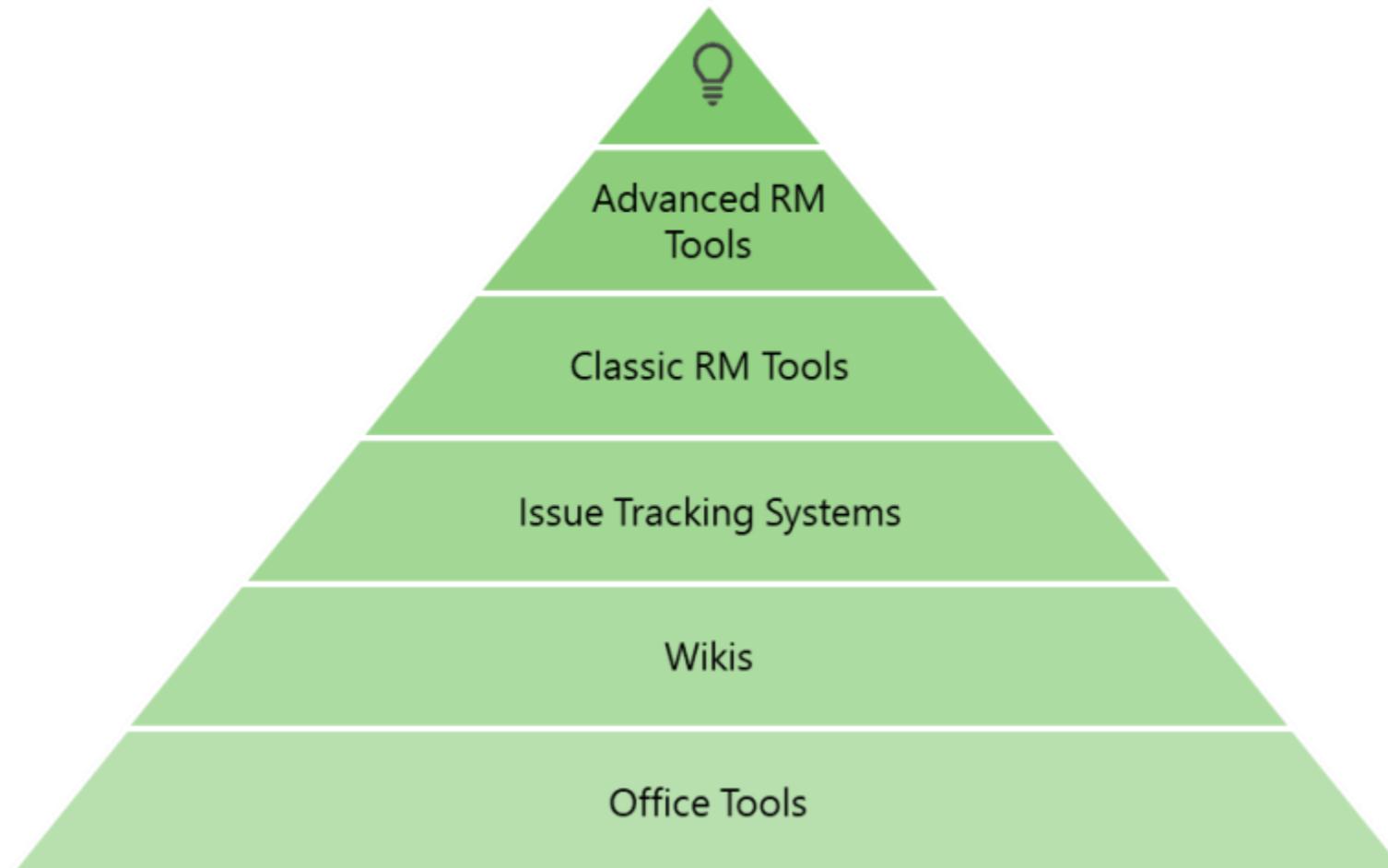
- Zur Steuerung und Unterstützung des RE Prozesses
 - Messung und Berichterstattung über den RE Prozess und Workflow
 - Messung und Berichterstattung über die Produktqualität zur Suche nach Fehlern & Mängeln, die zur Verbesserung der Produktqualität dienen
 - Verwaltung des RE-Workflows
- Zur Dokumentation des Kenntnisstandes über die Anforderungen
 - Gemeinsame Nutzung von Anforderungen
 - Schaffung eines gemeinsamen Verständnisses der Anforderungen
 - Auskunft über den Zustand des Arbeitsprodukts (wer kennt die Anforderung und hat bereits zugestimmt)
 - Sicherung der Anforderungen als rechtliche Verpflichtung
 - Dokumentation von Konflikten und deren Lösung

Werkzeuge im Requirements Engineering (3)

Arten von Werkzeugen und deren unterstützte Aspekte

- Zur Modellierung von Anforderungen
 - Strukturierung eigener Gedanken
 - Spezifikation von Anforderungen in formeller Sprache
 - Unterstützen von Erstellen und Verwalten von modellbasiert dokumentierten Anforderungen
- Zur Zusammenarbeit im RE
 - Tools zum kollaborativen Arbeiten
 - Videokonferenztools
- Zum Testen und/ oder Simulieren von Anforderungen
 - Tools zur Validierung

Werkzeuge



Das OSSENO-Reifegradmodell: Tools für Requirements Engineering (Anforderungsmanagement)

7.2 Werkzeugeinführung

Werkzeugeinführung (1)

- Werkzeug kann helfen den RE Prozess effektiv und effizient zu konfigurieren
- Werkzeuge sind **unterstützende Mittel** für Requirements Engineer und RE
- Bevor mit der Auswahl eines RE-Werkzeuges begonnen wird, müssen **Verantwortlichkeiten, RE-Vorgehensweisen** und **Techniken** feststehen
- Auswahl der geeigneten RE-Werkzeuge beginnt mit der Definition der Ziele und/oder Probleme, die im RE-Prozess gelöst werden sollen
- Im nächsten Schritt muss der Systemkontext bestimmt werden - d.h. Stakeholder, Prozesse, Ereignisse usw.

Werkzeugeinführung (2)

Für die Einführung eines RE-Tools relevante Aspekte

- **Alle Lebenszykluskosten berücksichtigen**
 - Lizenzkosten,
 - Anschaffungskosten,
 - Hardwareressourcen,
 - Ressourcenkosten für Implementierung, Betrieb und Wartung
 - Ressourcenkosten für Benutzerunterstützung (Support), sowie
 - Kosten für die Schulung von Mitarbeitern

Werkzeugeinführung (3)

Für die Einführung eines RE-Tools relevante Aspekte

- **Benötigte Ressourcen einplanen**
 - Personen, zur Leitung des Auswahlprozesses - Requirements Engineers
 - Ressourceneinsatz bei Implementierung, Betrieb, Wartung und Support
 - Ressourcen für Schulung von Mitarbeitern
- **Vermeidung von Risiken durch**
 - Durchführung von Pilotprojekten
 - Anwendung des Werkzeugs auf ein unkritisches Projekt/System
 - Redundante Verwendung des Werkzeugs zu einem bestehenden Projekt
 - Anwendung des Werkzeugs auf eine fiktive Situation/ein fiktives Projekt
 - Importieren/Kopieren von Anforderungen eines abgeschlossenen Projekts

Werkzeugeinführung (4)

Für die Einführung eines RE-Tools relevante Aspekte

- **Bewerten des Werkzeugs nach definierten Kriterien**
 - Durchführung einer Werkzeugevaluierung
 - Selektierung anhand Herstellerbefragung und definierte Muss-Kriterien
 - Erstellung eines Kriterienkatalogs
- **Unterweisung der Mitarbeiter*innen in der Toolanwendung**
 - Schulung der Mitarbeiter*innen zum Softwareeinsatz

Wissenswiederholung

LE 7.1 – 7.2

Wissenswiederholung (1)

- Welche Arten von Werkzeugen und deren unterstützten Aspekte kennen Sie?

Wissenswiederholung (1)

- Welche Arten von Werkzeugen und deren unterstützten Aspekte kennen Sie?
 - Zur Verwaltung von Anforderungen
 - Zur Unterstützung des RE Prozesses
 - Zur Dokumentation des Kenntnisstandes über die Anforderungen
 - Zur Modellierung von Anforderungen
 - Zur Zusammenarbeit im Requirements Engineering
 - Zum Testen und/ oder Simulieren von Anforderungen

Wissenswiederholung (2)

- Welche Ressourcen werden im Zuge der Werkzeugeinführung im RE benötigt?

Wissenswiederholung (2)

- Welche Ressourcen werden im Zuge der Werkzeugeinführung im RE benötigt?
 - Personen, zur Leitung des Auswahlprozesses - Requirements Engineers
 - Ressourceneinsatz bei Implementierung, Betrieb, Wartung und Support
 - Ressourcen für Schulung von Mitarbeitern

Wissenswiederholung (3)

- Welche Aspekte sind bei der Planung einer Werkzeugeinführung zu beachten?

Wissenswiederholung (3)

- Welche Aspekte sind bei der Planung einer Werkzeugeinführung zu beachten?
 - Berücksichtigung sämtlicher Lebenszykluskosten zusätzlich zu den Lizenzkosten
 - Benötigte Ressourcen einplanen
 - Risiken zu vermeiden, zB Pilotprojekte nutzen
 - Bewertung des Werkzeugs nach definierten Kriterien
 - Mitarbeiter in der Anwendung des Tools unterweisen