

# IREB - Requirements Engineering

## LE 1 Einleitung und Grundlagen (K1)

- Typische Gründe für mangelhaftes RE:
  - **Selbstverständlichkeit der Stakeholder für Annahmen**
  - **Kommunikationsprobleme durch untersch. Erfahrungs- & Wissensstände**
  - **Projektdruck des Arbeitgebers**
- 4 Haupttätigkeiten eines RE:
  - **Ermitteln | Dokumentieren | Prüfen/Abstimmen | Verwalten von Anforderungen**
- Unterschiedliche Ebenen von Anforderungen:
  - Stakeholder Anforderungen, System- bzw. Softwareanforderungen
- Eigenschaften eines RE die er mitbringen muss:
  - Analytisches Denken, Empathie, Konfliktlösungsfähigkeit, Moderationsfähigkeit, Selbstbewusstsein, Überzeugungsfähigkeit
- 3 Arten von Anforderungen:
  - **Funktionale Anforderungen | Qualitätsanforderungen | Randbedingungen**  
→ Nicht-funkt. Anforderungen = Qualitätsanforderungen | Randbedingungen
- **Qualitätsanforderungen:**
  - **Performanz | Sicherheit | Zuverlässigkeit | Benutzbarkeit | Änderbarkeit | Übertragbarkeit**

## LE 2 System und Systemkontext abgrenzen (K2)

- Anforderungen: Ursprung = Rechtfertigung  
→ liegt im Systemkontext des geplanten Systems. Ursprung = Menge aller Kontextaspekte im Systemkontext:
  - **Personen** (Stakeholder [-gruppen])
  - **Systeme im Betrieb** (HW, SW, Systeme)
  - **Prozesse** (technisch/physikalisch, Geschäftsprozesse)
  - **Ereignisse** (technisch / physikalisch)
  - **Dokumente** (z.B. Gesetze, Standards, Systemdokumentation)
- **Systemabgrenzung** = Festlegung der Aspekte welche durch das geplante System abgedeckt werden VS. Systemumgebung
- **Systemgrenze** oft erst gegen Ende des RE-Prozesses präzise festgelegt  
→ Es gibt eine Grauzone, in welcher die mögliche Systemgrenze liegt  
→ Grauzone: Verschiebung der Systemgrenze innerhalb Grauzone oder Verschiebung der Grauzone selbst
- **Kontextabgrenzung** = Identifikation der Umgebungsteile, welche eine Beziehung zum System haben
- **Kontextgrenze** kann sich auch verschieben, z.B. wenn eine gesetzliche Vorschrift wider Erwarten keine Auswirkung auf System hat  
→ Grauzone: Identifizierte Aspekte der Umgebung, bei welchen noch nicht klar ist, ob diese Aspekte eine Beziehung zum geplanten System haben oder nicht
- Zur Dokumentation vom Systemkontext (v.a. System- und Kontextgrenzen)  
→ oft via Use-Case- und Datenflussdiagrammen
- **Datenflussdiagramme:** Darstellung von Datenflüssen welche Ursprung und Endpunkt zwischen dem System und dessen Umgebung haben
- **Use-Case-Diagramme:** Modellierung von Akteuren (Personen / andere Systeme) und deren Nutzungsbeziehungen in Systemumgebung

## LE 3 Anforderungen ermitteln (K2)

- Grundlagen für die Anforderungsermittlung = **Systemkontext | Anforderungsquellen**
- Anforderungsquellen = **Stakeholder | Dokumente | Altsysteme**
- Aufgabe RE → **Ziele & Anforderungen** aus den Quellen zu sammeln
- Dokumentation der Anforderungsquellen brauchen mindestens folgende Informationen:
  - Name und Funktion
  - Weitere Personen- und Kontaktdaten
  - Zeitliche und räumliche Verfügbarkeit während der Projektlaufzeit
  - Relevanz des Stakeholders und sein Wissensgebiet und -umfang
  - Seine Ziele und Interessen bezogen auf das Projekt
- Mit Stakeholder die Aufgaben, Verantwortungsbereiche, Weisungsbefugnisse usw. festlegen, schriftlich oder mündlich
- Aus Stakeholdervereinbarung resultieren für jeden Stakeholder **Rechte und Pflichten**
- Effektiver Umgang mit Stakeholdern beugt Mangel an Motivation und Konflikten vor

- Stakeholder sollten Projektbeteiligte und nicht nur Projektbetroffene sein
- Entscheidend für die Anforderungsermittlung: Wissen, welche Bedeutung die Anforderungen für die Zufriedenheit der Stakeholder haben
- Zufriedenheit nach Kano-Modell → 3 Kategorien: **Basisfaktoren | Leistungsfaktoren | Begeisterungsfaktoren**
  - **Basisfaktoren:** Sind so grundlegend und selbstverständlich, dass sie den Kunden erst bei Nichterfüllung bewusst werden (implizite Erwartungen). Werden die Grundforderungen nicht erfüllt, entsteht Unzufriedenheit; werden sie erfüllt, entsteht aber keine Zufriedenheit. Die Nutzensteigerung im Vergleich zur Differenzierung gegenüber Wettbewerbern ist sehr gering.
    - Am Beispiel Auto: Sicherheit, Rostschutz
  - **Leistungsfaktoren:** Sind dem Kunden bewusst, sie beseitigen Unzufriedenheit oder schaffen Zufriedenheit abhängig vom Ausmaß der Erfüllung.
    - Am Beispiel Auto: Fahreigenschaften, Beschleunigung, Lebensdauer, Verbrauch
  - **Begeisterungsfaktoren:** Sind Nutzen stiftende Merkmale, mit denen der Kunde nicht unbedingt rechnet. Sie zeichnen das Produkt gegenüber der Konkurrenz aus und rufen Begeisterung hervor. Eine kleine Leistungssteigerung kann zu einem überproportionalen Nutzen führen. Die Differenzierungen gegenüber der Konkurrenz können gering sein, der Nutzen aber enorm.
    - Am Beispiel Auto: Sonderausstattung, besonderes Design
- **Ermittlungstechniken:** Zweck, folgende Anforderungen herauszufinden:
  - **bewusste | unbewusste | unterbewusste**
- **Wichtige Einflussfaktoren** auf Wahl der Ermittlungstechnik:
  - **Risikofaktoren | menschliche Einflüsse | organisatorische Einflüsse | fachlich-inhaltliche Einflüsse | angestrebter Detaillierungsgrad der Anforderungen**
- Für unterschiedliche Produkte des RE werden verschiedene **Techniken** benötigt:
  - **Befragungstechniken** → Interview, Fragebogen
  - **Kreativitätstechniken** → Brainstorming, Brainstorming paradox, Perspektivenwechsel, Analogietechnik
  - **Dokumentenzentrierte Techniken** → Systemarchäologie, Perspektivenbasiertes Lesen, Wiederverwendung von Anforderungen
  - **Beobachtungstechniken** → Feldbeobachtung, Apprenticing
  - **Unterstützende Techniken** → Mind Mapping, Workshops, CRC-Karten (Class-Responsibility-Collaboration-Karten), Audio- und Videoaufzeichnungen, Use-Case-Modellierung, Prototypen
- **Einsatz geeigneter Ermittlungstechniken = projektentscheidende Schlüsselkompetenz**
- Beste Ergebnisse bei **Kombination verschiedener Ermittlungstechniken**

#### LE 4 Dokumentation von Anforderungen (K2)

- Anforderungsdokumente sind komplex
- In RE wichtig, alle wichtigen Informationen zu dokumentieren
- Dokumentationstechnik → jegliche Art der mehr oder weniger formalen Darstellung von Anforderungen, angefangen von der Beschreibung in Prosaform bis hin zu Diagrammen mit einer formalen Semantik
- Viele Personen sind im Lebenszyklus eines Anforderungsdokuments eingebunden
- Dokumentation hat bei der Kommunikation eine zielgerichtete, unterstützende Funktion
- Folgende Faktoren machen diese Unterstützung notwendig:
  - Anforderungen sind langlebig
  - Rechtlich relevant
  - Sollten allen zugänglich sein
- Anforderungsdokumente umfassen (unter anderem) **funktionale Anforderungen**, welche üblicherweise folgende drei verschiedenen Perspektiven eines Systems repräsentieren:
  - **Strukturperspektive | Verhaltensperspektive | Funktionsperspektive**
- Alle drei Perspektiven sind dokumentierbar mittels natürlicher Sprache
- Konzeptuelle Modelltypen (siehe unten) sind eher auf eine dieser Perspektiven spezialisiert
- Effektiv einsetzbare **Formen der Dokumentation:**
  - Natürlichsprachige Anforderungsdokumentation
  - Konzeptuelle Anforderungsmodelle, z.B.:
    - **Use-Case-Diagramme | Klassendiagramme | Aktivitätsdiagramme | Zustandsdiagramme**
  - Mischformen von Anforderungsdokumentation
- Zentrale Bestandteile von Anforderungsdokumenten = **Anforderungen an das betrachtete System**
- Daneben: **Info über Systemkontext | Abnahmebedingungen | Merkmale der technischen Realisierung**
- Um Handhabbarkeit dieser Anforderungsdokumente zu gewährleisten, müssen solche Dokumente möglichst geeignete inhaltliche Strukturierung aufweisen: **Referenzstrukturen** → mehr oder weniger flexibel, praxiserprobte inhaltliche Strukturierung, z.B.: ISO/IEC/IEEE 29148:2011

→ Es hat sich gezeigt, dass die Verwendung von Referenzstrukturen viele positive Effekte mit sich bringt, z.B. fürs Testen

→ Meist sind Referenzstrukturen jedoch nicht 1:1 übernehmbar, da inhaltliche Struktur häufig noch an domänen-, unternehmens- oder projektspezifische Gegebenheiten angepasst werden muss

- Anforderungsdokumente = **Grundlage** für verschiedene Aufgaben im Laufe des Projektes:
  - Planung
  - Architekturentwurf
  - Implementierung
  - Test
  - Änderungsmanagement
  - Systemnutzung und Systemwartung
  - Vertragsmanagement
- **Qualitätskriterien für Anforderungsdokument** nötig, sodass diese geeignete Basis für nachgelagerte Entwicklungsschritte ist:
  - Eindeutigkeit und Konsistenz
  - Klare Struktur
  - Modifizierbarkeit und Erweiterbarkeit
  - Vollständigkeit und Verfolgbarkeit
- **Qualitätskriterien für Anforderungen:**
  - Abgestimmt, Eindeutig, Notwendig
  - Konsistent, Prüfbar, Realisierbar
  - Verfolgbar, Vollständig, Verständlich
- Weitere **Stilregeln** für natürlichsprachige Anforderungen zur Förderung der Lesbarkeit:
  - Kurze Sätze und Absätze
  - Nur eine Anforderung pro Satz formulieren
- **Glossar** hilft Konflikte zu vermeiden, welche im RE auftreten können durch unterschiedliches Begriffsverständnis der beteiligten Personen. Glossar = Sammlung von Begriffsdefinitionen für:
  - Kontextspezifische Fachbegriffe
  - Abkürzungen und Akronyme
  - Alltägliche Begriffe, die im gegebenen Kontext eine spezifische Bedeutung haben
  - Synonyme & Homonyme (= Gleiches Wort aber mit unterschiedlichen Bedeutungen)
- Für Glossar wichtige **Umgangsregeln:**
  - Zentrale Verwaltung
  - Verantwortlichkeiten zur Glossarpflege
  - Projektbegleitende Pflege
  - Allgemeine Zugänglichkeit
  - Verbindliche Verwendung
  - Herkunft der Begriffe (sollte im Glossar enthalten sein)
  - Abstimmung mit Stakeholdern
  - Einheitliche Struktur der Einträge ins Glossar
- Möglichst früher Start des Glossars = Reduktion des späteren Angleichungsaufwandes

## LE 5 Natürlichsprachige Dokumentation von Anforderungen (K2)

- Natürliche Sprache oft mehrdeutig und interpretierbar
  - notwendig, bei Einsatz von Sprache genau diesem Gesichtspunkt besondere Aufmerksamkeit zu widmen
- Bei den Vorgängen «**Wahrnehmung**» und «**Darstellung**» treten sogenannte **Transformationsprozesse** auf
- Tatsache, dass diese Transformationsprozesse gewissen Regeln gehorchen, kann der Anforderungsingenieur nutzen: Gezieltes Nachfragen, was der Autor wirklich gemeint hat
- Die fünf wichtigsten Transformationsprozesse sind:
  - Nominalisierung
  - Substantive ohne Bezugsindex
  - Universalquantoren (= Umfangswörter wie: alle, keine, jeder, niemals, etc.)
  - Unvollständig spezifizierte Bedingungen
  - Unvollständig spezifizierte Prozesswörter
- **Satzschablone** = einfach erlern- und einsetzbarer Ansatz zur Reduzierung sprachlicher Effekte während der Formulierung von Anforderungen
- Sie unterstützt den Autor effektiv dabei, qualitativ hochwertige Anforderungen zu erstellen
- 5 Schritte zur Formulierung von Anforderungen mittels Satzschablone:
  - Festlegen der rechtlichen Verbindlichkeit
  - Benennung des Kerns der Anforderung

- Charakterisierung der Aktivität des Systems
- Objekte einfügen
- Formulieren von logischen und zeitlichen Bedingungen
- Festlegung der Verbindlichkeit der Verben «muss», «sollte», «wird», «kann» kann im Anforderungstext erfolgen
- Ändern sich diese Verbindlichkeiten, dann ändern sich auch die Anforderungen
- Weitere Möglichkeit für die Dokumentation der Anforderungen: Einsatz von **Attributen**
- Beste Ergebnisse = Einsatz von Satzschablone schulen & als Hilfsmittel darstellen, aber nicht als Muss vorschreiben

## LE 6 Anforderungen modellbasiert dokumentieren (K2)

- Hier bei RE: Modell = abstrahiertes Abbild einer existierenden Realität / Vorbild einer zu schaffenden Realität
- Generell: Modell = abstrakte Darstellung einer vorhandenen Entität oder einer Entität, die erst noch erzeugt wird  
→ Entität = Ausschnitt aus der Realität / beliebige andere Menge von Elementen / Phänomenen, kann auch anderes Modell sein
- In Bezug auf Modell: Entität = Original
- Modelle besitzen 3 zentrale Eigenschaften:
  - **Abbildungseigenschaft:** Modelle bilden eine Realität ab
  - **Verkürzende Eigenschaft:** Modelle verkürzen die abgebildete Realität
  - **Pragmatische Eigenschaft:** Modelle werden für eine spezifische Verwendung konstruiert
- Im RE: oft konzeptuelle Modelle, welche die abzubildende Realität durch eine Menge grafischer Elemente modelliert
- Dafür eingesetzte konzeptuelle Modellierungssprachen, definiert über:
  - **Syntax** (= Modellierungselemente und deren gültige **Kombination**) und
  - **Semantik** (= **Bedeutung** der Modellierungselemente)
- Anforderungsmodelle = konzeptuelle Modelle, die Anforderungen an das zu entwickelnde System dokumentieren
- Folgende Vorteile ergeben sich damit im Vergleich zur natürlichsprachigen Dokumentation:
  - **Bildhafte Darstellung** von Information → Schnellere Erfassung und Memorisierung
  - **Modellierung** einer gezielten Perspektive
  - Zweckmäßige **Abstraktion** der Realität
- Kombination von natürlicher Sprache und Anforderungsmodellen vereinigt die Vorteile beider Dokumentationsarten
- Ziel = Beschreibung der Intention eines Stakeholders
- Intention = charakteristisches Merkmal des zu entwickelnden Systems / des zugehörigen Entwicklungsprojekts
- Dokumentation von Zielen: natürlichsprachig / in Form von Modellen
- Wesentlicher Bestandteil der Dokumentation der Ziele = Beschreibung von **Verfeinerungsbeziehungen** (= **Dekompositionsbeziehungen**) zwischen einem übergeordneten und untergeordneten Zielen
- Es gibt 2 Arten der Dekomposition:
  - **UND-Dekomposition** → alle Teilziele müssen erfüllt sein, um das übergeordnete Ziel zu erfüllen
  - **ODER-Dekomposition** → mindestens ein Teilziel muss erfüllt sein, um das übergeordnete Ziel zu erfüllen
- Dokumentation solcher Deko-Beziehungen → häufig in Form von Und-Oder-Bäumen
- **Use Cases** → dokumentiert die Funktionalität eines Systems **aus Nutzungssicht**
- Es gibt zwei sich ergänzende Dokumentationstechniken: **Use Case-Diagramme** | **Use Case-Spezifikationen**
- Use Case-Diagramme sind leicht verständliche Modelle, welche die aus Nutzungssicht notwendigen Funktionen des betrachteten Systems, deren Beziehungen untereinander und den Kontext des Systems dokumentieren
- Typische Modellierungselemente von Use Case-Diagrammen:
  - Akteure (Personen oder andere Systeme) im Systemkontext

- Systemgrenze
- Use Cases
- Verschiedene Typen von Beziehungen zwischen diesen Modellierungselementen
- **Use Case-Spezifikationen** = genauere Spezifikation der wesentlichen Eigenschaften einzelner Use Cases, im Gegensatz zu den überblicksartigen **Use Case-Diagrammen**  
→ Dafür werden vorgegebene Schablonen ausgefüllt, welche u.a. folgende Abschnitte hat:
  - Eindeutiger Bezeichner
  - Name
  - Auslösendes Ereignis
  - Akteure
  - Ergebnis
  - Vor- und Nachbedingung
  - Verschiedene Arten von Szenarien
- **Szenarien** beschreiben exemplarische Ereignisfolgen, die zur erfolgreichen Ausführung des Use Case führen (**Hauptszenarien & Alternativszenarien**) oder explizit beschreiben, wie innerhalb des Ausführung des Use Case auf Ausnahmesituationen reagiert werden soll (**Ausnahmeszenarien**)
- Die drei überlappenden Modellierungsperspektiven für die Modellierung und deren typische Vertreter:
 

○ <b>Strukturperspektive:</b>	Entity-Relationship-Modell		UML-Klassendiagramm
○ <b>Funktionsperspektive:</b>	Datenflussdiagramme		UML-Aktivitätsdiagramme
○ <b>Verhaltensperspektive:</b>	Endliche Automaten		State Charts

### Strukturperspektive

- Dokumentation der Struktur von Daten sowie von Nutzungs- und Abhängigkeitsbeziehungen im Systemkontext
- Traditionell modelliert durch **Entity-Relationship-Diagramme**, welche die Struktur der zu modellierenden Realität durch 3 Modellelemente dokumentieren:
  - **Entitätstypen | Beziehungstypen | Attribute**
- **Kardinalität:** Dokumentation der Häufigkeit, in der eine Instanz (Entität) eines Entitätstyps an einer Beziehung eines spezifischen Beziehungstyps teilnimmt (Kardinalität = Beziehungen dazwischen: 1:1, 1:n, n:m, etc.)
- Weiterer Ansatz: **UML-Klassendiagramme** → besteht aus einer Menge von Klassen und Assoziationen zwischen diesen Klassen. Häufig verwendete Modellelemente von UML-Klassendiagrammen:
  - Klassen
  - Assoziationen (mit Multiplizitäten und Rollen)
  - Aggregations- und Kompositionsbeziehungen
  - Generalisierungsbeziehungen

### Funktionsperspektive

- Fokus liegt auf der Verarbeitung von **Eingabedaten** aus der Umgebung des betrachteten Systems in **Ausgabedaten** für die Umgebung
- Ansätze: Funktionsmodelle, wie z.B. die strukturierte Analyse nach Tom DeMarco mittels **Datenflussdiagrammen**
- **Kontextdiagramm:** Grafische Darstellung eines Systems mit seinem Systemkontext
- Datenflussdiagramm = Kontextdiagramm, falls damit das System grafisch dargestellt wird, insbesondere wenn es zur Kontextabgrenzung verwendet wird
- Modellelemente in Datenflussdiagrammen:
  - **Prozesse | Datenflüsse | Datenspeicher | Terminatoren**
- In Datenflussdiagrammen ist **kein Kontrollfluss** oder die innere Arbeitsweise von Prozessen ersichtlich  
→ Ergänzung durch zusätzliche, strukturierte Beschreibungsformen  
→ z.B. Minispezifikation der strukturierten Analyse  
→ Definition der inneren Abläufe von Prozessen
- UML 2.0 → Datenflüsse lassen sich durch explizite Modellierung von Objektflüssen in Aktivitätsdiagrammen repräsentieren → Herstellung einer Analogie zu den Datenflussdiagrammen
- Aktivitätsdiagramme → Modellierung von Aktivitätsknoten & Kontrollflüsse zwischen diesen Knoten

- Objektflüsse = spezielle Ausprägungsform von Kontrollflüssen
- Synchronisationsbalken in Aktivitätsdiagrammen erlauben Modellierung von nebenläufigen Kontroll- und Objektflüssen
- Alternative Kontroll- und Objektflüsse → Beschreibbar durch Entscheidungsknoten
- **Modellelemente in UML-Aktivitätsdiagrammen** der UML 2:
  - Aktion, Start- und Endknoten
  - Kontrollfluss und Objektfluss
  - Entscheidungsknoten
  - Zusammenführung alternativer Kontrollflüsse
  - Fork und Join (→ Nebenläufigkeiten)
  - Hierarchisierungselemente

### Verhaltensperspektive

- Anforderungsmodellierung in der Verhaltensperspektive  
→ Modellierung des dynamischen Verhaltens eines Systems
- Fokus 1: auf den unterschiedlichen Zuständen, in denen sich ein System befinden kann
- Fokus 2: auf den Ergebnissen, die für einen Zustandswechsel verantwortlich sind
- Modellelemente, welche in UML-Zustandsdiagrammen genutzt werden, welche auf dem Prinzip der **endlichen Automaten** basieren:
  - Zustand und Zustandsübergang
  - Start- und Endzustand
  - Nebenläufigkeit

### LE 7 Anforderungen prüfen und abstimmen

- Ziele:
  - Anforderungen dahingehen zu überprüfen, ob sie festgelegten Qualitätskriterien genügen
  - Um etwaige Fehler in den Anforderungen möglichst frühzeitig im RE erkennen und beheben zu können
- Anforderungsdokumente = Grundlage für weitere Entwicklungsaktivitäten  
→ Fehler in den Anforderungen = überproportionale Beeinträchtigung aller weiteren Entwicklungstätigkeiten  
→ Aufwand für Behebung des Fehlers, welcher in den Anforderungen gemacht wurde und unentdeckt blieb, steigt im Verlaufe der Entwicklung erheblich an
- Grund: Nicht nur der eigentliche Fehler muss behoben werden, sondern alle darauf aufbauenden Artefakte (z.B. Architekturentwurf, Implementierung, Testfälle) ebenfalls
- Unaufgelöste Konflikte in den Anforderungen → Nichtumsetzung von Anforderungen einer Gruppe von Stakeholdern bzw. nur unzureichende Akzeptanz und Nutzung des Systems im späteren Betrieb
- Ziel der Abstimmung von Anforderungen = unter den relevanten Stakeholdern ein gemeinsames und übereinstimmendes Verständnis hinsichtlich der Anforderungen an das zu entwickelnde System zu erarbeiten
- Unterscheidung von 3 Qualitätsaspekten für Anforderungen: **Inhalt | Dokumentation | Abgestimmtheit**
- Beurteilung der Qualitätsaspekte einer Anforderung durch Prüfkriterien
- Acht Prüfkriterien für den Qualitätsaspekt **Inhalt**:
 

○ Vollständigkeit der einzelnen Anforderungen	&	des Anforderungsdokuments
○ Verfolgbarkeit	&	Korrektheit / Adäquatheit
○ Keine vorzeitigen Entwurfsentscheidungen	&	Konsistenz
○ Überprüfbarkeit	&	Notwendigkeit
- Vier Prüfkriterien für den Qualitätsaspekt **Dokumentation**:
 

○ Konformität zum Dokumentationsformat und zur Dokumentationsstruktur	
○ Konformität mit Dokumentationsregeln	
○ Verständlichkeit	& Eindeutigkeit
- Drei Prüfkriterien für den Qualitätsaspekt **Abgestimmtheit**:
  - Abstimmung
  - Abstimmung nach Änderung





- Konflikte aufgelöst
- Verschiedene **Prinzipien** für die Prüfung von Anforderungen, welche gewährleisten, dass im Zuge der Prüfung möglichst viele Fehler in den Anforderungen identifiziert werden können
- **Sechs Prinzipien in der Prüfung** von Anforderungen:
  - Beteiligung der richtigen Stakeholder
  - Trennung von Fehlersuchen und Fehlerkorrektur
  - Prüfung aus unterschiedlichen Sichten
  - Geeigneter Wechsel der Dokumentationsform
  - Konstruktion von Entwicklungsartefakten, die auf Anforderungen beruhen
  - Wiederholte Prüfung
- Verschiedene **Techniken für die systematische Prüfung** von Anforderungen, welche teilweise auch ergänzend zueinander eingesetzt werden, um Anforderungen möglichst umfassend hinsichtlich festgelegter Prüfkriterien zu prüfen
- Prüfetechniken für Anforderungen:
  - **Stellungnahme | Walkthrough | Inspektion**
- Weitere Techniken:
  - **Perspektivenbasiertes Lesen | Prüfung durch Prototypen | Einsatz von Checklisten**
- **Abstimmung von Anforderungen** → zielt darauf ab, ein gemeinsames Verständnis der Anforderungen an das zu entwickelnde System unter allen relevanten Stakeholdern herzustellen
- Aufgaben in der Abstimmung von Anforderungen:
  - **Konfliktidentifikation | Konfliktanalyse | Konfliktauflösung | Dokumentation von Konfliktlösungen**
- **Konfliktanalyse** → Unterscheidung verschiedener Konfliktarten bezüglich der Anforderungen, welche unterschiedliche Strategien der Konfliktauflösung notwendig machen
- Verschiedene Konflikttypen:
  - **Interessenskonflikt** – Stakeholder haben faktisch unterschiedliche Bedürfnisse oder persönliche Interessen
    - Objektive Konflikte → faktisch unterschiedliche Bedürfnisse
    - Subjektive Konflikte → unterschiedliche persönliche Interessen
  - **Sachkonflikt** – Stakeholder interpretieren Information unterschiedlich oder kennen die Information nicht
  - **Wertekonflikt** – Stakeholder haben unterschiedliche Werte und Präferenzen
  - **Beziehungskonflikt** – Stakeholder haben in ihren persönlichen Beziehungen untereinander Probleme
  - **Strukturkonflikt** – Der Konflikt entsteht durch unterschiedliche Hierarchie- und Entscheidungsbefugnis-Ebenen in einem Unternehmen
- In Praxis: bei auftretenden Konflikten ist die Konfliktursache häufig vermischt  
→ Für Konfliktlösung sollten alle relevanten Stakeholder berücksichtigt werden
- Konfliktlösungstechniken:
  - Einigung
  - Kompromiss
  - Abstimmung
  - Variantenbildung
  - Ober-Sticht-Unter
  - Consider-All-Facts
  - Plus-Minus-Interesting
  - Entscheidungsmatrix
- Nach Konfliktauflösung → geeignete Dokumentation des Konfliktes: Konfliktursache, beteiligte Stakeholder, Meinungen der einzelnen Stakeholder, Art der Konfliktauflösung, mögliche Alternativen, Entscheidungen, Gründe für die Entscheidungen

## LE 8 Anforderungen verwalten (K2)

- Verwaltung von Anforderungen über gesamten Lebenszyklus  
→ Infos an Anforderungen als Attribute strukturiert erfassen

- Definition der Attributstruktur für Anforderungen  
→ über Attributierungsschema, tabellarisch / in Form eines Informationsmodells definiert
- Typische Attribute:
  - Identifikator
  - Name
  - Beschreibung
  - Quelle
  - Stabilität
  - Risiko
  - Priorität
  - Rechtliche Verbindlichkeit (Zusätzliches Attribut)
- Anpassung der Attributierungsschemata an Projekt auf Basis bestimmter Rahmenbedingungen, dazu gehört:
  - Spezifische Merkmale des Projektes
  - Vorgaben seitens des Unternehmens
  - Vorschriften des Anwendungsgebiets
  - Randbedingungen des Entwicklungsprozesses
- In Praxis steigt die Anzahl Anforderungen über die Zeit
- Um diese steigende Komplexität der Anforderungsbasis für einzelne MA's beherrschbar zu halten  
→ reduzierter Zugriff / Filtern von Anforderungen in Abhängigkeit vom Verwendungszweck:
  - **Selektive Sichten** → Darstellung einer **Teilmenge der Attributwerte** von Anforderungen
  - **Verdichtende Sichten** → Darstellung verdichteter Informationen zu den über definierte Selektionskriterien ausgewählten Anforderungen
- **Priorisierung von Anforderungen** zu verschiedenen Zeitpunkten in verschiedenen Aktivitäten nach unterschiedlichen Kriterien
- Vorbereitung der Priorisierung → basierend auf einfacher Systematik:
  - Bestimmung der **Ziele und Randbedingungen** der Priorisierung
  - Bestimmung der **Priorisierungskriterien**
  - Bestimmung der relevanten **Stakeholder**
  - Auswahl der zu priorisierenden **Artefakte**
- Darauf aufbauend: Auswahl einer oder mehrerer Techniken zur Priorisierung und eigentliche Priorisierung
- Zu Priorisierungstechniken zählen:
  - Ranking und Top-Ten-Technik
  - Ein-Kriterium-Klassifikation
  - Kano-Klassifikation
  - Wieger'sche Priorisierungsmatrix
- **Verfolgbarkeit von Anforderungen** wichtig für die Verwaltung von Anforderungen  
→ Aufzeichnung, Organisation und Pflege von Verfolgbarkeitsinformationen
- Nutzen der Verfolgbarkeit bezieht sich auf:
  - Vereinfachung der **Nachweisbarkeit**
  - Identifikation von unnötigen **Systemeigenschaften**
  - Identifikation von unnötigen **Anforderungen**
  - Unterstützung der **Auswirkungsanalyse**
  - Unterstützung der **Wiederverwendung**
  - Unterstützung der Festlegung der **Zurechenbarkeit**
  - Unterstützung der **Wartung und Pflege**
- Es gibt 3 Klassen von Verfolgbarkeitsbeziehungen:
  - Pre-Requirements-Specification-Traceability
  - Post-Requirements-Specification-Traceability
  - Traceability zwischen Anforderungen
- Aufzeichnung nur solcher Informationen, für welche eine klare Verwendung existiert
- Repräsentation der Verfolgbarkeitsinformationen von Anforderungen unterschiedlich:
  - Textuelle Referenzen und Hyperlinks



- Verfolgbarkeitsmatrizen
  - Verfolgbarkeitsgraphen
- **Versionierung von Anforderungen & Konfiguration** → ermöglicht es, über den Lebenszyklus eines Systems / Produkt hinweg spezifische Entwicklungsstände von Anforderungen und -dokumenten verfügbar zu halten
- Versionsnummer einer Anforderung besitzt dabei mindestens die beiden Bestandteile **Version & Inkrement**
- **Anforderungskonfiguration** = Menge logisch zusammengehöriger Anforderungen, wobei jede Anforderung in nur in einer Version enthalten ist
- Bildung von Anforderungskonfigurationen entlang zwei Dimensionen:
  - **Produktdimension:** die einzelnen Anforderungen der Anforderungsbasis
  - **Versionsdimension:** die verschiedenen Versionsstände einer Anforderung
- Eigenschaften von Anforderungskonfigurationen:
  - **Sachlogischer Zusammenhang** der Anforderungen einer Konfiguration
  - **Konsistenz** der Anforderungen innerhalb der Anforderungskonfiguration
  - Eindeutiger **Identifikator** der Anforderungskonfiguration
  - **Unveränderbarkeit** der Anforderungen innerhalb der Anforderungskonfiguration
  - Grundlage für das **Rücksetzen** auf frühere Versionen der Anforderungsbasis
- Anforderungsbasislinie = ausgezeichnete Anforderungskonfigurationen, die stabile Versionen von Anforderungen umfassen & oftmals auch Auslieferungsstufen des Systems (→ Systemreleases) definieren
- Verwaltung von Anforderungsänderungen nötig, da sich Anforderungen über den gesamten Lebenszyklus verändern
- Änderungen werden in einem systematischen Änderungsmanagementprozess verwaltet und bearbeitet
- Darin ist das Change-Control-Board für die Bearbeitung eingehender Änderungsanträge verantwortlich
- Aufgaben des Change-Control-Boards für Änderungsanträge:
  - Klassifikation
  - Aufwandsbestimmung
  - Beurteilung hinsichtlich Aufwand / Nutzen
  - Definition neuer Anforderungen auf Basis eingehender Änderungsanträge
  - Entscheidung über Annahme oder Ablehnung
  - Priorisierung
  - Zuordnung zu Änderungsprojekten
- Typische Vertreter im Change-Control-Board: Änderungsmanager, Auftraggeber, Architekt, Nutzervertreter, Qualitätsbeauftragter, Anforderungsingenieur
- Als notwendig erachtete Änderungen von Anforderungen  
→ Dokumentation in Form von Änderungsanträgen und Übermittlung an das Change-Control-Board
- Mindestens folgende Informationen sind im **Änderungsantrag**:
  - Identifikator
  - Titel
  - Beschreibung der Änderung
  - Begründung der Notwendigkeit
  - Datum der Beantragung
  - Antragsteller
  - Priorität der Änderung aus Sicht des Antragstellers
- 3 Arten von Änderungsanträgen: **Korrektiv | Adaptiv | Ausnahmeänderung**
- Tätigkeiten im Änderungsmanagement:
  - Auswirkungsanalyse und Beurteilung
  - Priorisierung
  - Zuordnung zu einem Änderungsprojekt
  - Kommunikation der Annahme / Ablehnung
- Anforderungsmessung → Analyse der Qualität von Anforderungsdokumenten, anhand von Informationen aus Anforderungvalidierung und -verwaltung (z.B. Fehler, Attribute, Verfolgbarkeitsbeziehungen oder Änderungen)

- Darauf aufbauend: Identifikation von Verbesserungen
- Typische Messdaten: **Änderungsraten von Anforderungen & Fehler in Anforderungen**

### LE 9 Werkzeugunterstützung (K1)

- Viele Systementwicklungswerkzeuge können auch RE unterstützen, z.B. Testverwaltungs- oder Konfigurationswerkzeuge, WIKIs, Bürosoftware, Visualisierungswerkzeuge, Modellierungswerkzeuge
- Nur für RE gedacht: Requirements-Management-Werkzeug sollte folgende Eigenschaften haben:
  - Verschiedene Informationen verwalten
  - Logische Beziehungen zwischen Informationen verwalten
  - Jedes Artefakt eindeutig identifizieren
  - Informationen flexibel und sicher zugänglich machen, z.B. durch Zugriffskontrolle
  - Sichten auf die Informationen unterstützen
  - Informationen organisieren, z.B. durch Attributierung und Hierarchiebildung
  - Berichte über die Informationen erstellen
  - Dokumente aus den Informationen generieren
- Standard-Büro-SW → Unterstützung dieser Eigenschaften nur in geringem Umfang
- Spezialisierte Werkzeuge verfeinern diese, z.B. durch Verfolgbarkeitsmanagement
- **Werkzeugauswahl und -einführung erst nach Einführung von RE-Vorgehensweisen und -techniken möglich**
- Voraussetzung für Werkzeugeinführung: klare Verantwortlichkeiten und Vorgehensweisen im RE
- Folgende Gesichtspunkte sind dabei zu beachten:
  - Benötigte Ressourcen planen
  - Risiken durch Pilotprojekte umgehen
  - Evaluierung anhand von definierten Kriterien
  - Über Lizenzkosten hinausgehende Kosten berücksichtigen
  - Benutzer schulen
- Beurteilung von Werkzeugen lässt sich durch folgende sieben Sichten strukturieren:
  - Projektsicht (z.B. Unterstützung der Projektplanung)
  - Benutzersicht (insbesondere Bedienung)
  - Produktsicht (Funktionalität)
  - Prozesssicht (methodische Unterstützung)
  - Anbietersicht (z.B. Service des Anbieters)
  - Technische Sicht (z.B. Interoperabilität, Skalierbarkeit)
  - Betriebswirtschaftliche Sicht (Kosten)
- Für jede Sicht → klare Kriterien definieren