

*Grundlagen und Rahmenwerk (Buch: Kapitel 1-4)*

**Softwareintensive Systeme:** wesentliche Eigenschaften werden durch Software realisiert

- **Informationssysteme:** erfasst, speichert, transformiert, überträgt und verarbeitet Informationen
- **Eingebettete Systeme:** hohe Integration der Software mit der Hardware (z.B. ABS-System)

**Herausforderungen bei der Entwicklung softwareintensiver Systeme:**

- **Steigende Komplexität:** wachsende Anzahl an Funktionen, zunehmende Vernetzung von Systemen, Integration verschiedener Systeme, bereitzustellende Anzahl an Varianten
- **Steigender Kostendruck:** Verdrängungswettbewerb, Forderung nach niedrigen Preisen
- **Kürzere Entwicklungszeiten:** zunehmender Wettbewerb, innovative Produkte nötig
- **Softwarebasierte Innovationen:** Software rückt in den Zentrum der Produktentwicklung
- **Steigender Qualitätsanspruch:** trotz Zeit- und Kostendruck

**Einfluss auf den Projekterfolg:** 48 % Fehlschläge aufgrund mangelndem Requirements Engineering

- mangelnde Einbeziehung von Benutzern
- unvollständige sowie sich ändernde Anforderungen
- unrealistische Erwartungen
- unklare Ziele

**Steigende Kosten für die Beseitigung mangelhafter Anforderungen:**

- mangelhafte Anforderungen sind die Ursache für 50 % der Fehler im Quellcode
- frühzeitige Fehlerbeseitigung reduziert die Kosten

**Definition Requirementsanalyse:**

- (1) Requirementsanalyse ist das Ermitteln von Benutzerbedürfnissen mit dem Ziel, Anforderungen für ein System, eine Hard- oder eine Software zu definieren.
- (2) Requirementsanalyse ist das Studieren und Verfeinern von Anforderungen an ein System, an die Hard- oder an die Software.

**Definition Anforderung:**

- (1) Eine Bedingung oder Eigenschaft, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
- (2) Eine Bedingung oder Eigenschaft, die ein System oder eine Systemkomponente aufweisen muss, um einen Vertrag zu erfüllen oder einem Standard, einer Spezifikation oder einem anderen formell auferlegten Dokument zu genügen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft wie in (1) oder (2) definiert.

**Definition Anforderungsartefakt:** eine dokumentierte Anforderung

**Definition Stakeholder:** Personen oder Organisationen mit Interesse an dem geplanten System

**Definition Anforderungsquelle:** benennt den Ursprung einer Anforderung, d.h. den realweltlichen Aspekt, der für die Existenz der Anforderung ursächlich ist.

**Anforderungsarten:**

- **Funktionale Anforderung:** definiert eine vom System bzw. von einer Systemkomponente bereitzustellende Funktion oder einen bereitzustellenden Service. Als Benutzeranforderung kann eine funktionale Anforderung sehr allgemein beschrieben sein. Als Bestandteil einer Spezifikation beschreibt eine funktionale Anforderung detailliert die Eingaben und Ausgaben sowie bekannte Ausnahmen.
- **Qualitätsanforderung:** definiert eine qualitative Eigenschaft des gesamten Systems, einer Systemkomponente oder einer Funktion.
- **Nichtfunktionale Anforderungen:** sind meist unterspezifizierte funktionale Anforderungen und sollten bei einer entsprechenden Detaillierung in funktionale sowie ggf. Qualitätsanforderungen überführt werden.
- **Rahmenbedingung:** eine organisatorische oder technologische Anforderung, die die Art und Weise einschränkt, wie ein Produkt entwickelt wird. Im Allgemeinen kann zwischen Rahmenbedingungen unterschieden werden, die für das zu entwickelnde System gelten, und solchen, die den Entwicklungsprozess des Systems betreffen.

**Drei Sichten zur Beschreibung von funktionalen Anforderungen:**

- **Struktursicht:** Gliederung des Systems
- **Funktionssicht:** Strukturierung dessen, was das System leisten soll
- **Verhaltenssicht:** Zustände und Zustandsübergänge des Systems

**Problem vs. Lösung:** Der Entwicklungsprozess alterniert zwischen Problemdefinition ("Was?") und Lösungsbeschreibung ("Wie?"). Die Lösungsbeschreibung eines höheren Abstraktionsniveaus ist dabei die Problemdefinition für das jeweils niedrigere Abstraktionsniveau. Durch die Wechselwirkung zwischen Problemdefinition und Lösungsbeschreibung wird die Anzahl der möglichen Lösungen (d.h. der Lösungsraum) in jeder Phase des Entwicklungsprozesses sukzessive verringert.

**Wechselwirkungen zwischen Anforderungen und Systemarchitektur:** Die Definition der Anforderungen und der Entwurf einer adäquaten Systemarchitektur werden alternierend mit zunehmendem Detaillierungsgrad durchgeführt. Zunächst definieren Anforderungsartefakte auf einem hohen Abstraktionsniveau den Rahmen für einen ersten Entwurf einer Systemarchitektur. Erkenntnisse aus dem Architekturentwurf können dann zur Modifikation bzw. Konkretisierung der ursprünglichen Anforderungen führen, wodurch wiederum der Entwurf der Systemarchitektur konkretisiert und ggf. verbessert werden kann.

**Traditionelle Systemanalyse:** Es werden zwei Arten der Analyse und somit auch zwei Arten von Modellen unterschieden:

- (1) **Ist-Analyse:** Ermittlung von Fakten über existierende Systeme und Prozesse sowie Dokumentation in Ist-Modellen
- (2) **Soll-Analyse:** Identifizierung von Verbesserungsmöglichkeiten und Erstellung des Soll-Modells

**Definition Essenz:** Die Essenz eines Systems ist die vollständige Menge der wahren Anforderungen an das System. Eine wahre Anforderung ist eine Eigenschaft oder eine Fähigkeit, die das System haben muss, unabhängig davon, wie das System später realisiert wird.

**Definition Inkarnation:** Die Inkarnation eines Systems ist die Gesamtheit aller Elemente (z.B. Personen, PCs, Software), die dazu benutzt werden, die Essenz eines Systems zu realisieren.

#### Essenzielle Systemanalyse:

- (1) Modelliere bestehendes System → Physikalisches Ist-Modell
- (2) Leite die Essenz ab → Logisches (essentielles) Ist-Modell
- (3) Definiere gewünschte Anforderungen → Logisches (essentielles) Soll-Modell
- (4) Wähle Inkarnation aus → Physikalisches Soll-Modell

#### Vorteile der essenziellen Systemanalyse:

- Essenzielle Modelle sind stabiler und wesentlich kleiner als physikalische Modelle
- Keine Vorwegnahme von Entwurfsentscheidungen
- Alt- und Neusystem unterscheiden sich in der Regel nicht signifikant in ihrer Essenz

#### Nachteile des phasenbezogenen Requirements Engineering:

- **Keine Kontinuität:** zeitlich befristete Ausführung des RE, während der Entwicklung auftretende Änderungen werden nicht nachgehalten, somit wird die Anforderungsspezifikation für eine Wiederverwendung unbrauchbar
- **Notwendigkeit von Ist-Analysen:** da Anforderungsartefakte nicht aktualisiert werden, liegt nur eine veraltete Anforderungsspezifikation vor, so dass für jedes Folgeprojekt erneut eine zeit- und kostenaufwendige Ist-Analyse erforderlich ist
- **Keine systematische Wiederverwendung von Anforderungen:** Wiederverwendung von Anforderungen nur ad hoc
- **Produktfokussierung:** relevante Informationen für andere Produkte des Unternehmens werden durch Fokussierung auf das aktuell zu entwickelnde System ignoriert

#### Kontinuierliches Requirements Engineering:

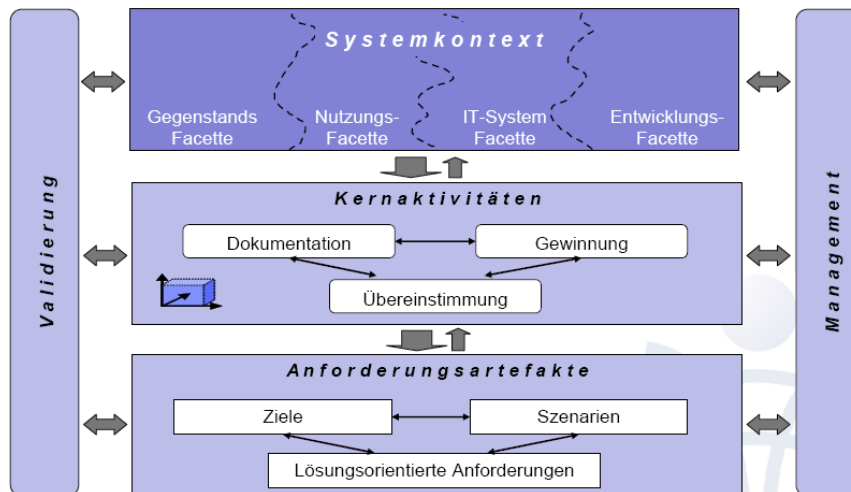
- **Phasenübergreifend:** RE als Querschnittstätigkeit, die den gesamten Entwicklungsprozess begleitet und eine konsistente und nachvollziehbare Erfassung und Verwaltung von Anforderungen gewährleistet
- **Projektübergreifend:** Anforderungen werden ständig – und somit nicht zwingend an konkrete Entwicklungsprojekte gebunden – erfasst und verwaltet → hierdurch können zu einem beliebigen Zeitpunkt die von einem neuen System umzusetzenden Anforderungen aus den bekannten aktuellen Anforderungen zusammengestellt werden

#### Vorteile des projektübergreifenden Requirements Engineering:

- **Systematischer Lernprozess:** fortlaufende Weiterentwicklung einer Wissens- und Anforderungsbasis, auf die bei der Entwicklung zukünftiger Produkte zurückgegriffen werden kann
- **Anforderungen sind immer aktuell:** Anforderungen spiegeln den aktuellen Stand der Entwicklungsprozesse sowie der Marktbedürfnisse wider
- **Kürzere Produktentwicklungszeiten:** durch Wegfall der Ist-Analyse
- **Projektübergreifende Wiederverwendung:** von Anforderungsartefakten und zugehörigen Softwareartefakten
- **Klare Zuständigkeiten:** von Personen für die Pflege und Aktualisierung der Anforderungen, dedizierte Ansprechpartner in Projekten

**Vision:** Eine Vision definiert eine angestrebte Veränderung. Sie beschreibt dabei nur das Ziel, nicht aber die Realisierung des Ziels. Sie dient als Leitgedanke für alle Stakeholder. Die in der Vision enthaltenen Informationen sind bei weitem nicht ausreichend, um die Anforderungen an das geplante System in dem erforderlichen Detailgrad zu spezifizieren. Folglich benötigen die Stakeholder zusätzliche Informationen, wobei Kunden, Systemnutzer, Domänenexperten, Dokumente und existierende Systeme als Quelle dienen können.

#### Requirements Engineering Rahmenwerk:



#### Die vier Kontextfacetten:

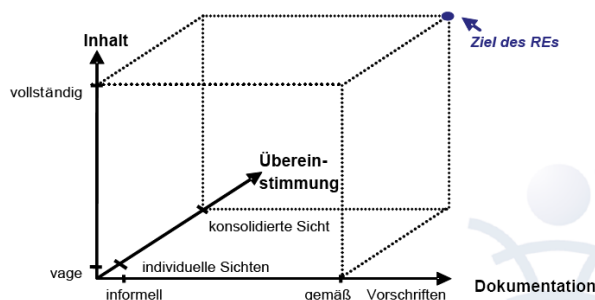
- **Gegenstandsfacette:** umfasst alle Gegenstände und Ereignisse, die ein System berücksichtigen muss, sowie alle Aspekte, die diese Abbildung beeinflussen (z.B. Gesetze, Standards)
- **Nutzungsfacette:** umfasst alle für das geplante System relevanten Aspekte der Systemverwendung, wie bspw. Nutzungsabläufe, sowie Aspekte, die die Systemverwendung einschränken oder beeinflussen
- **IT-Systemfacette:** umfasst alle Kontextaspekte, die sich aus der existierenden oder zukünftigen IT-Systemumgebung ergeben, einschließlich der existierenden IT-Strategien
- **Entwicklungsfacette:** umfasst alle Kontextaspekte, die die Entwicklung des geplanten Systems betreffen bzw. beeinflussen (z.B. Aspekte des Entwicklungsprozesses, computergestützte Entwicklungswerkzeuge, Reifegradmodelle, Qualitätszertifizierungen, Vertrauenswürdigkeit sowie Sicherheit und Zuverlässigkeit von softwareintensiven Systemen)

**Definition Requirements Engineering:** Das Requirements Engineering ist ein kooperativer, iterativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass

- (1) alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind,
- (2) die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen,
- (3) alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind.

### Die drei Dimensionen des Requirements Engineering:

- **Inhaltsdimension:** beschreibt das erreichte Verständnis über die Anforderungen an das geplante System. Ziel: Alle relevanten Anforderungen sind in dem erforderlichen Detaillierungsgrad bekannt und verstanden.
- **Übereinstimmungsdimension:** beschreibt die Übereinstimmung der Stakeholder bzgl. der Anforderungen. Ziel: Etablierung einer ausreichenden Übereinstimmung über die bekannten Anforderungen.
- **Dokumentationsdimension:** beschreibt die Art der Dokumentation von Anforderungen unter Verwendung verschiedener Dokumentationsformate. Ziel: Dokumentation aller Anforderungen konform zu den vorgegebenen Dokumentationsregeln.



**Die drei Kernaktivitäten:** Aus den drei Dimensionen des Requirements Engineering leiten sich die drei Kernaktivitäten des RE ab. Jede Kernaktivität trägt wesentlich zur Erreichung eines der drei Teilzeile des RE bei. Die Aktivitäten weisen jedoch auch zahlreiche Wechselwirkungen untereinander auf.

- **Dokumentation:** Dokumentation der Anforderungen gemäß der Dokumentationsvorschriften und Spezifikation der Anforderungen gemäß der Spezifikationsvorschriften
  - Herausforderungen: konsistente Dokumentation von Anforderungen innerhalb eines Dokumentationsformates, unterschiedliche Dokumentationsformate für unterschiedliche Verwendungszwecke, Konsistenz über verschiedene Dokumentationsformate hinweg, einheitliche Dokumentation ähnlicher Anforderungen auch über Projektgrenzen hinweg
  - Lösungsansätze: projekt- und unternehmensspezifische Festlegung von Dokumentationsformaten für ausgewählte Verwendungszwecke, Aufbau von Nachvollziehbarkeitsinformationen innerhalb einer Dokumentation und zwischen unterschiedlichen Dokumentationsformaten, Einsatz von Schablonen zur einheitlichen und strukturierten Dokumentation
- **Gewinnung:** Verbesserung des inhaltlichen Verständnisses über die Anforderungen an das geplante System sowie Gewinnung von existierenden Anforderungen von Stakeholdern und Entwicklung von innovativen Anforderungen.
  - Herausforderungen: Identifikation von Quellen, Aufdecken von verborgenen Anforderungen, Entwicklung neuer Anforderungen
  - Lösungsansätze: Strukturierte Identifikation von Quellen durch Berücksichtigung der vier Kontextfacetten, Identifikation und Verfeinerung neuer Anforderungen, Definition eines Gewinnungstechniken-Portfolios je Projekttyp
- **Übereinstimmung:** Aufdeckung und Auflösung von Konflikten unter den Stakeholdern über die bekannten Anforderungen

- Herausforderungen: Identifikation und Auflösung von (impliziten) Konflikten, Übereinstimmungsprozess projektspezifisch (abhängig von beteiligten Stakeholdern)
- Lösungsansätze: Berücksichtigung der Konfliktidentifikation und –auflösung, Techniken zur Konfliktidentifikation, Nutzung von Übereinstimmungstechniken

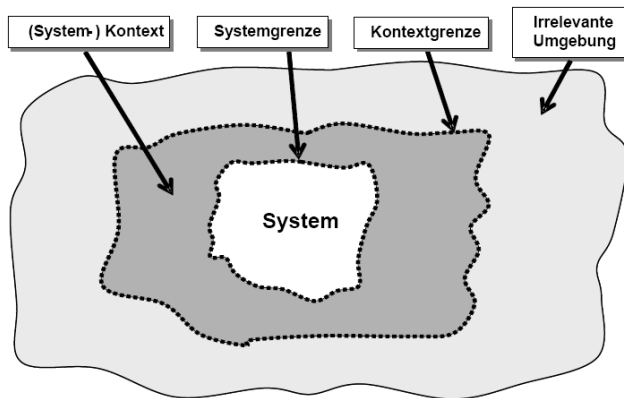
#### Die zwei Querschnittsaktivitäten:

- **Validierung:**
  - **Validierung der Anforderungsartefakte:** Aufdeckung von Fehlern in Anforderungen, jedes Anforderungsartefakt muss die Tore für den Inhalt, die Dokumentation sowie die Übereinstimmung passieren
  - **Validierung der (Kern-)Aktivitäten:** Überprüfung der Konformität der durchgeführten Aktivitäten zu Prozessvorgaben und Aktivitätsbeschreibungen
  - **Validierung der Kontextbetrachtung und -berücksichtigung:** Überprüfung der vollständigen und korrekten Erfassung aller vier Kontextfacetten
- **Management:**
  - **Management der Anforderungsartefakte:** Ablage, Priorisierung, Konfigurationsmanagement und Änderungsmanagement der Anforderungen
  - **Management der Aktivitäten:** Planung, Steuerung und Kontrolle der Aktivitäten
  - **Beobachtung des Systemkontextes:** Identifizierung von etwaigen Veränderungen des Kontextes sowie wenn notwendig Initiierung von zusätzlichen Aktivitäten, um die Anforderungen an den veränderten Systemkontext anzupassen

**Wechselwirkung zwischen den Aktivitäten:** Die fünf Aktivitäten des Rahmenwerks beeinflussen sich gegenseitig. Fortschritte in einer Dimension können dabei in einer anderen Dimension neue offene To-Dos erzeugen und somit den Fortschritt in einer anderen Dimension mindern. Daher können die einzelnen Aktivitäten im Requirements Engineering für gewöhnlich nicht vollkommen isoliert voneinander durchgeführt werden.

#### Die drei Arten von Anforderungsartefakten:

- **Ziele:** dokumentieren die Intentionen der Stakeholder und abstrahieren dabei sowohl von der Nutzung des Systems als auch von Aspekten der Systemrealisierung. Ziele verfeinern die Vision auf die Ebene einzelner, charakteristischer Merkmale des geplanten Systems. Häufig sind Ziele hierarchisch strukturiert, wobei an der Wurzel der Hierarchie die Vision steht.
- **Szenarien:** beschreiben oder bewerten exemplarisch konkrete (positive und negative) Beispiele für Interaktionsfolgen zur Erfüllung (oder Nichterfüllung) eines oder mehrerer definierter Ziele, häufig in Form von Use Cases. Ziele und Szenarien verhalten sich komplementär zueinander, da Ziele die Gewinnung von Szenarien stimulieren und andersrum. Szenarien selbst können sich widersprechen und können in allen RE-Aktivitäten verwendet werden. Sie unterstützen zudem die Kommunikation zum Stakeholder.
- **Lösungsorientierte Anforderungen:** definieren die Daten, Funktionen und das Verhalten eines softwareintensiven Systems. Darüber hinaus beinhalten sie auch Qualitätsanforderungen. Im Gegensatz zu Zielen und Szenarien, die im Wesentlichen unabhängig von einer intendierten Lösung definiert werden können, setzt die Definition von lösungsorientierten Anforderungen die Annahme einer bestimmten angestrebten Lösung voraus.

*Systemkontext (Buch: Kapitel 5-6)*

**Definition Systemkontext:** Der Systemkontext ist der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen an das System relevant ist. Der Systemkontext besteht aus den vier Kontextfacette Gegenstandsfacette, IT-Systemfacette, Nutzungsfacette und Entwicklungsfacette.

**Definition Kontextaspekt:** Kontextaspekte sind materielle und immaterielle Bestandteile des Systemkontexts, wie z.B. Personen, Systeme, Prozesse oder physikalische Grundgesetze.

**Definition Systemgrenze:** Die Systemgrenze separiert das geplante System von seiner Umgebung. Sie grenzt das System von den Teilen der Umgebung ab, die durch den Entwicklungsprozess nicht verändert werden können.

**Veränderbare und konstante Artefakte:** Materielle und immaterielle Objekte, die innerhalb der Systemgrenze liegen, sind bei der Systementwicklung gestaltbar und somit veränderbar. Artefakte außerhalb der Systemgrenze sind vom Systementwicklungsprozess nicht gestaltbar und werden somit als unveränderbar angesehen.

**Anforderungen existieren nur im Kontext:** Erst die Festlegung der Systemgrenze und der Kontextgrenze ermöglicht eine korrekte Definition von Anforderungen. Anders formuliert: Anforderungen existieren nur mit Kontextbezug und können nur unter Kenntnis der Kontexts definiert und korrekt interpretiert werden.

**Quellen und Senken:** Zur Abgrenzung des Systems von dessen Umgebung können u.a. die Quellen und Senken des Systems betrachtet werden. Quellen liefern Eingaben an das System, Senken erhalten Ausgaben vom System. Beispiele für Quellen und Senken sind Menschen, andere Systeme, Sensoren und Aktuatoren.

**Grauzone der Systemabgrenzung:** Die Systemschnittstellen sowie die Systemgrenze sind typischerweise erst gegen Ende des RE-Prozesses stabil. Zuvor sind einige oder mehrere Schnittstellen oft nicht detailliert beschreiben bzw. noch gar nicht bekannt, da z.B. zu Beginn noch nicht alle Funktionalitäten des Systems festgelegt sind. Aufgrund dieser zunächst unscharfen Abgrenzung des Systems von seiner Umgebung existiert eine Grauzone, in der die zu einem bestimmten Zeitpunkt bekannten, potenziellen Verläufe der Systemgrenze enthalten ist. Neben einer etwaigen Verschiebung der Systemgrenze innerhalb der Grauzone kann während des RE auch die Grauzone selbst verschoben werden, z.B. wenn Objekte, die aktuell zur Umgebung gehören, verändert werden sollen.



**Definition Kontextgrenze:** Die Kontextgrenze separiert den relevanten Teil der Umgebung eines Systems von dem irrelevanten Teil, d.h. dem Teil, der keinen Einfluss auf die Systementwicklung hat und daher im Requirements Engineering nicht betrachtet wird.

**Grauzone zwischen Systemkontext und irrelevanter Umgebung:** Zu Beginn des RE-Prozesses kennen die meisten Stakeholder nur einen kleinen Teil der Umgebung und deren Einfluss auf das geplante System. Die Kontextgrenze ist daher zu Beginn nur äußerst lückenhaft beschrieben. Im Projektverlauf werden die Grenzen zwischen Systemkontext und irrelevanter Umgebung immer besser verstanden und die identifizierten Objekte entweder dem Kontext oder der irrelevanten Umgebung zugeordnet. Da die Kontextgrenze den Systemkontext von dem gesamten restlichen Universum abgrenzt, ist eine vollständige und präzise Kontextabgrenzung für komplexe softwareintensive Systeme praktisch nicht erreichbar. Dadurch entsteht auch bei der Kontextabgrenzung eine Grauzone, die identifizierte aber unklassifizierte Objekte enthält, deren Rolle in der Systementwicklung aktuell nicht genau bekannt ist.

**Kontext einer Anforderung:** wird durch zwei Bereiche bestimmt:

- Die Beziehung der Anforderung zu verschiedenen Kontextobjekten
- Die Beziehung der Anforderung zu anderen Anforderungen sowie deren Beziehungen zu verschiedenen Kontextobjekten

**Strukturierungsprinzipien:**

- Unterteilung des Kontexts in Teilbereiche, die sogenannten Kontextfacetten
  - zur Fokussierung auf Teilbereiche → Komplexitätsreduktion
  - zur Überprüfung auf Vollständigkeit
- Klassifikation der Kontextaspekte in den jeweiligen Kontextfacetten
  - Verwendung einer einheitlichen Schablone zur Erläuterung
  - zur Berücksichtigung verschiedener Aspekte und zur Unterscheidung zwischen Beziehungen zwischen Kontextaspekten und Anforderungen

**Drei Typen von Kontextaspekten:**

- **Anforderungsquellen:** der Ursprung aller Anforderungen an das geplante System
  - **Stakeholder:** verfügt typischerweise über Wissen über einen oder mehrere Kontextaspekte aus einer oder mehreren Kontextfacetten. Ein Stakeholder muss jedoch nicht direkt mit dem zukünftigen System in Beziehung stehen, sondern kann bspw. auch übergeordnete Interessen bei der Systementwicklung vertreten und somit Einfluss auf die Anforderungen des Systems nehmen (z.B. Datenschutzbeauftragter, Rechtsanwälte, IT-Strategieverantwortliche oder Betriebsratsvertreter)
  - **Dokument:** zahlreiche Informationen, aus denen Anforderungen für das System gewonnen werden, sind nur in Form von allgemeingültigen oder organisationsspezifischen Dokumenten verfügbar. Ein Beispiel für die Gewinnung von Anforderungen aus Dokumenten ist die Wiederverwendung von Entwicklungsartefakten eines Altsystems.
  - **System:** es wird zwischen Alt- und Vorgängersystemen, Konkurrenzsystemen und Systemanalogien (Systeme, die für eine andere Domäne entwickelt wurden als das geplante System, die jedoch für das geplante System wichtige Aspekte besitzen) unterschieden. Durch die Analyse bzw. das Experimentieren mit einem existierenden System erhalten



die Stakeholder einen unmittelbaren Eindruck bereits realisierter Anforderungen. Basierend darauf werden Anforderungen definiert, die versuchen, Eigenschaften eines existierenden Systems nachzubilden und zu erweitern, oder die Mängel und bereits behobenen Fehler in Alt- und Vorgängersystemen zu vermeiden.

- **Betrachtungsgegenstände:** natürliche und juristische Personen sowie materielle und immaterielle Objekte aus den verschiedenen Kontextfacetten, die während des RE berücksichtigt werden müssen und über die die Stakeholder folglich Informationen gewinnen müssen. Für alle vier Systemfacetten müssen die für die Systementwicklung relevanten Betrachtungsgegenstände identifiziert werden. Anforderungsquellen sind teilweise ebenfalls Betrachtungsgegenstände des Systems (z.B. Systemnutzer, über den Informationen gespeichert und verarbeitet werden).
- **Eigenschaften und Beziehungen der Betrachtungsgegenstände:** erlauben eine genauere Charakterisierung der Betrachtungsgegenstände in einer Kontextfacette, da die Identifikation der Betrachtungsgegenstände in der Regel nicht ausreicht, um Anforderungen an das geplante System abzuleiten. Beziehungen existieren sowohl zwischen den Betrachtungsgegenständen einer Kontextfacette als auch zwischen Gegenständen unterschiedlicher Kontextfacetten.

**Definition Stakeholder:** Ein Stakeholder ist eine Person oder eine Organisation, die ein potenzielles Interesse an dem zukünftigen System hat und somit in der Regel auch Anforderungen an das System stellt. Eine Person kann dabei die Interessen von mehreren Personen oder Organisationen vertreten, d.h. mehrere Rollen einnehmen.

#### Gegenstandsfacette:

- Anforderungsquellen:
  - Domänenexperten: geben Auskunft über den Gegenstandsbereich des geplanten Systems sowie die im System abzubildenden Betrachtungsgegenstände
  - Juristen und Datenschützer: welche Dateien müssen (anonymisiert) gespeichert werden, wie lange müssen die Daten aufbewahrt werden
  - Referenzmodelle des Gegenstandsbereichs: beinhalten bereits zahlreiche relevante Betrachtungsgegenstände, Eigenschaften und Beziehungen
  - Fachbücher und Gesetzestexte
  - Existierende Systeme: z.B. Analyse von Ein- und Ausgabemasken
- Betrachtungsgegenstände:
  - Personen, über die Daten gespeichert werden (z.B. Kunden, Lieferanten)
  - materielle/immaterielle Objekte (z.B. Produktionsgüter/Temperatur)
  - Prozesse (z.B. Produktions- oder Geschäftsprozesse, die abgebildet werden sollen)
- Eigenschaften und Beziehungen der Betrachtungsgegenstände:
  - Eigenschaften der identifizierten Betrachtungsgegenstände (z.B. Reifendruck, Abstand des vorausfahrenden Fahrzeugs)
  - Abbildungsfunktion, welche die Präzision und die Aktualität der Abbildung festlegt und evtl. bestehende gesetzliche Vorgaben beachtet
  - Präzision der Abbildung: beschreibt, wie exakt der reale Gegenstand im System abgebildet wird und umfasst sowohl die Genauigkeit numerischer Daten als auch den Abstraktionsgrad der Abbildung

- Aktualität der Abbildung: definiert die Art und Weise sowie die Intervalle, in denen im System abgebildete Eigenschaften aktualisiert werden
- Gesetzliche Vorgaben: können die Abbildung sowie die Aktualisierung von Betrachtungsgegenständen reglementieren

**IT-Systemfacette:**

- Anforderungsquellen:
  - Personen, die sich mit der Planung, dem Entwurf und dem Betrieb der IT-Systemumgebung befassen (z.B. Systemarchitekten, Hardwareentwickler)
  - Technologieberater und Zulieferer: besitzen Informationen über Markttendenzen und Strategien in Bezug auf relevante Hard- und Softwarekomponenten
  - Relevante Dokumente: IT-Strategiedokumente der eigenen Unternehmung oder des Kunden, Infrastrukturdokumente
  - Referenzarchitektur: definiert u.a. eine gemeinsame Struktur für eine Familie von softwareintensiven Systemen
  - Existierender Systeme: erfordert Analyse des Aufbaus und der Komponenten des Systems
- Betrachtungsgegenstände:
  - Hard- und Softwarekomponenten
  - IT-Strategien (z.B. Verwendung bestimmter Hersteller oder Technologien, Berücksichtigung von Standards oder Einhaltung von Entwicklungsprozessen gewissen Reifegrads)
  - Betrieb und Wartung (z.B. wann und in welcher Weise Datensicherungen stattfinden müssen)
- Eigenschaften und Beziehungen der Betrachtungsgegenstände:
  - technische Daten von Komponenten (z.B. Leistungsdaten, Ausfallraten, Kosten)
  - Betrieb (Verfügbarkeit des Systems, Profile der erwarteten Netzwerklast)

**Nutzungsfacette:**

- Anforderungsquellen:
  - Direkte Nutzer: haben bestimmte Anforderungen an die Benutzungsschnittstelle
  - Indirekte Nutzer: nutzen das System nicht selber, üben aber Einfluss auf die Nutzung aus oder profitieren indirekt von der Nutzung
  - Experten für Benutzungsschnittstellen: unterstützen innovative Anforderungen
  - Standards, Gesetze und Vorschriften: befassen sich mit Qualitätsaspekten der Benutzungsschnittstelle oder mit erlaubten Nutzungsabläufen
  - Domänenmodelle: legen u.a. die erlaubten Nutzungsabläufe fest und betten die Nutzung in den umgebenden Kontext ein (z.B. Geschäftsprozessmodelle)
  - Systemanalogien: existierende Systeme, die auf vergleichbare Weise verwendet werden
- Betrachtungsgegenstände:
  - Nutzergruppen: Kenntnis der Wünsche und Bedürfnisse wichtiger Faktor
  - Typ der Benutzerschnittstelle (z.B. Touchscreen, Tastatur oder Sprache)
  - Nutzungsabläufe: Kernaspekt der Nutzungsfacette, nicht nur Prozesse sind Betrachtungsgegenstände, sondern auch die mit den Abläufen assoziierten Objekte (z.B. Rollen, Tätigkeiten und Zuständigkeiten) sowie der erwartete Mehrwert der Nutzung

- Interaktion mit anderen Systemen: Mehrwert für andere technische oder softwarebasierte Systeme ebenfalls sehr wichtig, da sich daraus Anforderungen an die erforderlichen Schnittstellen ergeben
- Eigenschaften und Beziehungen der Betrachtungsgegenstände:
  - Beziehungen zwischen Nutzungsabläufen, da sie oft in wechselseitiger Beziehung zueinander stehen
  - Beziehung zwischen Nutzergruppe und Nutzungsablauf (z.B. Einschränkungen)
  - Beziehung zur IT-Systemfacette (z.B. Zuordnung zu verschiedenen Betriebsarten)
  - Beziehung zur Gegenstandsfacette: das System bearbeitet Betrachtungsgegenstände und deren Eigenschaften aus der Gegenstandsfacette und gibt sie in aufbereiteter Form an den Systemnutzer weiter. Dieser muss in der Lage sein, die angezeigten Informationen korrekt zu interpretieren und mit den Gegenständen zu assoziieren.

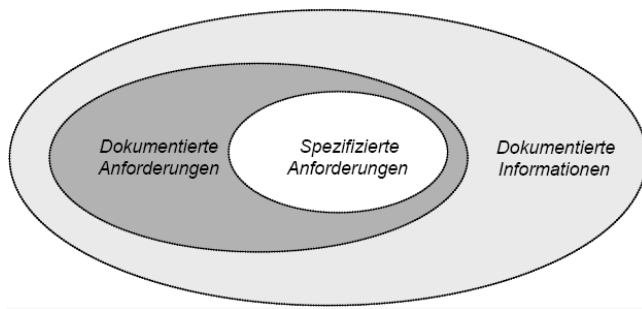
#### Entwicklungsfacette:

- Anforderungsquellen:
  - Prozess-Stakeholder (Prozessingenieure, -manager, -ausführende)
  - Dokumente, die Vorgaben an oder Informationen über den Entwicklungsprozess beinhalten (Entwicklungsstandards und –richtlinien, Best Practices, alte Projektpläne)
- Betrachtungsgegenstände sowie deren Eigenschaften und Beziehungen:
  - Ressourcen: bestimmter Einführungszeitpunkt, zur Verfügung stehende Anzahl und Qualifikationen der Entwickler, finanzielle Ausstattung
  - Entwicklungsumgebungen: Vorschrift von bestimmten Werkzeugen und Sprachen für Entwicklungsaktivitäten
  - Koordination von Entwicklungsprozessen, wenn mehrere Unternehmen ein komplexes System entwickeln
  - Schnittstellen von Entwicklungsprozessen: Bereitstellung der richtigen Artefakte zum richtigen Zeitpunkt in benötigter Qualität
  - Prozessqualität: Pflicht zur Einhaltung von Qualitätsstandards durch den Auftraggeber

#### *Kernaktivität Dokumentation (Buch: Kapitel 15-18)*

##### Vorteile der Dokumentation:

- Persistenz von gewonnen Informationen
- Gemeinsame Informationsbasis, auf die alle Projektbeteiligten zurückgreifen können
- Fördert Kommunikation zwischen den Stakeholdern
- Fördert Objektivität, da die Inhalte stabiler als beim verbalen Austausch sind
- Eigenständige Einarbeitung neuer Mitarbeiter wird erleichtert
- Vermindert Abhängigkeit von Wissensträgern in Projekten
- Reflexion mit den Inhalten vermeidet Unvollständigkeiten oder Fehler

**Dokumentation vs. Spezifikation:**

- Allgemeine Dokumentationsvorschriften: umfassen Vorgaben über den Aufbau und die Erstellung von Protokollen, Struktur und Inhalt von Änderungsanträgen etc.
- Dokumentationsvorschriften für Anforderungen: legen fest, wie Anforderungen abhängig vom jeweiligen Verwendungszweck zu dokumentieren sind (Dokumentationsformen, Abstraktions- und Detaillierungsgrad, erlaubter Umfang der Abweichung)
- Spezifikationsvorschriften für Anforderungen: legen Merkmale und Eigenschaften spezifizierter Anforderungen fest (häufig Verwendung von testbaren Qualitätskriterien)

**Qualitätskriterien für Anforderungsartefakte:**

- Vollständigkeit einzelner Anforderungen und Vollständigkeit des Anforderungsdokuments
- Nachvollziehbarkeit bezogen auf Ursprung und Evolution der Anforderungen sowie deren Einfluss auf die Realisierung des Systems
- Korrektheit muss von den Stakeholdern bestätigt werden (verhindert Gold-Plating)
- Eindeutigkeit, so dass Anforderungen nur eine Interpretation zulassen
- Verständlichkeit, so dass den Inhalte der Anforderung leicht erfasst werden kann
- Konsistenz schließt sich widersprechende und konfliktär zueinander stehende Anforderungen aus
- Überprüfbarkeit durch Definition von Abnahmekriterien
- Bewertet, in dem Bedeutung und/oder die Stabilität der Anforderungen ermittelt wird
- Aktualität, in dem sie die aktuellen Gegebenheiten des zu entwickelnden Systems und des dazugehörigen Kontexts widerspiegelt
- Atomarität, so dass die Anforderung einen isolierten Sachverhalt beschreibt

**Abnahmekriterien für Anforderungsartefakte:** legen fest, welche Bedingungen ein Anforderungsartefakt erfüllen muss, damit es erfolgreich abgenommen wird.

- **für einzelne Anforderungsartefakte:** können unabhängig von anderen Anforderungsartefakten angewendet werden und konkretisieren Dokumentationsvorschriften und Qualitätskriterien durch die Vorgabe detaillierter Regeln bzw. Prüfanweisungen (meist in Form von Checklisten)
- **für Anforderungsdokumente:** konkretisieren Dokumentationsvorschriften und Qualitätskriterien für das gesamte Anforderungsdokument, indem sie z.B. messbare, quantitative Vorgaben an die Vollständigkeit des Dokuments definieren

**Abnahmekriterien für das System:** legen fest, unter welchen Bedingungen sich der Auftraggeber verpflichtet, das realisierte System abzunehmen. Sie beinhalten Anforderungen für die Ableitung von Testfällen sowie die Vorgabe von Ein- und Ausgaben.

- **für einzelne funktionale Eigenschaften und Qualitätseigenschaften:** definieren Vorschriften zur Überprüfung des Systems oder einer Systemkomponente gegen einzelne funktionale oder Qualitätsanforderungen. Sie dienen dem Nachweis der in den Anforderungen spezifizierten funktionalen und Qualitätseigenschaften des Systems.
- **für das Gesamtsystem:** Gegenstand sind Eigenschaften des Systems, die sich aus dem Zusammenwirken von mehreren im Anforderungsdokument spezifizierten Anforderungen an das System ergeben. Die Abnahmekriterien für das Gesamtsystem können sich auf die für die einzelnen funktionalen und Qualitätseigenschaften definierten Abnahmekriterien beziehen.

**Definition Lastenheft:** Das Lastenheft enthält eine Definition der Systemvision, eine Beschreibung der wesentlichen Systemziele (Funktionen und Qualitäten) und benennt wichtige Kontextaspekte (z.B. Rahmenbedingungen) der vier Kontextfacetten sowie ihrer Beziehungen zur Vision und zu den definierten Systemzielen.

**Verwendung des Lastenhefts:**

- Erstellung des Pflichtenhefts
- Ausschreibungen bzw. Vertragsverhandlungen
- Bewertung alternativer Realisierungsmöglichkeiten
- Risikoabschätzung

**Definition Pflichtenheft:** Das Pflichtenheft detailliert die im Lastenheft beschriebene Vision und die Systemziele (abstrakte Funktionen und Qualitäten) sowie ggf. im Lastenheft definierte Rahmenbedingungen im Hinblick auf die angestrebte technische Umsetzung (Realisierung) des Systems.

**Verwendung des Pflichtenhefts:**

- Projektplanung: Ableitung von Arbeitspaketen und Meilensteinen aus dem Pflichtenheft
- Architekturentwurf, Implementierung
- Tests: Ableitung von Testfällen
- Systemabnahme: Definierung von Abnahmekriterien
- Vertragsmanagement: Nachverhandlungen bei Änderungen im Pflichtenheft
- Änderungsmanagement: zur Analyse der Auswirkungen bei Anpassungen
- Systemnutzung und -wartung

**Qualitätskriterien für Anforderungsdokumente:**

- **Vollständigkeit:** jede einzelne Anforderung muss vollständig spezifiziert und alle relevanten Anforderungen müssen im Dokument enthalten sein
- **Konsistenz:** jede einzelne Anforderung muss konsistent sein und es dürfen keine Konflikte zwischen Anforderungen im Anforderungsdokument existieren
- **Änderungsfreundlichkeit und Lesbarkeit:** werden durch Struktur und Layout des Dokuments bestimmt. Es muss zum einen eine einfache, vollständige und konsistente Modifikation der Anforderungen gewährleistet sein. Zum anderen muss der Leser die Inhalte möglichst einfach erfassen können.

**Vorteile der natürlichsprachlichen Dokumentation:**

- Universell: in beliebigen Domänen einsetzbar
- Flexibel: ermöglicht beliebige Abstraktionen und Detaillierungen
- Leicht anwendbar: keine Schulungen oder spezielle Werkzeuge notwendig

**Nachteile der natürlichsprachlichen Dokumentation:**

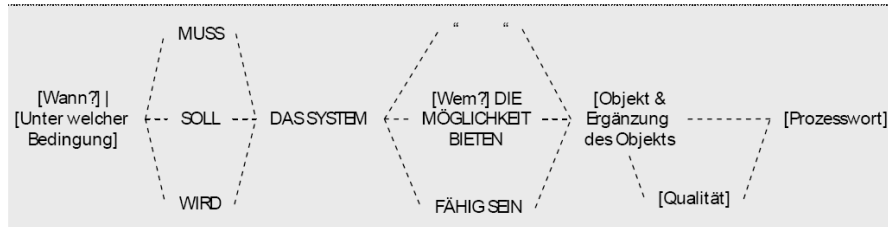
- Mehrdeutigkeit aufgrund von Unterspezifikation: fehlende inhaltliche Details lassen ein Spektrum möglicher Interpretationen zur Vervollständigung der Anforderungen zu
- Mehrdeutigkeit aufgrund von Defekten in der natürlichen Sprache: lexikalische, syntaktische, semantische und referentielle Mehrdeutigkeit sowie Vagheit und Generalität von Begriffen
- **Lexikalische Mehrdeutigkeit:**
  - Synonym: ein Wort, das trotz unterschiedlicher Benennung die gleiche Bedeutung wie ein anderes Wort besitzt (Streichholz – Zündholz)
  - Homonym: ein Wort, das trotz gleichartiger Benennung in unterschiedlichen Kontexten verschiedenartige Bedeutungen besitzt (Bank, Steuer)
- **Syntaktische/Strukturelle Mehrdeutigkeit:**
  - eine Wortfolge kann mehr als einer grammatikalischen Struktur zugeordnet werden und die Wortfolge weist dabei verschiedene Bedeutungen auf (der Benutzer gibt seine Zugangskarte mit dem Zugangscode ein)
- **Semantische Mehrdeutigkeit:**
  - im gegebenen Kontext kann ein Satz verschiedenartig interpretiert werden, auch wenn keine lexikalische oder syntaktische Mehrdeutigkeit vorliegt (z.B. bei mehrfacher Verwendung von „und“ und „oder“). Lexikalische und syntaktische Mehrdeutigkeiten führen ebenfalls zu semantischen Mehrdeutigkeiten.
- **Referentielle Mehrdeutigkeit:**
  - für eine Anapher (referentieller Rückwärtsverweis) oder eine Katapher (referentieller Vorwärtsverweis) kann die zugehöriger Antezedenz (referenzierter Ausdruck) für den Interpretierenden nicht eindeutig bestimmt werden (Der Kunde führt die Zugangskarte in den Kartenleser ein und gibt anschließend seine PIN ein. Ist *diese* nicht korrekt, wird der Zutritt verweigert).
- **Vagheit und Generalität von Begriffen:**
  - Vagheit: Extensionsmenge nicht exakt bekannt (Mittelklassewagen)
  - Generalität: Umfang des Begriffs kann nicht genau bestimmt werden

**Techniken zur Vermeidung von Mehrdeutigkeiten:**

- **Glossare:**
  - Definition: Ein Glossar ist eine Sammlung von Fachbegriffen, die Bestandteil einer Sprache (Terminologie) sind. Das Glossar legt für diese Fachbegriffe deren spezifische Bedeutung fest. Ein Glossar kann zudem Verweise auf verwandte Begriffe sowie Beispiele zur Erläuterung der Begriffe umfassen.
  - Vorschlag für eine Struktur: Begriff, Definition, Synonyme, Verwandte Begriffe (Generalisierungen, Spezialisierungen), Beispiele

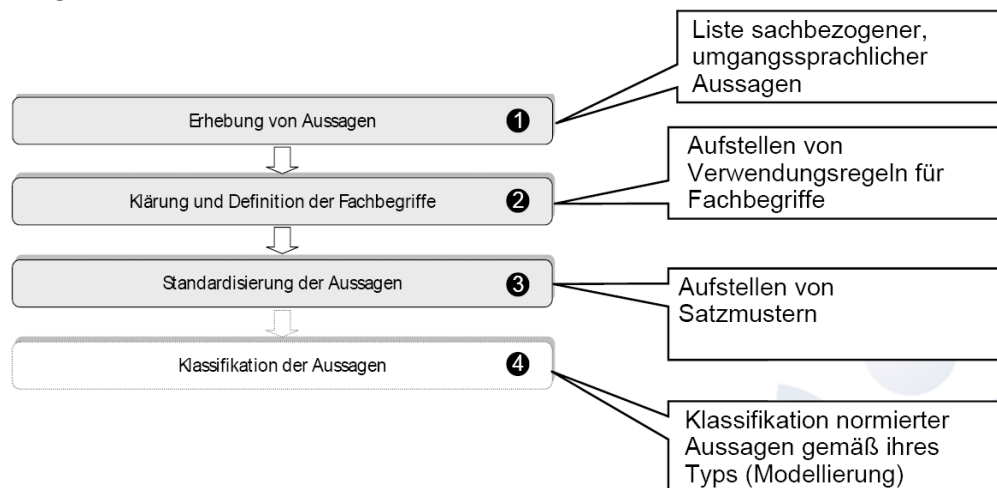
- **Syntaktische Anforderungsmuster:**

- Definition: Ein syntaktisches Anforderungsmuster dokumentiert wiederverwendbares Wissen über die Formulierung natürlichsprachlicher Anforderungen, indem es syntaktische Strukturen für die Dokumentation von natürlichsprachlichen Anforderungen vorgibt.



- **Normsprachen:**

- Definition: Eine Normsprache besitzt eine in Bezug auf eine spezifische Domäne eingeschränkte natürlichsprachliche Grammatik (Syntax) und definiert eine Menge von Begriffen (Semantik), die unter Verwendung der vorgegebenen Grammatik zur Konstruktion von Aussagen über die Domäne verwendet werden können.
- Normsprachen können als Erweiterung zu syntaktischen Anforderungsmustern gesehen werden, die neben den syntaktischen Strukturen von Anforderungen auch teilweise deren Semantik festlegen bzw. einschränken.
- Vorteile: normsprachliche Aussagen sind einfach verständlich, besitzen eine höhere Eindeutigkeit als in natürlicher Sprache verfasste Aussagen und sind aufgrund der formalen Grammatik semantisch überprüfbar
- Vorgehensweise:



**Vorteile von Referenzstrukturen für Anforderungsdokumente:** Die Verwendung von Referenzstrukturen erhöht die Änderbarkeit und Lesbarkeit eines Dokuments und ermöglicht ein einfacheres Überprüfen des Anforderungsdokuments auf Vollständigkeit. Darüber hinaus bietet der Einsatz einer Referenzstruktur zur Gliederung von Anforderungsdokumenten eine Reihe weiterer Vorteile:

- Erprobte Strukturen durch wiederverwendetes Expertenwissen
- Referenz für Vollständigkeit
- Konzentration auf die Inhalte statt Fokus auf Form oder Strukturierung
- Gleiche Informationen am gleichen Ort für Dokumentierende und Benutzer
- Werkzeugunterstützung



**Attributierung von Anforderungen:** dient der strukturierten Dokumentation von Informationen, die während des RE-Prozesses in Bezug auf eine Anforderung anfallen (z.B. Status, Quelle, Priorität).

**Definition Anforderungsattribut:** Ein Anforderungsattribut wird durch einen Attributnamen, eine zugehörige Attributsemantik sowie den Wertebereich des Attributs und der jeweiligen Wertesemantik definiert.

**Definition Attributierungsschema:** Ein Attributierungsschema definiert die Attribute für eine Klasse von Anforderungen. Für jedes Attribut legt das Attributierungsschema einen eindeutigen Attributnamen, die Attributsemantik, den Wertebereich und die Wertesemantik fest.

**Attribute von Anforderungsartefakten:**

- Identifizierbarkeit (Bezeichner, Name)
- Kontextbeziehungen (Quelle, Kontextfacette, Begründung, Verantwortlicher)
- Dokumentationsaspekte (Dokumentationsformate, -regeln, Validierungsstatus)
- Inhaltsaspekte (Anforderungstyp, Kurzbeschreibung, Querbezüge)
- Übereinstimmungsaspekte (Status der Übereinstimmung sowie der Validierung, Konflikte)
- Validierungsaspekte (Konformität zu Eingangskriterien, Validierungstechniken)
- Managementaspekte (Risiko, Kritikalität, Priorität, Stabilität, Version, Aufwand, Kosten)

**Schablonen und Informationsmodelle:** Die vorgestellten Attribute für Anforderungen können auf verschiedene Arten für die strukturierte Dokumentation von Informationen verwendet werden: Die zwei wichtigsten Arten sind dabei:

- **Schablonenbasierte Dokumentation natürlichsprachlicher Anforderungen:** Durch die Bereitstellung von Schablonen, die basierend auf den Attributen erstellt werden, können textuelle Anforderungen einfach und effektiv in einer strukturierten Form dokumentiert werden. Sie bringen folgende Vorteile:
  - Festlegung der benötigten Informationen (zulässige Attributwerte, Semantik)
  - Erkennung von Lücken aufgrund von leeren Bereichen in der Schablone
  - Einarbeitung von Mitarbeitern (gezielte Suche, Wichtigkeit von Informationen)
  - Gleiche Informationen an gleicher Stelle: dadurch erleichterter Vergleich von Informationen sowie die Möglichkeit des selektiven Zugriffs durch den Leser
- **Informationsmodellbasierte Dokumentation natürlichsprachlicher Anforderungen:** erweitern die Attribute der Schablonen um die Beziehungen zwischen den Attributen. Die Modelle werden unter Verwendung einer Modellierungssprache erstellt (z.B. UML).
  - Grundlage für die Werkzeugunterstützung: häufig Basis für den Entwurf eines relationalen Datenbankschemas zur Ablage und Verwaltung von Anforderungsartefakten zusammen mit ihren Attributen und Beziehungen
  - Verfeinerung von Informationsmodellen durch Hinzufügen von Konsistenzbedingungen möglich, Überprüfung sollte idealerweise automatisiert über der RE-Werkzeug laufen



**Definition Modell:** Ein Modell ist ein abstrahierendes Abbild einer existierenden oder fiktiven Realität.

- **Physikalisches Modell:** werden in den klassischen Ingenieursdisziplinen verwendet, um im Vorhinein wichtige Aussagen und Erkenntnisse über den zu konstruierenden Gegenstand zu erhalten
- **Grafisches Modell:** fördern die Kommunikation, da sie in der Regel einfacher zu verstehen sind als die Dokumentation der Sachverhalte in natürlichsprachlichen Texten. In der Software Entwicklung werden sie als **konzeptuelle Modelle** bezeichnet, die vor allem dazu dienen, spezifische Aspekte des Gegenstandsbereichs anschaulich zu visualisieren und die Komplexität der Betrachtung zu reduzieren.

- **Verkürzung:** Die verkürzende Eigenschaft von Modellen beruht auf den zwei Abstraktionsmechanismen Selektion (nur bestimmte Aspekte) und Verdichtung (Zusammenfassung von Aspekten), die mit dem Ziel der Komplexitätsreduzierung eingesetzt werden. Präterierte Attribute werden nicht mit in das Modell übernommen.
- **Erweiterung:** Es werden im Modell Aspekte dargestellt, die im zugrunde liegenden Gegenstandsbereich in dieser Form nicht vorhanden sind (abundante Attribute). Z.B. werden bei Straßenkarten häufig Längen- und Breitengrade hinzugefügt oder Straßen unterschiedlich eingefärbt.
- **Pragmatisches Merkmal:** Subjektbezug, Zeitbezug, Problembezug

**Vorteile konzeptueller Modelle:**

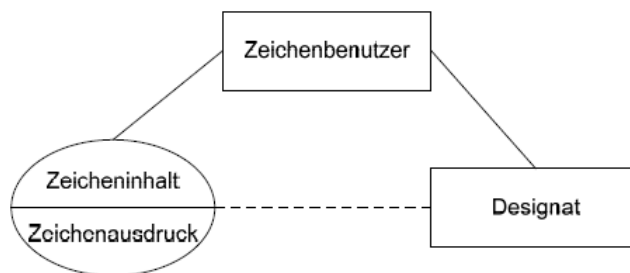
- ermöglichen Abstraktion auf das Wesentliche mit unterschiedlichen Detaillierungsstufen
- helfen Komplexität zu bewältigen
- fördern die Kommunikation(!) und dienen als Diskussionsgrundlage
- sind günstiger als Prototypsysteme

**Methoden:**

- sind planmäßig angewandte, begründete Vorgehensweisen zur Erreichung von festgelegten Zielen (im allg. im Rahmen festgelegter Prinzipien)
- enthalten den Weg zu etwas hin, d.h. sie machen etwas anwendbar
- sollen konkret sein, im Sinne der Wegleitung, einer Aufteilung in Arbeitsschritte
- Anwendung und Erfolg einer Methode sollen verifizierbar und messbar sein

**Merkmale eines Zeichens:**

- ein Zeichen besteht aus Inhalt und Ausdruck
- die Zeichenbenutzung besteht aus drei Komponenten: das Zeichen selbst, das Designat (worauf sich das Zeichen bezieht) sowie den Zeichenbenutzer

**Zeichendimensionen:**

- **Syntaktische Dimension:** beschreibt die gültigen Kombinationen sprachlicher Zeichen (im Deutschen steht das Adjektiv vor dem Substantiv)
- **Semantische Dimension:** Bedeutung sprachlicher Zeichen in einer Sprachgemeinschaft (Auto = vier Räder, Motor, Lenkrad, Pedale etc.)
- **Pragmatische Dimension:** Bedeutung sprachlicher Zeichen für einen Sprecher (Ferrari = teures Luxusobjekt)

**Qualität konzeptueller Modelle:**

- **Syntaktische Qualität** (Qualität der Form): bezieht sich auf die Einhaltung syntaktischer Regeln
- **Semantische Qualität** (Qualität der Bedeutung): bezieht sich auf die Beziehungen zwischen Repräsentationen (d.h. verwendete Sprachkonstrukte) und den Konzeptualisierungen der betrachteten Domäne und prüft, inwieweit die Informationen im Modell mit dem betrachteten Gegenstandsbereich korrespondieren.
  - Validität des Modells: misst die Korrektheit der abgebildeten Aspekte
  - Vollständigkeit des Modells: Maß für den Umfang der korrekt abgebildeten Aspekte
- **Pragmatische Qualität** (Qualität der Verwendung): bezieht sich auf die Eignung des Modells für den jeweiligen Verwendungszweck

- oftmals können Fehler in konzeptuellen Modellen nicht eindeutig einer der drei Qualitätsklassen zugeordnet werden

#### Merkmale einer Sprache:

- Sprache = System von Zeichen
- das Systemische wird durch die **Syntax** beschrieben
- die Sprachzeichen erhalten Bedeutung (**Semantik**) durch Sprechen (Aktualisieren) von Mitgliedern einer Sprachgemeinschaft
- jedes Mitglied der Sprachgemeinschaft weist sprachlichen Zeichen eine geringfügig unterschiedliche Bedeutung zu (**Pragmatik**)

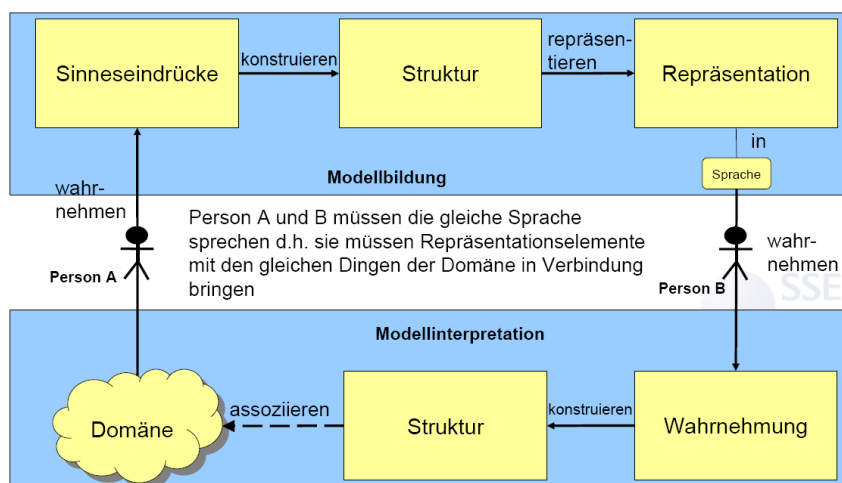
#### Konzeptuelle Modellierungssprachen:

- **Abstrakte Syntax:** Menge der möglichen Kombinationen der Sprachkonstrukte ohne Festlegung einer Notation
- **Konkrete Syntax:** stellt den Bezug zwischen den abstrakten Sprachkonstrukten und der konkreten grafischen/textuellen Repräsentation (Notation) dieser Konstrukte her
- **Semantik:** legt die Bedeutung von Sprachkonstrukten fest
- sind sowohl Syntax als auch Semantik formal spezifiziert, handelt es sich um eine formale Modellierungssprache

**Metamodellierung:** Eine konzeptuelle Modellierungssprache kann selbst durch ein konzeptuelles Modell definiert werden (Metamodell). Das Metamodell kann wiederum unter Verwendung einer Modellierungssprache (Metasprache) definiert werden. Die Konstruktion von Metaebenen kann beliebig fortgeführt werden, mehr als vier Ebenen sind aber nicht üblich.

**Modellbildung und –interpretation:** Im Rahmen der Modellbildung treten zwei Transformationen auf, die zu Informationsverlust, Unvollständigkeiten bzw. Mehrdeutigkeiten führen können:

- Wahrnehmungstransformation: beobachtete Realität wird von einer Person nicht vollständig oder verfälscht konzeptualisiert
- Darstellungstransformation: abhängig von Wissen, persönlicher Prägung und Erfahrungen sowie der aktuellen Situation werden gedankliche Bilder unterschiedlich dokumentiert



## Datenmodellierung

### Ziele der Datenmodellierung:

- Physische Datenunabhängigkeit: unabhängig von den Strukturierungsbeschränkungen (z.B. Datenbanken)
- Logische Datenunabhängigkeit: neutral gegenüber den Einzelanwendungen und deren lokalen Sichten
- Anwendungsübergreifendes gleiches Verständnis, Unternehmensgesamtansicht

### Vorgehensweise bei der Modellbildung:

- 1) Domäne: Entity (Gegenstand), Relationship (Beziehung)
- 2) Klassifikation: Entity-Set, Relationship-Set (jeweilige Objektmenge)
- 3) Modellwelt: Entity-Typ, Relationship-Typ (jeweiliger Objekttyp)

### ERM-Sprachelemente:

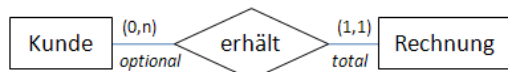
- Basiskonstrukte: Entitäten, Attribute, Beziehungen
- Erweiterte Konstrukte: Kardinalitäten, Rollen, Generalisierung, Vererbung

### Notationsform für Kardinalität und Partizipation:



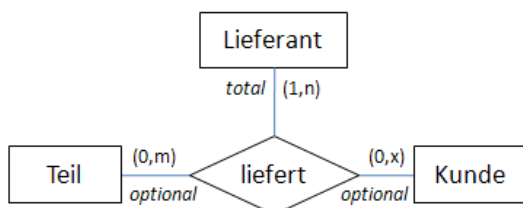
Eine Person kann mehrere Bücher besitzen, wobei jedes Buch genau einen Besitzer hat.

### Optionale und totale Beziehungen:



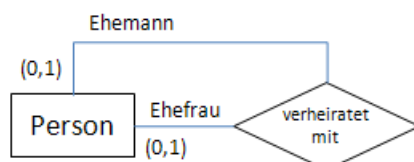
Beziehungstypen können als kann-Beziehung (optional) und muss-Beziehungen (total) unterschieden werden.

### Ternäre Beziehungen:



Ein Lieferant muss mindestens an einer „liefert“-Beziehung beteiligt sein, höchstens an n. Ein Kunde kann an x „liefert“-Beziehungen teilhaben, muss er aber nicht. Gleiches gilt für ein Teil.

### Zyklische Beziehungen und Rollen:

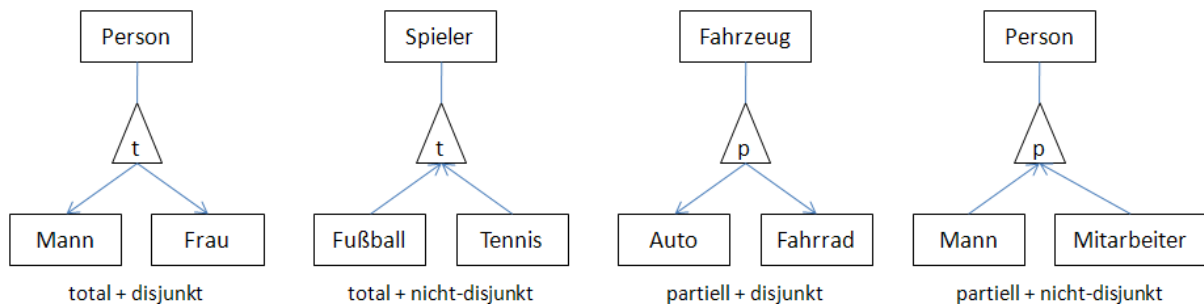


Durch die Einführung von Rollen kann die Verständlichkeit zyklischer Beziehungen verbessert werden.

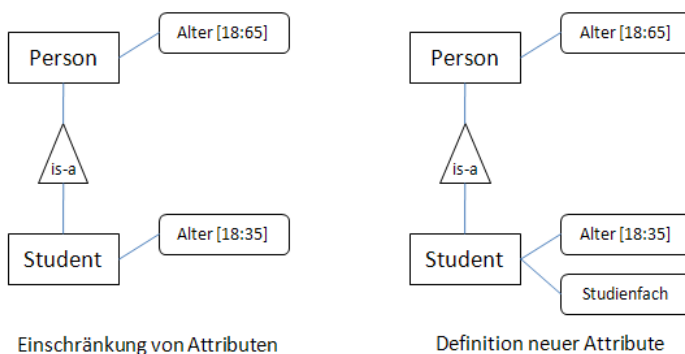
### Generalisierung und Spezialisierung:

- durch Generalisierung lassen sich semantisch ähnliche Entitätsmengen (Subtyp) zu einer übergeordneten Menge (Supertyp) zusammenfassen.

- **disjunkt:** Ein Objekt kann entweder zu der einen oder zu der anderen Entitätsmenge gehören, nicht aber zu beiden.
- **total:** Es gibt keine anderen Subtypen außer den angegebenen.



### Vererbung von Eigenschaften:



## Funktionsmodellierung

### Bestandteile der strukturierten Analyse:

- **Datenflussdiagramm (DFD)** beschreibt grafisch die Funktionalität des Systems
- **Minispezifikation** definiert die in den DFD dargestellten Funktionalitäten detailliert (für atomare Funktionen)
- **Data Dictionary (Glossar)** definiert die Daten (Begriffe), die in Datenflussdiagramm und Minispezifikation verwendet werden, an zentraler Stelle

### Elemente des Datenflussdiagramms:

- **Datenfluss:** Pipeline, durch die Informationspakete mit bekannter Zusammensetzung fließen (Daten in Bewegung)
- **Knoten:** stellen Prozesse dar, d.h. Transformationen von Eingangsdatenflüssen in Ausgangsdatenflüsse
- **Datenspeicher:** stellen Ablagen dar (Daten in Ruhe)
- **Datenquelle/-senke:** Terminatoren sind Personen, Organisationen oder Systeme außerhalb der Grenzen des betrachteten Systems, die Sender oder Empfänger von Daten des Systems sind

**Zu vermeidende Modellierungsfehler:**

- zu komplexe Schnittstellen (zu viele ein- und ausgehende Datenflüsse)
- Informationssenke (nur eingehende Datenflüsse)
- Wundersame Datenschaffung (nur ausgehende Datenflüsse)
- Write- bzw. Read-Only Datenspeicher (Datenspeicher benötigen jeweils mind. einen Schreib- und einen Lesezugriff)
- Prozess ohne Funktion (einfache Weiterreichung von Informationen ohne Verarbeitung)
- Anzeichen von Flussdiagrammen (keine Kontrollstrukturen abbildbar)
- Springende Daten (Datenspeicher greift direkt auf anderen Datenspeicher zu)
- Vorgänge außerhalb des Systems (Informationsfluss zwischen Terminatoren)
- Nichtssagende Fehler

**Darstellung als Hierarchien:**

- auf oberster Ebene Kontextdiagramm mit Darstellung der Terminatoren
- dann schrittweise verfeinerte Überblicksdiagramme (Ebene 0, Ebene 1, ...)
- auf unterster Ebene Darstellung der Basisfunktionen
- Empfehlung:  $7 \pm 2$  Knoten/Datenspeicher pro Ebene
- Ein- und Ausgaben im Eltern-Knoten müssen auch Ein- und Ausgaben beim Kind-Knoten sein
- Arten des Balancings:
  - **Sichtbares Balancing:** ein Prozess wird in Teilprozesse aufgeteilt, Ein- und Ausgaben bleiben erhalten
  - **Dictionary Balancing:** Aufteilung von Datenflüssen, Erläuterung finden sich im Dictionary
  - **Datenspeicher Balancing:** ein Datenspeicher tritt zum ersten Mal in einem Diagramm auf, wenn er Schnittstelle zwischen (mind.) zwei Knoten ist. Ab dann müssen alle Zugriffe auch in den Verfeinerungen aufgezeigt werden.

**Data Dictionary:**

- enthält Erklärungen zu jedem Datenfluss und jedem Datenspeicher
- vermeidet wechselnde Interpretationen der lesenden Personen
- ohne Data Dictionary fehlt es den Diagrammen an Aussagekraft, ohne DFD ist das Data Dictionary wenig brauchbar
- Tipps: Vermeidung von Redundanz, Verwendung selbstdefinierter Ausdrücke

**Data Dictionary Notation:**

a = b	a besteht aus b
a + b	a zusammen mit b
x{...}y	mehrere davon (x ist Untergrenze, y ist Obergrenze; x und y sind optional)
[a b c]	entweder a, b oder c
(a)	a ist optional
„...“	primitives Datenelement
* ... *	Kommentar

**Minispezifikationen:**

- beschreiben die Strategie der Erzeugung der Ausgabedaten aus den Eingabedaten
- beschreiben keinen Algorithmus und keine Handlungsanweisungen



- Tipps: ½ bis 1 Seite Text pro Knoten, Verwendung von eindeutigen Sprachstil (Begriffe aus Dictionary verwenden, kurze und positive Aussagen)

#### Vorgehensweisen:

- **Kontextabgrenzung:**
  - Zweck des Systems festlegen
  - Kontextdiagramm erstellen
  - Ereignisliste erstellen (Stimulus von Umwelt ans System, Response zurück an die Umwelt)
- **Datenverfolgung:**
  - Ein- und Ausgabedaten sowie die Verfolgung deren Metamorphose stehen im Mittelpunkt der Betrachtung
  - Vorteile: erfolgreich bei Systemen mit starken Transformationscharakter, hilfreich beim Erlernen des Denken in Datenflüssen sowie zur Unterscheidung von wesentlichen und unwesentlichen Funktionen
  - Nachteile: Zwischendatenstrukturen sind häufig nicht offensichtlich, schwierig bei größeren Systemen, erzeugt meist große Modelle
- **Ereignisorientierte Zerlegung:**
  - Betrachtung von Ereignissen (externe oder Zeitereignisse) in der realen Welt, die das System betreffen und Modellierung der geplanten Reaktionen in Form von Szenarien
  - Ereignisse können nicht beeinflusst, sondern nur wahrgenommen werden, da sie in der Umwelt stattfinden
  - Vorgehensweise: Ziele des Systems festlegen, Kontextabgrenzung, Finden der wichtigen Ereignisse, Modellieren von Ereignisszenarien, Zusammenfassung & Hierarchisierung, Vervollständigen des Modells (Data Dictionary, Minispezifikationen)
  - Vorteile: Zusammen mit den Techniken für das Kontextdiagramm die bewährteste Vorgehensweise, Szenarien sind leicht zu erstellen, hilft beim Denken in Zusammenhängen und bei Diskussionen mit Anwendern
  - Nachteile: führt nicht immer zu guter Datenspeichergliederung, bei sehr großen Systemen wird das Problem der Diagrammzerlegung auf das Problem der Ereignisbündelung verlagert

### Verhaltensmodellierung

#### Hauptkonzepte der Verhaltensmodellierung:

- Ein **Zustand** ist die Ausprägung der Eigenschaften eines Objekts zu einer Zeit
- Der **mögliche Zustandsraum** ist das kartesische Produkt aller Eigenschaftswertebereiche eines Objekts.
- Die Verhaltensmodellierung beschäftigt sich mit der Eingrenzung des möglichen Zustandsraums und damit der Definition des erlaubten Zustandsraums.
- Der **erlaubte Zustandsraum** ist die Teilmenge des möglichen Zustandsraums, die nur erlaubte Zustände enthält.
- Ein **erlaubter Zustand** ist ein Zustand, der gegen kein Zustandsgesetz verstößt.





**Weiterführende Konzepte:**

- **Ereignis:**
  - geordnetes Paar von Zuständen
  - kein Zeitverbrauch, Zeit ist ein implizites Attribut von einem Ereignis
  - logische Reihenfolge oder unabhängig
  - gerichteter Datenaustausch
  - keine Unterscheidung zwischen Fehlerereignissen und normalen Ereignissen
- **Transformation/Transition:** Übergang von einem Zustand zum anderen
- **Systemreaktion:**
  - legt Reaktion auf ein empfangenes Ereignis fest
  - Ergebnis kann bestehen aus Ausführung einer Aktion oder einer Zustandsänderung
  - repräsentiert ein Zeitintervall zwischen zwei Ereignissen

**Endliche Automaten:**

- sind als mathematische Modelle diskreter Systeme verwendbar (Zustände sowie Ein- und Ausgabezeichen sind Elemente einer diskreten und endlichen Menge)
- Systeme mit endlicher Zustandsanzahl sind in vielen Bereichen von Bedeutung (OCR-Texterkennung, Texteditoren, grep-Befehl)
- Theorie von Automaten und formalen Sprachen für eine große Klasse von Mustererkennungen anwendbar

**Notationslemente für Endliche Automaten:**

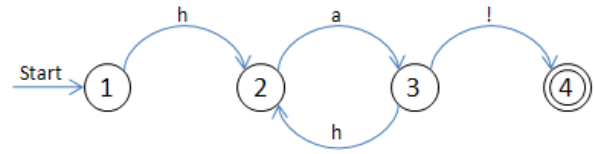
Element	Bedeutung	Bemerkung
	Zustand (state)	Beschriftung durch eindeutige Nummerierung oder Zustandsbeschreibung
	Übergang (transition)	Beschriftung durch Eingabezeichen
	Startzustand (start state)	
	Endzustand (final state)	Der Automat hat eine gültige Folge von Eingabezeichen erkannt → akzeptierter Zustand

**Deterministisch endlicher Automat (DEA):**

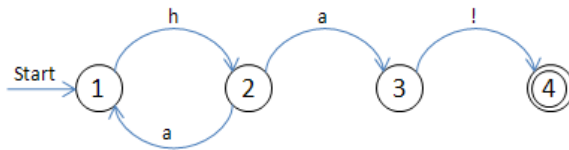
- wird durch ein Fünf-Tupel beschrieben:  $\langle Z, \Sigma, \delta, s, F \rangle$
- $Z$ : endliche, nicht leere Menge von Zuständen
- $\Sigma$ : endliches Eingabealphabet
- $\delta$ : Übergangsfunktion  $Z \times \Sigma \rightarrow Z$
- $s$ : Startzustand  $s \in Z$
- $F$ : endliche Menge von Endzuständen  $F \subseteq Z$

**Beispiel für einen DEA:**

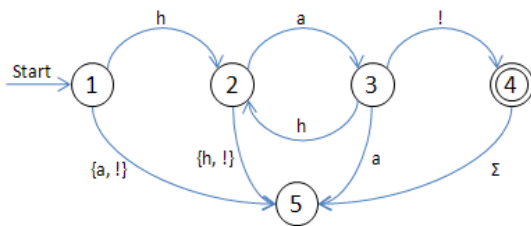
- Zustände  $Z = \{1, 2, 3, 4\}$
- Eingabealphabet (Alphabet):  $\Sigma = \{a, h, !\}$
- Übergangsfunktion:  $\delta = \{ \langle 1, h, 2 \rangle, \langle 2, a, 3 \rangle, \langle 3, h, 2 \rangle, \langle 3, !, 4 \rangle \}$
- Startzustand:  $s = 1$
- Endzustand  $F = \{4\}$

**Nicht-deterministisch endlicher Automat (NEA):**

- Automat, der für ein Paar (Zustand, Eingabe) mehrere Folgezustände besitzen kann
- kann simultan in mehreren Zuständen sein  $\rightarrow$  nutzt den Zustand, der zu einem Endzustand führt
- lässt sich in einen DEA überführen, der die gleiche Sprache spricht
- wird durch ein Fünf-Tupel beschrieben:  $\langle Z, \Sigma, \delta, s, F \rangle$
- $Z$ : endliche, nicht leere Menge von Zuständen
- $\Sigma$ : endliches Eingabealphabet
- $\delta$ : Übergangsfunktion  $Z \times \Sigma \rightarrow 2^Z$  (Abbildung in die Potenzmenge von  $Z$ )
  - Beispiel: Übergangsfunktion:  $\delta = \{ \langle 1, h, \{2\} \rangle, \langle 2, a, \{1, 3\} \rangle, \langle 3, !, \{4\} \rangle \}$
- $s$ : Startzustand  $s \in Z$
- $F$ : endliche Menge von Endzuständen  $F \subseteq Z$



**Partieller und vollständiger DEA/NEA:** Ein DEA/NEA ist **vollständig**, wenn für jedes Zeichen aus  $\Sigma$  in jedem Zustand aus  $Z$  genau ein (bei DEA) bzw. mindestens ein (bei NEA) Zustandsübergang definiert ist. Ansonsten ist er **partiell**.

**Probleme von endlichen Automaten:** Zustandsexplosion

- Beispiel Bankautomat
- Problem: Abbrechen des gesamten Vorgangs  $\rightarrow$  von jedem Zustand einen Zustandsübergang in den Initialzustand
- Lösung: Erweiterung von endlichen Automaten  $\rightarrow$  Statecharts

**Statecharts:**

- **Bedingungen für Zustandsübergänge:**
  - ein Zustandsübergang wird durch ein Ereignis ausgelöst

- die Bezeichnung für das auslösende Ereignis wird an den Zustandsübergang geschrieben
- das Ereignis kann durch eine Bedingung eingeschränkt werden
- Aktionen können an den Zustandsübergang oder an den Zustand geschrieben werden
- **Hierarchisierung:**
  - haben verschiedene Zustände den gleichen Zustandsübergang zu einem Folgezustand, so lassen sich diese zu einem „Super-State“ zusammenfassen
- **Default Eingänge:**
  - der Super-State hat einen zentralen Eingang
  - alle Ereignisse der Außenwelt lösen „intern“ den gleichen Zustand aus
  - der Eingang wird an einen definierten Zustand (Default Entry) geleitet
- **Nebenläufige Zustände:**
  - Zustände werden i.d.R. sequentiell abgearbeitet
  - durch Aufteilen des Super-States in mehrere Unterteile wird eine parallele Abarbeitung von Zuständen möglich
  - die Zustände der Partiellen haben je einen Default Entry

### *Kernaktivität Gewinnung (Buch: Kapitel 20-22)*

**Motivation:** Ziel der Gewinnung ist der Fortschritt in der Inhaltsdimension und somit letztendlich ein möglichst vollständiges inhaltliches Verständnis aller Anforderungen an das geplante System

**Gewinnung von Anforderungen an das geplante System:** in der Gewinnungsaktivität unterscheiden wir drei Arten von Anforderungen:

- **Existierende Anforderungen:**
  - Vorstellungen von Stakeholdern über das geplante System, definierte Schnittstellen sowie Anforderungen, die in Dokumenten enthalten sind
  - können z.B. durch Interviews von Stakeholdern oder durch gezieltes Lesen von Dokumenten gewonnen werden
- **Rahmenbedingungen:**
  - Einschränkungen, die für das geplante System gelten
  - können ebenfalls in Anforderungsquellen enthalten sein
  - Gewinnung von Rahmenbedingungen erfolgt durch die gleichen Techniken wie die Gewinnung von existierenden Anforderungen
- **Innovative Anforderungen:**
  - sind in keiner Anforderungsquelle enthalten bzw. keiner Anforderungsquelle bekannt
  - werden in einem kreativen und häufig iterativen Prozess entwickelt

**Ziele und Szenarien in der Gewinnung:**

- Ziele und Szenarien werden im Rahmen der Gewinnungsaktivität gewonnen, unterstützen aber gleichzeitig die Gewinnung von Anforderungen
- Ziele versetzen die Stakeholder in die Lage, ihre Intentionen auf abstrakte und einfache Weise zu formulieren. Stakeholder können bekannte Ziele für das geplante System bspw. in einem

Workshop durch Zielbäume verfeinern und somit ein besseres Verständnis über das geplante System etablieren.

- Szenarien bieten Stakeholdern eine intuitive Möglichkeit, Anforderungen an das geplante System in konkreter Form zu beschreiben, z.B. durch konkrete Interaktionen mit dem geplanten System. Ausgehend von den definierten Zielen formulieren die Stakeholder Szenarien, die die Erfüllung der Ziele anhand konkreter Beispiele illustrieren.

**Gewinnung von Anforderungen:** Die Keraktivität Gewinnung verfolgt drei Teilziele:

**(1) die Identifikation von Anforderungsquellen im Kontext des geplanten Systems**

- Quellen von Anforderungen sind nicht immer offensichtlich
- Einbeziehung aller *relevanten* Quellen ist essenziell
- iterative Identifikation, in dem basierend auf bereits bekannten Anforderungsquellen weitere potenzielle Anforderungsquellen identifiziert werden
- Berücksichtigung aller vier Kontextfacetten
- Aufgrund von Ressourcenknappheit kann es nötig sein, die Relevanz von potentiellen Anforderungen zu bestimmen (z.B. durch die Vergabe von Relevanzpunkten)

**(2) die Gewinnung von existierenden Anforderungen**

- von Stakeholdern: durch Gespräche, durch schriftliche Befragung oder durch Beobachtung
- aus Dokumenten: durch Lesen und Analysieren, wichtig ist die Gewinnung relevanter Anforderungen aus den Dokumenten mit akzeptablem Aufwand
- aus existierenden Systemen: durch Beobachtung des Systemverhaltens und der Systemnutzung, Befragung von Stakeholdern existierender Systeme sowie Analyse der Dokumentation existierender Systeme

**(3) die Entwicklung von innovativen Anforderungen**

- können nicht von Stakeholdern erfragt, aus Dokumenten gewonnen oder durch Beobachtung entdeckt werden
- werden in einem kreativen, iterativen Prozess entwickelt, z.B. Ausarbeitung einer neuartigen Idee oder eine neuartige Kombination bekannter Technologien

**Gewinnungstechniken:**

- **Interview:**
  - der Interviewer erarbeitet sich ein Verständnis der Anforderungen der Stakeholder
  - zwei Arten: exploratives (Abweichung von Fragen möglich) und standardisiertes (ermöglicht bessere Vergleiche) Interview
  - Einzel- und Gruppeninterviews möglich
  - Geschlossene (vorgegebene Antworten) und offene (zur Gewinnung innovativer Ideen) Fragen
  - nicht primär für die Entwicklung innovativer Anforderungen bestimmt, aber grundsätzlich sind Interviews für alle Gewinnungsarten nützlich
- **Workshop:**
  - Anforderungen werden in Gruppenarbeit erarbeitet
  - Verwendung von verschiedenen Assistenztechniken (z.B. Brainstorming, Diskussion, Kleingruppenarbeit mit Präsentation der Ergebnisse)
  - geplantes Vorgehen für den Workshop wird in einer Agenda festgelegt

- Ergebnisse werden während des Workshops dokumentiert
- sehr hoher Planungs- und Koordinationsaufwand nötig
- für alle Gewinnungsarten nützlich
- **Beobachtung:**
  - Stakeholder beschreiben ihre Tätigkeiten während der Durchführung wesentlich besser als retrospektiv nach einer durchgeführten Tätigkeit
  - direkte Beobachtung: Beobachtung des eigenständigen Betriebs des Systems
  - ethnografische Beobachtung: Beobachter erwirbt tiefes Verständnis über die beobachteten Tätigkeiten, in dem er – soweit möglich – alle Arbeitsabläufe der Stakeholder selbst ausführt
  - nur zur Gewinnung von existierenden Anforderungen nützlich
- **Schriftliche Befragung:**
  - Stakeholder beantworten vorgegebene Fragen schriftlich
  - strukturiertes Aufschreiben unterstützt die Reflektion der Gedanken
  - gut geeignet, um von einer großen Anzahl an Stakeholdern eine initiale Menge von Anforderungen oder auch potenzieller Anforderungsquellen zu gewinnen
  - eingeschränkte Möglichkeit für Rückfragen
  - nicht so gut für die Entwicklung innovativer Anforderungen geeignet
- **Perspektivenbasiertes Lesen:**
  - ein Dokument wird aus einer zuvor festgelegten Perspektive gelesen (z.B. aus der Sicht eines Nutzers oder eines Testers)
  - für die jeweilige Perspektive nicht relevante Details können ignoriert werden
  - besonders für die Gewinnung von existierenden Anforderungen aus Dokumenten geeignet
  - sequenzielles (von vorne nach hinten) oder Top-down-Lesen (mit Hilfe von Verzeichnissen)

#### Assistenztechniken zur Gewinnung:

- **Brainstorming:**
  - Ziel ist die Entwicklung einer großen Anzahl neuer Ideen zu einem vorgegebenen Thema
  - Durchführung in einer Gruppensitzung
  - Regeln: Quantität vor Qualität, freies Assoziieren und Phantasieren explizit erwünscht, dagegen ist Kritik nicht zugelassen, Verständnisfragen sind erlaubt
  - gesammelte Ideen sollten anschließend priorisiert werden
  - sehr gut für die Entwicklung innovativer Anforderungen geeignet, aber auch für die Identifikation potenzieller Anforderungsquellen
- **Prototypen:**
  - ein Prototyp ist eine initiale Version eines Softwaresystems zur Demonstrierung von Konzepten, zur Erprobung von Entwürfen und zum Verbessern des generellen Verständnisses
  - Prototypen können für die Validierung verwendet werden und können Stakeholder dazu stimulieren, Anforderungen an das geplante System zu formulieren
  - Verwendung: Wegwerf- oder evolutionärer Prototyp
  - Funktionalität: horizontaler oder vertikaler Prototyp

- geeignet für die Gewinnung existierender und Entdeckung innovativer Anforderungen
- **Kartenabfrage:**
  - jeder Teilnehmer kann seine Gedanken unabhängig von anderen Teilnehmern aufschreiben (gut für eher introvertierte Personen)
  - Stimulation zusätzlicher Ideen im Unterschied zu Brainstorming eher eingeschränkt
  - somit nicht so sehr für die Entwicklung innovativer Anforderungen geeignet
- **Mind Maps:**
  - Denkwerkzeug zur Gedächtnishilfe, zur Strukturierung und Systematisierung von Informationen, zur Unterstützung des kreativen Denkens und zur Vorbereitung des Schreibprozesses
  - alles, was das Denken bezüglich einer zentralen Idee hervorbringt, soll möglichst anschaulich und einprägsam visualisiert werden
  - können in allen Teilaktivitäten zur Gewinnung von Anforderungen eingesetzt werden
- **Checklisten:**
  - fasst eine Anzahl von Fragen oder Aussagen zu einem Sachverhalt zusammen
  - sind sehr hilfreich, wenn bei einem komplexen Sachverhalt viele Aspekte berücksichtigt werden müssen und hierbei möglichst kein Aspekt vergessen werden soll
  - sind für alle Gewinnungsbereiche gut geeignet, werden auch zur Validierung und Dokumentation verwendet