

Fragenkatalog

Systems Engineering SYE3IL

Erstellt von:

Darius Faje

Alex

Studiengang: MB

13.12.2024

Inhaltsverzeichnis

1	Teil	3
1.1	Wie kann Komplexität von Systemen charakterisiert werden?	3
1.2	Nennen Sie externe und interne Komplexitätstreiber!	3
1.3	Was versteht man unter Cyber-Physikalischen Systemen	4
1.4	Was versteht man unter Internet of Things	4
1.5	Was versteht man unter Systems of Systems?	4
1.6	Was ist ein Produkt-Service-System (PSS)?	5
1.7	Welche Klassen von Produkt-Service-Systemen gibt es?	5
1.8	Aus welchen Bestandteilen besteht ein System?	6
1.9	Grundelemente von Systems Engineering?	7
1.10	Begriffe wofür Systems Engineering steht!	7
1.11	Nenne die 6 generische Schritte die uns helfen Aufgaben zu lösen?	8
1.12	Was bedeutet SADT + Beispiel.	9
1.13	Was sind Swimlane-Diagramme + Beispiel	9
1.14	Vier Grundregeln des Agilen Manifests	10
1.15	Bedeutung SCRUM + Rollen	10
2	Teil	11
2.1	Was versteht man unter einem Requirement?	11
2.2	Was ist ein Stakeholder?	11
2.3	Wie kann man Requirements Engineering und Requirements Management unterscheiden?	11
2.4	Wie werden Anforderungen angegeben? —nicht sicher—	11
2.5	Nennen Sie Funktionale Anforderungen. (Zusatz!)	12
2.6	Nennen Sie Nicht-Funktionale Anforderungen.	12
2.7	Was sind typischen Schritte im Requirements Engineering?	13
2.8	Nennen Sie Quellen für Anforderungen?	13
2.9	Was sind Qualitätskriterien für Anforderungen?	14
2.10	Was sind typische Bestandteile einer Anforderungsliste?	14
2.11	Mit welcher Methode können Zielkonflikte erkannt werden und erklären Sie diese!	15
3	Teil	16
3.1	Welche drei Schritte sind zur Bildung einer Modularen Produktarchitektur relevant?	16
3.2	Welche Funktions- und Bausteinarten in Baukästen kennen Sie?	16
3.3	Was sind Vor- und Nachteile von Produktplattformen?	17
3.4	Was versteht man unter Kundenindividuelle Massenproduktion?	17
3.5	Was ist ein Graph und welche Arten gibt es?	17
3.6	Erklären Sie einen gewichteten Graphen?	19
3.7	Wofür steht der Begriff DSM?	19
3.8	Welche Arten von DSM gibt es?	20
3.9	Was bedeutet Sequenzierung von DSM?	22
3.10	Was bedeutet Clustering von DSM?	24

4	Teil	26
4.1	Was ist ein Modell?	26
4.2	Was bedeutet Problemlösung durch Abstraktion (Modellbildung) und Interpretation?	26
4.3	Welche Verfahren der Modellbildung gibt es?	27
4.4	Wie ist der Ablauf zur Entstehung eines rechnerinternen Modells?	27
4.5	Was ist bei der Interpretation von Ergebnissen wichtig?	28
4.6	Was bedeutet Simulation und wofür ist dies relevant?	28
5	Teil	30
5.1	Erklären Sie in eigenen Worten den Begriff Digitaler Zwilling!	30
5.2	Was ist ein Digitaler Schatten?	30
5.3	Was bedeutet Functional Mock-Up Interface?	30

1 Teil

1.1 Wie kann Komplexität von Systemen charakterisiert werden?

Komplexität beschreibt, dass das Verhalten eines Modells schwierig zu beschreiben ist, obwohl ausreichend Information über seine Komponenten und deren Verflechtungen vorliegen. Was als komplex gilt kommt immer auf die Sichtweise drauf an. → Schwer es allgemein zu definieren.

Ein **komplexes System** ist ein System mit einer großen Zahl an Komponenten und Verknüpfungen, Interaktionen oder Abhängigkeiten, welches schwierig beschreibbar, planbar oder steuerbar ist.

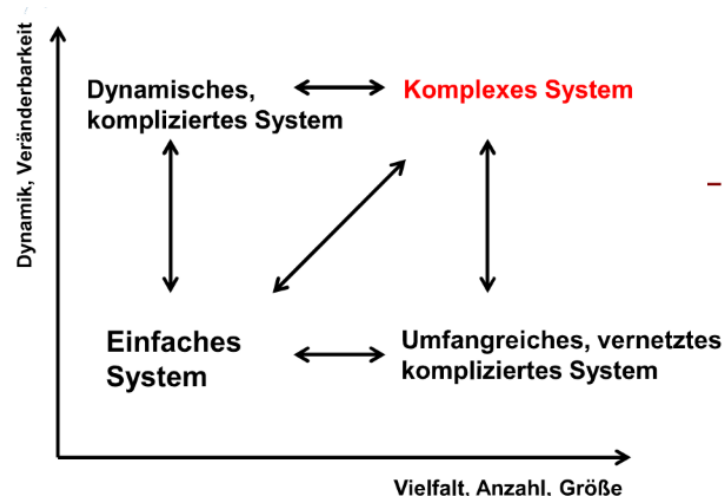


Abbildung 1: Swimlane Diagramm

1.2 Nennen Sie externe und interne Komplexitätstreiber!

Externe	Interne
Heterogene Märkte	Aufwendiges Produktkonzept
Individuelle Kundenbedürfnisse	Hohe Komponentenvielfalt
Erforderliche Variantenvielfalt	Komplexer Entwicklungsprozess
Wettbewerb und Gesetzte	Komplexe Fertigung
Preisdruck	Viele Zulieferer und Projektbeteiligte

Tabelle 1: Vergleich von externen und internen Faktoren

Die kann die internen Komplexitätstreiber selber vielleicht beeinflussen die externen nicht!

1.3 Was versteht man unter Cyber-Physikalischen Systemen

- **Eine Integration von Computern und physischen Prozessen:**
Physische Prozesse werden durch eingebettete Computer und Netzwerke überwacht, geregelt und gesteuert.
- **Gegenseitige Rückkopplung:**
Computer und physische Prozesse beeinflussen sich gegenseitig. (Wechselwirkung)
- **Beispiele:**
Autonome Fahrzeuge, Smartwatch, Produktionsanlagen.

1.4 Was versteht man unter Internet of Things

- **Ein globales Netzwerk von physischen und virtuellen Objekten:**
Diese Objekte kommunizieren und arbeiten zusammen.
- **Technologie zur Unterstützung des Menschen:**
Ermöglicht Interaktion zwischen Mensch und vernetzten Systemen sowie zwischen den Systemen selbst.
- **Beispiele:**
Kühlschränke mit automatischer Nachbestellung. Hat sich besonders bei Supermärkten durchgesetzt

1.5 Was versteht man unter Systems of Systems?

Ein Systems of Systems (SoS) ist:

- **Ein Verbund autonomer Systeme:**
Diese Systeme arbeiten zusammen, um ein neues, komplexeres System mit zusätzlicher Funktionalität zu schaffen.
- **Charakteristische Merkmale:**
 - **Operative Unabhängigkeit der Subsysteme:** Jedes Subsystem arbeitet unabhängig.
 - **Unabhängige Verwaltung der einzelnen Systeme:** Die Systeme werden eigenständig verwaltet.
 - **Evolutionäre Entwicklung:** Ständige Weiterentwicklung und Anpassung der Systeme/Komponenten. (ohne das das gesamte System zusammenbricht)
 - **Emergentes Verhalten:** Ausfall von einem Teil führt nicht zum Ausfall des gesamten Systems.
 - **Geografische Verteilung der Systeme:** Die Systeme können an verschiedenen geografischen Orten verteilt sein.

1.6 Was ist ein Produkt-Service-System (PSS)?

Ist ein **Leistungsangebot** mit signifikanten **Anteilen an Sachgütern als auch Dienstleistungen**. Dienstleistungen sind integraler Bestandteil des PSS.

Z.B. Carsharing, Leasing

1.7 Welche Klassen von Produkt-Service-Systemen gibt es?

Einteilung in 3 Teile:

- **Reines Produkt**
 - Produkt im Besitz des Kunden.
 - Kunde selber verantwortlich.
 - z.B. Kauf Waschmaschine, Auto
- **Reiner Service**
 - Reine Dienstleistung.
 - Produkt nicht im Besitz des Kunden.
 - z.B. Waschsalon, Beratung
- **Produkt Service Systeme:**
 - **Produktorientiert:**
 - * Klassischer Verkauf im Vordergrund.
 - * Produktbezogen, Beratung (Auto + Einschulung oder Wartung)
 - **Nutzungsorientiert:**
 - * Nutzung des Produktes im Vordergrund.
 - * Produkt im Besitz des Anbieters.
 - * z.B. Leasing, Mieten, Pooling
 - **Ergebnisorientiert:**
 - * Ergebnis steht im Vordergrund
 - * Übereinkunft zwischen Anbieter und Kunde über zu erreichendes Ziel.
 - * Die Wahl des Produktes und des Vorgehens obliegt dem Anbieter.
 - * Honorar nach Ergebnis

1.8 Aus welchen Bestandteilen besteht ein System?

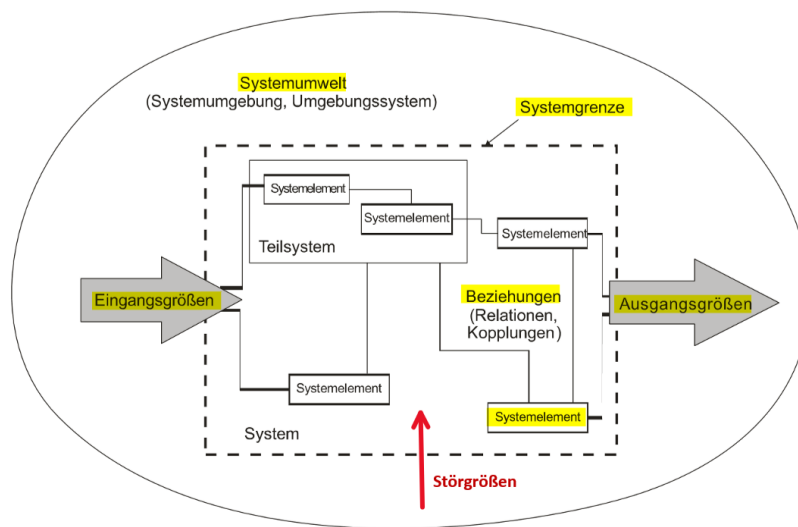


Abbildung 2: zentrale Darstellung eines Systems

- **Systemumwelt**
 - Alles, was nicht in das System einbezogen ist (abgetrennt an Systemgrenze).
- **Stoff-, Energie-, Informationsfluss**
 - Art und Weise der Interaktion mit Systemumwelt.
 - Systemgrenze so wählen, dass Kopplung zur Umgebung sehr viel schwächer als Kopplung im Inneren ist.
- **Eingang**
 - Stellt Relation der Umwelt zum System dar.
 - Vom Verhalten des Systems nicht beeinflusst.
 - **Stellgrößen**: gezielte Veränderung des Eingangs.
 - **Störgrößen**: unkontrollierte Veränderung des Eingangs (z.B. Rauschen).
- **Ausgang**
 - Stellt Relation vom System zur Umgebung dar (z.B. Messgröße).
- **Teilsystem**
 - Element eines Systems, welches weitere Elemente enthält (enthält eigenes System - Subsystem).
- **Störgrößen**
 - Nicht gewünschte Eingangsgrößen (Bsp. Rauschen)

1.9 Grundelemente von Systems Engineering?

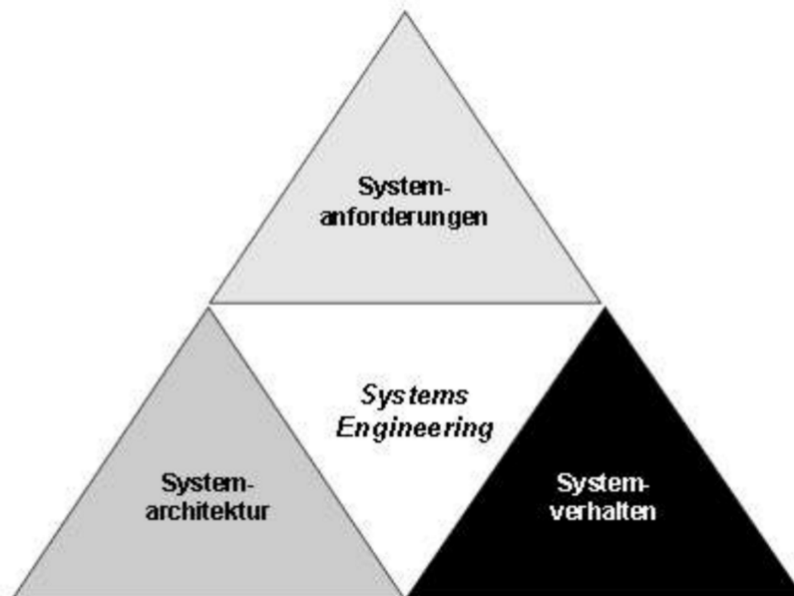


Abbildung 3: Grundelemente System Engineering

- Systemanforderungen:
 - Was ist notwendig um eine Anforderung zu beschreiben
- Systemarchitektur:
 - Wie baue ich ein System aus?
 - Aus welchen Modulen?
 - Wie baue ich die Module zusammen?
- Systemverhalten
 - Wie beschreibe ich das Systemverhalten?
 - Modellierung und Simulation

1.10 Begriffe wofür Systems Engineering steht!

- Interdisziplinär
- Früh (in Entwicklungsprozess)
- Dokumentieren (Anforderungen)
- Gesamtheitlich (auf System Bezogen)
- Wirtschaftlich (Bedürfnisse des Kunden)
- Technisch

1.11 Nenne die 6 generische Schritte die uns helfen Aufgaben zu lösen?

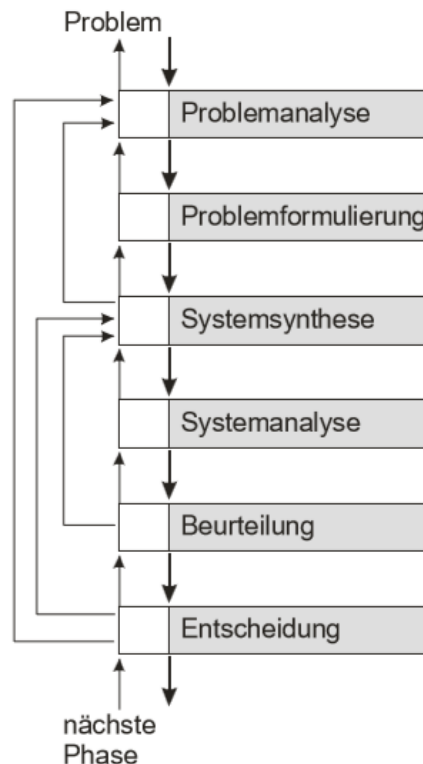


Abbildung 4: 6 generische Schritte

- Problemanalyse:
 - Das Problem analysieren bzw. verstehen.
- Problemformulieren:
 - Das Problem in Worte beschreiben.
 - Anforderungen definieren.
- Systemsynthese:
 - In diesen Schritt werden Konzepte bzw. Ideen erstellt/ überlegt.
- Systemanalyse:
 - analysieren der Konzepte.
- Beurteilung:
 - Die Konzepte werden nach verschiedenen Kriterien Beurteilt bzw. Verglichen.
- Entscheidung:
 - Entscheiden ob die Lösung passt und umgesetzt wird.

1.12 Was bedeutet SADT + Beispiel.

SADT = Structured Analysis and Design Technique

Beschreibt Aktivitäten und Informationsflüsse. z.B. Bauteilbearbeitung

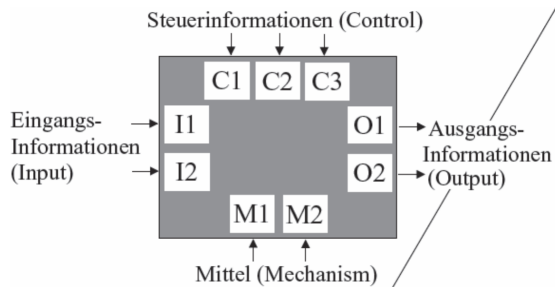


Abbildung 5: SADT Allgemein

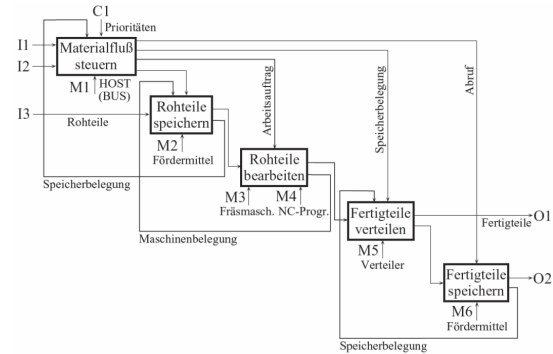


Abbildung 6: SADT Praxisbeispiel

1.13 Was sind Swimlane-Diagramme + Beispiel

Sind eine Art von Flussdiagrammen, welche zeigen wer in einem Prozess zuständig ist.

Verantwortungsbereiche = Swimlanes

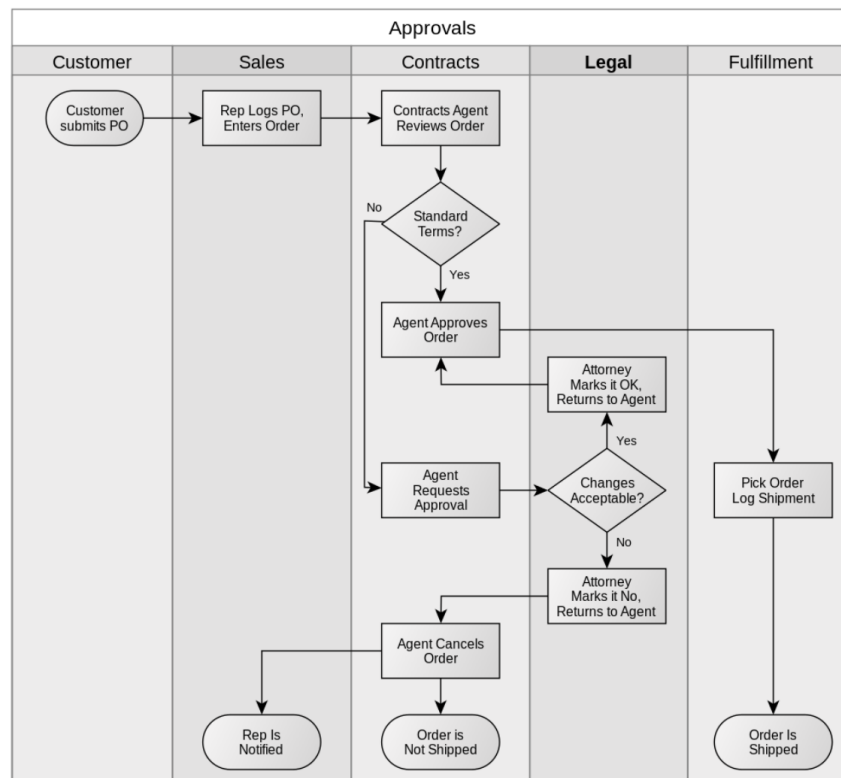


Abbildung 7: Swimlane Diagramm

1.14 Vier Grundregeln des Agilen Manifests

Grundlagen - Theoretisch

- Individuen und Interaktionen haben Vorrang zu Prozessen und Werkzeugen
- Funktionsfähige Produkte haben Vorrang zu ausgedehnter Dokumentation
- Zusammenarbeit mit Kunden hat Vorrang vor Vertragshandlungen
- Reagieren auf Änderungen hat Vorrang vor strikter Planverfolgung

Grundlagen - Praxisbeispiele

- Arbeiten im Team
- Fokus auf technische Aufgabenstellung
- Denke an Kunden
- Flexibel für Änderungen

1.15 Bedeutung SCRUM + Rollen

SCRUM = Vorgehensmodell der agilen Softwareentwicklung.

Geht davon aus, dass Entwicklungen aufgrund ihrer Komplexität nicht im Vorraus detailliert planbar sind.

Stärken: überschaubares Rahmenwerk mit wenig Rollen, Artefakten und Ereignissen.

Rollen

- Product Owner
 - Für wirtschaftlichen Erfolg des Produktes verantwortlich.
 - Vertritt Kundeninteressen. Sind aber nicht identisch.
- Entwickler Team
 - Keine hierarchischen Strukturen.
 - Setzt Anforderungen von Product Owner um.
- Scrum Master
 - Moderiert alle Ereignisse und sorgt für störungsfreies Arbeiten.
 - Kein Teammitglied und nicht weisungsbefugt.
 - Stellt Einhaltung von Scrum-Regeln sicher.

2 Teil

beinhaltet folgende Foliensätze:

- Teil 2: Systemanforderungen (Requirements Engineering)

2.1 Was versteht man unter einem Requirement?

Ein **Requirement** definiert geforderte oder gewünschte Eigenschaften.

Betrifft **Funktionalität**, ein **Qualitätsmerkmal**, eine **Bedingung** oder **Fähigkeit**.

2.2 Was ist ein Stakeholder?

Ist eine **Person** oder **Organisation**, welche Einfluss (direkt oder indirekt) auf die Anforderungen des Systems hat.

2.3 Wie kann man Requirements Engineering und Requirements Management unterscheiden?

Requirements Engineering

- Entwickeln der Anforderungsbasis.
- Analysieren, Ermitteln, Strukturieren, Spezifizieren von Anforderungen.

Requirements Management

- Arbeiten mit Anforderungen.
- Dokumentieren, Ändern, Rückverfolgen, Versionieren von Anforderungen.

2.4 Wie werden Anforderungen angegeben? —nicht sicher—

Anforderung

- Geforderte Eigenschaft bezogen auf Produkt oder Entwicklung.
- Formal durch Merkmale und Ausprägungen ausdrücken.
- Repräsentiert ein Entwicklungsziel.

Merkmal

- Beschreibt Bezugsobjekt der Anforderung (z.B. Namen).

Ausprägung

- Bezeichnet den Sollwert für das Anforderungsmerkmal.
- Beinhaltet bei quantitativen Anforderungen einen Wertebereich und Einheit.

- Beinhaltet bei qualitativen Anforderungen einen verbalen Ausdruck.

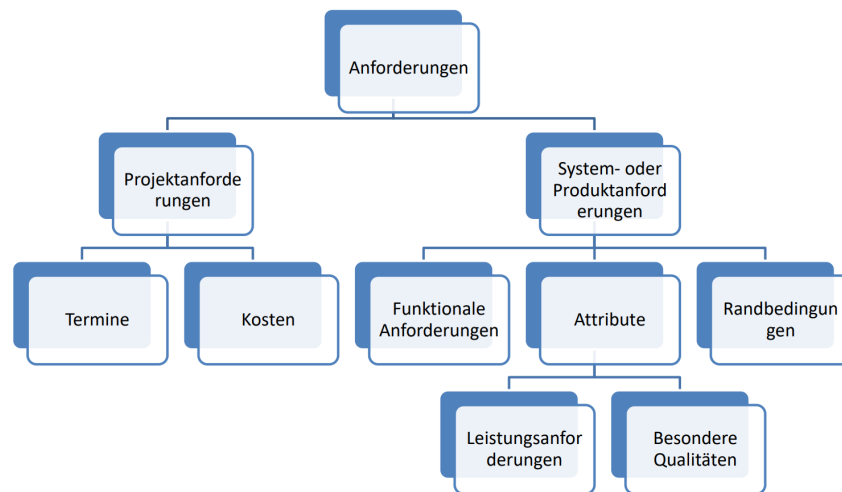


Abbildung 8: Klassifikation von Anforderungen

2.5 Nennen Sie Funktionale Anforderungen. (Zusatz!)

Beschreiben funktionale Aspekte des Systems.

- Fragestellung: **Was tut das System**
- Beschreibt Funktionen, Eingaben, Ausgaben (Daten, Fehlermeldungen, ...) des Systems.
- Beschreiben wichtige Systemzustände und das Verhalten des Systems.
- Bsp.: Der Alarm soll ausgelöst werden, sobald der Sensor das Zerbrechen der Fensterscheibe detektiert.

2.6 Nennen Sie Nicht-Funktionale Anforderungen.

- **Technologische Anforderungen**
 - schränken Lösungsraum des Systems ein.
 - Definieren vorgegebene Lösungswege.
- **Qualitätsanforderungen**
 - Definieren die Qualität des Produktes.
- **Anforderungen an Benutzeroberfläche**
 - Anforderungen an die Bedienung des Produktes.
- **Nebenprodukte**

- Notwendig für Funktion bzw. Verständnis des Produktes (z.B. Doku).
- Unterpunkt2
- **Anforderungen an durchzuführende Tätigkeiten**
 - Tätigkeiten, welche nach dem Entwicklungsprozess anfallen (z.B. Wartung)
- **Rechtlich- Vertragliche Anforderungen**
 - Anforderungen zwischen Vertragspartnern vor Entwicklungsbeginn.

2.7 Was sind typischen Schritte im Requirements Engineering?

- Anforderungen **erheben** (z.B. Befragung von Stakeholdern)
- Anforderungen **dokumentieren** (z.B. Lastenheft)
- Anforderungen **verifizieren** (z.B. Hinblick auf Spezifikation)
- Anforderungen **validieren** (z.B. Hinblick auf Zweck oder Anforderungen)
- Anforderungen **verwalten** (z.B. Administrative Werkzeuge wie: Freigeben, Ändern, Rückverfolgen)

2.8 Nennen Sie Quellen für Anforderungen?

3 Teilbereiche zur Gewinnung von Anforderungen

- Identifizierung relevanter Anforderungsquellen
- Gewinnung von existierenden Anforderungen
- Entwicklung von innovativen Anforderungen

Methoden

- Interviews
- Workshops
- Beobachtungen
- Schriftliche Befragung
- Perspektivenbasiertes Lesen

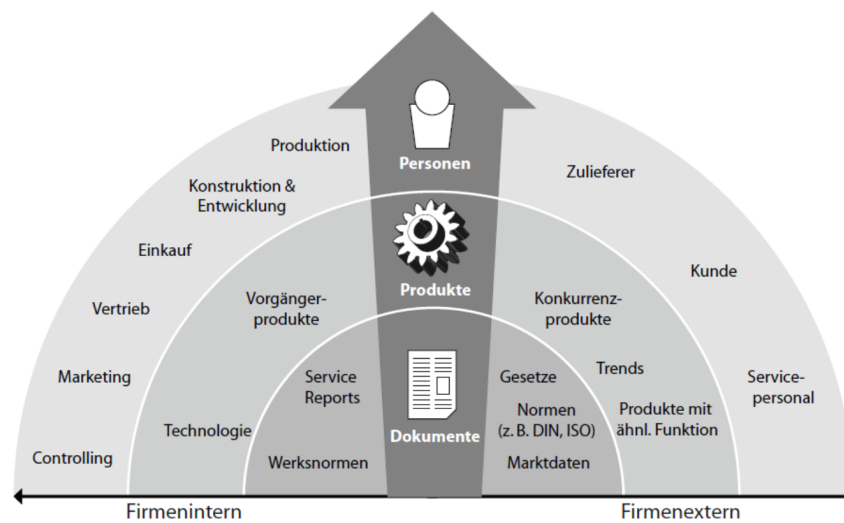


Abbildung 9: Klassifikation von Anforderungen

2.9 Was sind Qualitätskriterien für Anforderungen?

Smarte Anforderungen

- Spezifiziert
- Messbar
- Akzeptiert
- Realisierbar
- Terminiert
- Eindeutig
- Vollständig
- Bewertet

2.10 Was sind typische Bestandteile einer Anforderungsliste?

Inhalte der Anforderungsliste

- Forderungen
 - Müssen erfüllt werden.
 - Können auch als Mindset oder Maximalforderungen definiert.
- Wünsche
 - Sollen erfüllt werden.
 - Aufgabenlösung auch ohne Erfüllung der Wünsche.

- Quantitative Angaben
 - Forderungen und Wünsche so weit wie möglich mit konkreten Werten angeben (Eindeutige Interpretation).
- Qualitative Angaben
 - Besondere Anforderungen oft ohne definierte Werte angegeben.
- Administrative und ordnenden Angaben
 - Benutzer
 - Projektbezeichnung
 - relevante Datum
 - Ersteller
 - Änderer

2.11 Mit welcher Methode können Zielkonflikte erkannt werden und erklären Sie diese!

Methode : Korrelationsmatrix

Gründe für Korrelationsmatrix

- Bereinigung der Anforderungsliste um Inkonsistenzen und Redundanzen.
- Analyse von Wechselbeziehungen zwischen Anforderungen und Identifikation von Zielkonflikten.
- Priorisierung von Anforderungen.
- Wechselbeziehung zwischen Anforderungen eintragen.
- Gewichtung zur Darstellung der Stärke der Abhängigkeit .
- Matrix oft symmetrisch.

Anforderungen	Hohe Saugleistung	Hohe Betriebsdauer	Einfache Entleerung	Geringe Lautstärke	Ergonomisches Handling	Geringes Gewicht	Ansprechendes Design	Angemessener Preis
Hohe Saugleistung								
Hohe Betriebsdauer	-							
Einfache Entleerung								
Geringe Lautstärke	-							
Ergonomisches Handling			+					
Geringes Gewicht	--				++			
Ansprechendes Design					+			
Angemessener Preis	-						-	

+ positive Beeinflussung - negative Beeinflussung
 ++ stark positive Beeinflussung -- stark negative Beeinflussung

Abbildung 10: Beispiel Korrelationsmatrix

3 Teil

beinhaltet folgende Foliensätze:

- Teil 3a: Produktarchitektur, Modularisierung
- Teil 3b: Design Structure Matrix

3.1 Welche drei Schritte sind zur Bildung einer Modulare Produktarchitektur relevant?

- **Identifikation der Komponenten**
 - Produkt in Komponenten zerlegt, um Module zu bilden.
 - Granularität beschreibt den Grad der Zerlegung.
 - Granularität muss zu Komplexität und Zielen passen.
- **Analyses der Abhängigkeiten von Komponenten**
 - Interaktionen oder Abhängigkeiten der Komponenten untereinander analysieren.
 - z.B.: Mechanische Schnittstellen mit modularer Bauweise - mechanische Interaktion der Komponenten analysieren.
- **Gruppierung zu Modulen**
 - Komponenten zu Module zusammenfassen.

3.2 Welche Funktions- und Bausteinarten in Baukästen kennen Sie?

- **Grundfunktionen**
 - Grundlegende, immer wiederkehrende Funktionen.
 - Mit anderen Grundfunktionen zur Bildung der Gesamtfunktion.
- **Hilfsfunktionen**
 - Funktion Verbinden und Anschließen.
 - Kopplung zwischen Funktionsbausteinen.
- **Sonderfunktionen**
 - Spezielle, ergänzende oder spezifische Teilfunktionen zu Bildung der Gesamtfunktion.
- **Anpassungsfunktionen**
 - Schnittstelle zur Umwelt mit anderen Systemen.

- Auftragsspezifisch anzupassen.
- System wird mechanisch, elektrisch und informationstechnisch mit Umwelt verbunden.
- **Auftragsspezifische Funktionen**
 - Für projektspezifische Funktionen, welche nicht standardmäßig angeboten werden.
- **MUSS – Funktionen**
 - Umfassen Grund-, Hilfs-, und Anpassungsfunktionen.
 - Zwingend notwendig zur Bildung der Gesamtfunktion.
- **KANN – Funktionen**
 - Aus Sonderfunktionen gebildet.
 - Nice to Have-Elemente.

3.3 Was sind Vor- und Nachteile von Produktplattformen?

Vorteile

- Einfache Erstellung von Varianten. Anpassung an geänderte Varianten.
- Abdeckung weiterer Marktsegmente bei geringem Aufwand.
- Komponenten trotzdem in großen Stückzahlen produzierbar (Kostenvorteil).

Nachteile

- Zusätzlicher Aufwand für die Verwaltung der Varianten.
- Hoher Planungsaufwand zu Beginn.

3.4 Was versteht man unter Kundenindividuelle Massenproduktion?

Kompromiss aus Massenfertigung und individueller Anpassung. Massenproduktion ist effizient, aber nicht auf Kundenwunsch anpassbar. Einsatz von Produktplattformen. z.B. Autoindustrie: Selbes Modell in vielen unterschiedlichen Varianten (Farbe, Ausstattung, ...).

3.5 Was ist ein Graph und welche Arten gibt es?

- Mathematische Grundlage zur Darstellung eines beliebigen Netzwerks.
- Ursprung: Königsberger Brückenproblem.

- Zur generischen Beschreibung von Netzwerken.

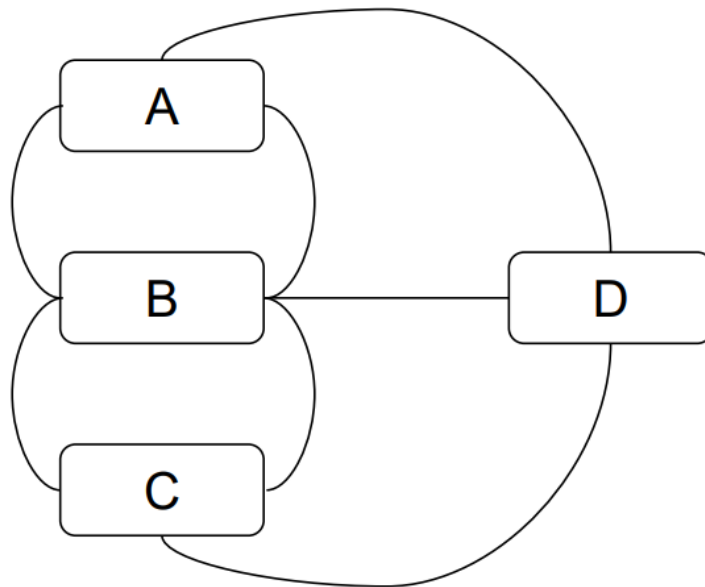


Abbildung 11: Graph allgemein

- **Eigenschaften:**

- Gerichtet, Ungerichtet oder beides
- Knoten als auch Kanten können gewichtet sein.
- Kante kann zu Ursprungsknoten führen.
- Keine, eine oder mehrere Kanten können zwei Knoten verbinden.
- Offene oder lose Enden.

- **Charakteristiken:**

- Unterscheidung Knoten und Kanten.
- Zwei Kanten angrenzend, wenn sie sich einen Knoten teilen.
- Knoten aneinander angrenzend, wenn sie sich eine Kante teilen.
- Eine Kante kann zu einem Knoten führen oder weggehen.
- Anzahl der Kanten zu einem Knoten bestimmt dessen Ordnung.

- **Arten von Graphen:**

- Ungerichtet, Gerichtet, Beides

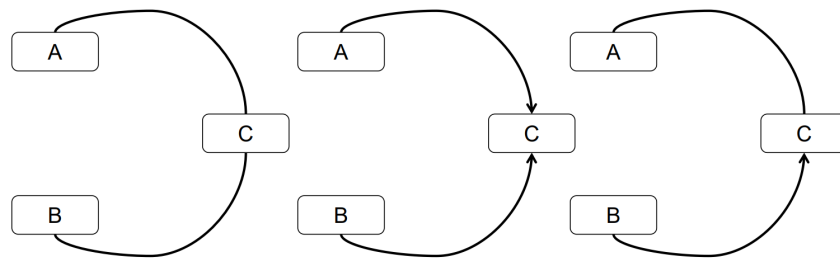


Abbildung 12: Arten von Graphen

3.6 Erklären Sie einen gewichteten Graphen?

Ein Gewichteter Graph hat **Gewichtungsfaktoren** für die Knoten und Kanten.
z.B. Kosten, Bearbeitungs- und Transportzeit

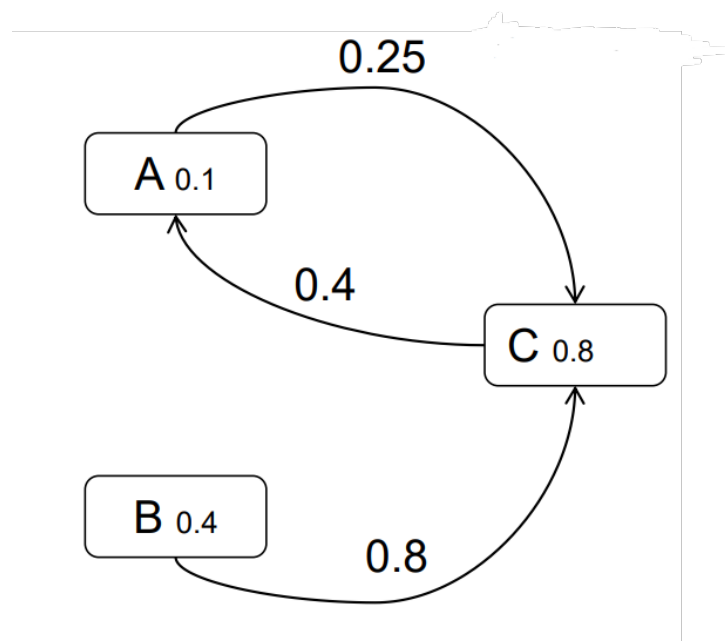


Abbildung 13: Beispiel Gewichteter Graph

3.7 Wofür steht der Begriff DSM?

Design Structure Matrix

- Zeigt Verbindungen zwischen Systemelementen in Matrixform.
- Elemente (z.B. Aktivitäten eines Prozesses) in Zeilen und Spalten einer symmetrischen Matrix anhand logischer Abhängigkeiten aufgetragen.
- Leserichtung aufgetragen: **Zeile beeinflusst Spalte**
- Diagonale bleibt unbeachtet.

- DSM innerhalb einer Domäne modellieren, um Aussagefähigkeit zu gewährleisten.
- **Vorteile:**
 - Einfache und präzise Möglichkeit komplizierte Systeme darzustellen.
 - Erlaubt Einsatz von leistungsfähigen Analysemöglichkeiten (Clustering, Sequenzierung, Matrixanalysen).
- **Nachteile:**
 - Knoten referenzieren nicht auf Kanten.
 - Eigenschaften der Kanten können nur bedingt dargestellt werden.
 - Zeitliche Änderungen der Struktur schwierig darzustellen.
 - Schwierig zu managen bei hierarchischer Aufteilung.

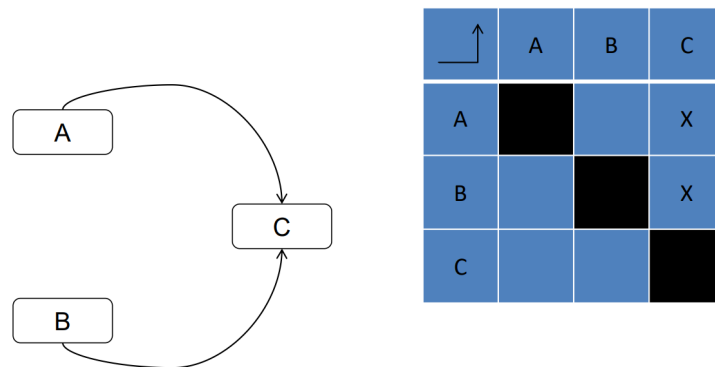


Abbildung 14: Beispiel Matrix Representation gerichteter Graph

3.8 Welche Arten von DSM gibt es?

- **Komponentenbasiert**
 - Verbindungen von Komponenten.
 - Anwendung in Systemarchitektur, Systems Engineering und System Design.

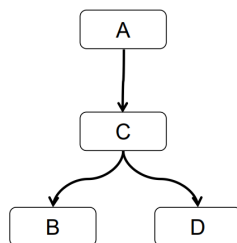
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Radiator	1		X														
Engine fan	2	X				X											
Heater core	3			X													X
Heater hoses	4				X												
Condenser	5		X			X		X									
Compressor	6				X	X		X	X								
Evaporator case	7					X	X										X
Evaporator core	8				X	X		X									X
Accumulator	9					X		X	X								
Refrigeration controls	10									X							
Air controls	11										X						
Sensors	12											X					
Command distribution	13												X				
Actuators	14													X			
Blower controller	15															X	X
Blower motor	16			X				X	X						X	X	

Abbildung 15: Komponentenbasierte DSM - Beispiel

• Personenbasiert

- Verbindung von Organisationseinheiten.
- Organisationsentwurf, Kommunikationsmanagement und Teamintegration.

– z.B. Organigramm



	A	B	C	D
A			X	
B				
C		X		X
D				

Abbildung 16: Personenbasierte DSM - Beispiel

• Aktivitätsbasiert

- Verbindungen von Aktivitäten (Input / Output).
- Prozessverbesserung, Projektzeitplanung, Iterationsmanagement und Informationsflussmanagement.

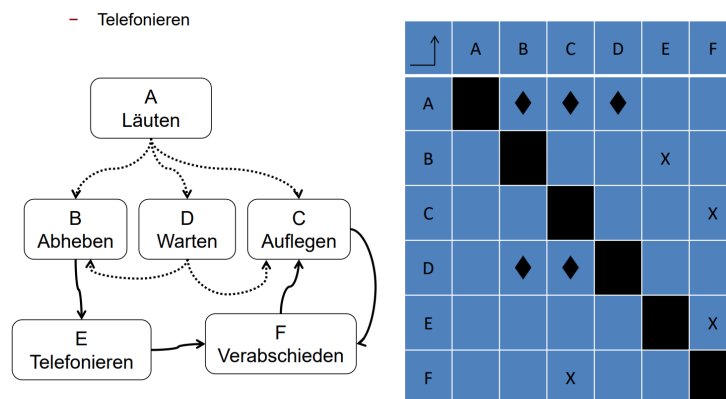


Abbildung 17: Aktivitätenbasierte DSM - Beispiel

- **Parameterbasiert**

- Verbindungen von Design-Parametern.
- Aktivitätsabfolgen und Prozessentwurf.

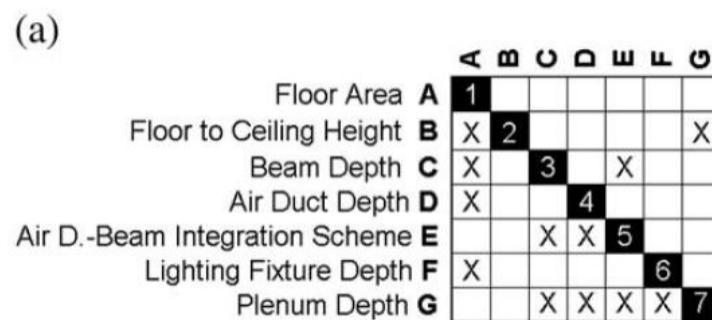


Abbildung 18: Parameterbasierte DSM - Beispiel

3.9 Was bedeutet Sequenzierung von DSM?

Sequenzierung ist die Umordnung der Zeilen und Spalten der DSM mit dem Ziel, dass es keine Schleifen gibt (Dreiecksmatrix).

- Identifikation der Elemente ohne Information anderer Elemente ausgeführt werden könne.
Werden **nach links geschoben**.
- Identifikation die keine Information für andere Elemente liefert (Gegenteil von Punkt 1).
Werden **nach rechts geschoben**.
- Wenn Matrix nicht leer ist existiert mindestens eine Schleife.
Identifikations-Methoden:
 - Pfad Suchen.
 - Adjazent Matrix.

- Erreichbarkeits Matrix Methode.
- Triangulations Algorithmus.
- Algorithms von Tarjan zur Bestimmung eines minimalen Spannungsbaumes.
- Beispiel einer Sequenzierung:

Sequenzierung Beispiel Schritt 1 und 2

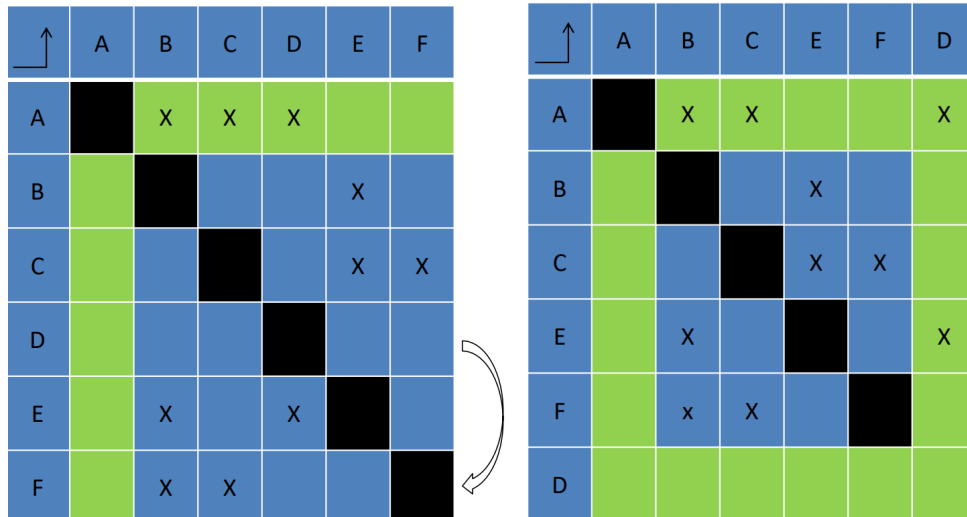


Abbildung 19: Sequenzierung Schritt 1 und 2 - Beispiel

Sequenzierung Beispiel Schritt 3

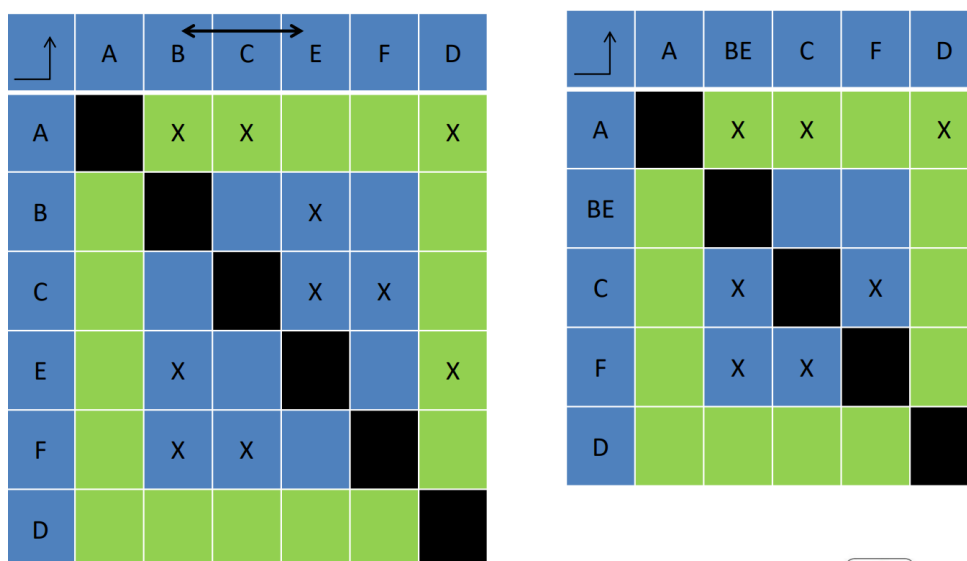


Abbildung 20: Sequenzierung Schritt 3 - Beispiel

Sequenzierung Beispiel Wiederholung Schritt 2

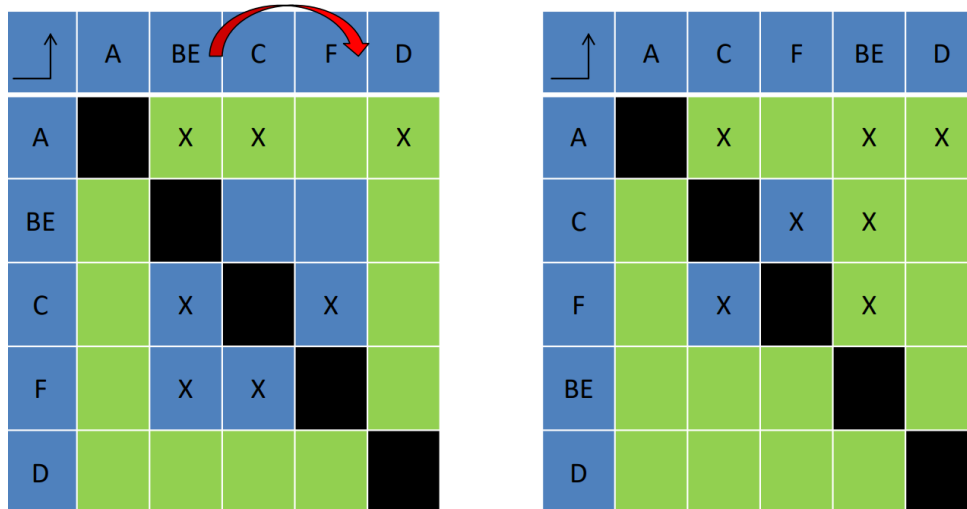


Abbildung 21: Sequenzierung Wiederholung Schritt 2 - Beispiel

Sequenzierung Beispiel Ergebnis

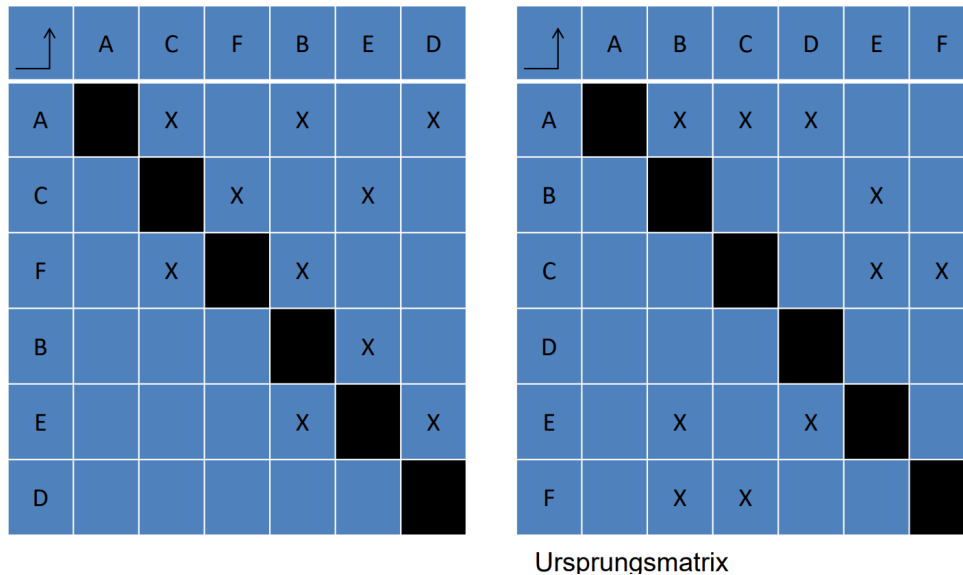


Abbildung 22: Sequenzierung Ergebnis - Beispiel

3.10 Was bedeutet Clustering von DSM?

Ziel ist es bei der Komponenten- bzw. Personenbasierten DSM **Cluster oder Module** zu identifizieren, in denen ein **Großteil der Interaktionen** stattfinden.

Clustering ermöglicht bessere Einblicke in die Teamorganisation und **Identifikation von Schlüsselparameter**.

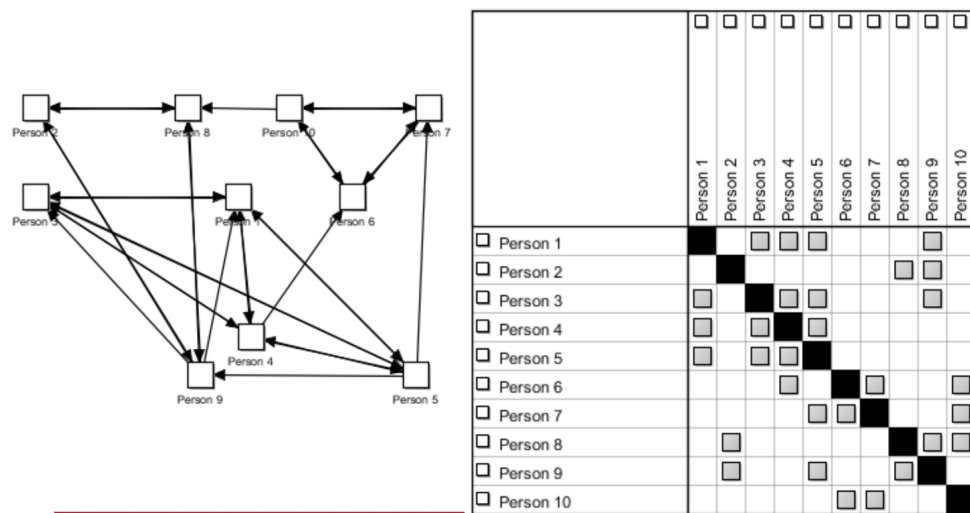


Abbildung 23: Clusterung - Beispiel

4 Teil

beinhaltet folgende Foliensätze:

- Teil 4: Systemverhalten (Grundlagen der Modellplanung und –Bildung und Simulation)

4.1 Was ist ein Modell?

Modelle sind Abstraktionen und Vereinfachung der Realität und zeigen deshalb nur Teilaspekte auf. Es ist daher wichtig, dass die Modelle im Hinblick auf die Situation und die Problemstellung aussagekräftig sind. Dies bedeutet, dass bei allen Überlegungen die Frage nach der Zweckmäßigkeit und der Problemrelevanz zu stellen ist.

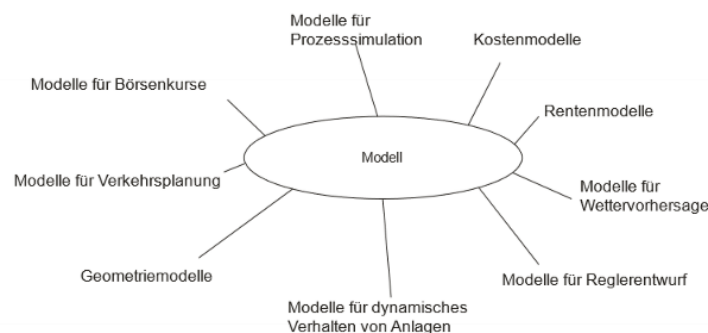


Abbildung 24: Beispiel Modelle verschiedener Lebensbereiche

4.2 Was bedeutet Problemlösung durch Abstraktion (Modellbildung) und Interpretation?

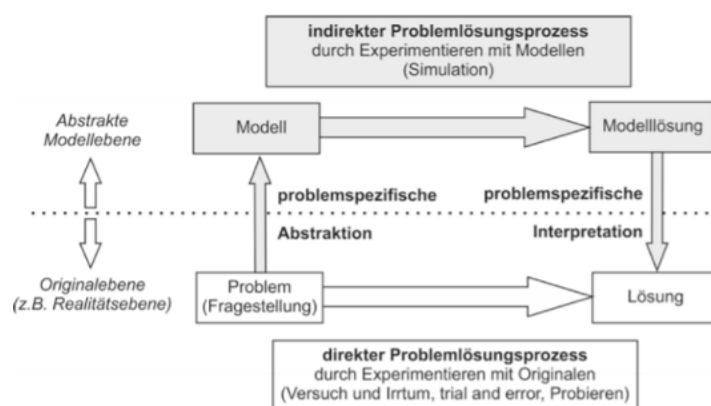


Abbildung 25: Problemlösung durch Abstraktion und Interpretation

4.3 Welche Verfahren der Modellbildung gibt es?

- **Rechnerische Verfahren:**

Dazu werden **mathematische Modelle** benötigt, die formal durch **Gleichungen** (algebraisch oder differenzial) beschrieben werden, zu deren Lösung heute neben den traditionellen analytischen Verfahren leistungsfähige numerische und symbolische Softwareprogramme zur Verfügung stehen.

- **Experimentelle bzw. messtechnische Verfahren:**

Zu ihrer ANwendung werden **physikalische (experimentelle) Modelle** benötigt, an denen Versuche, Messungen und Auswertungen durchgeführt werden können. Im Maschinenbau etwa sind dies typischerweise Prototypen, Testobjekte, VERSUCHSAnordnungen oder maßstäbliche Modelle. Mit Hilfe der physikalischen Modelle sollen alle wesentlichen Einflüsse "messtechnisch" erfasst werden.

- **Hybride Verfahren** (Kombination von Berechnung und Experiment bzw. Messung):

Diese Verfahren **nutzen sowohl Messgrößen als auch mathematische Modelle**. Während bei rechnerischen Verfahren Fehler aufgrund von Modellierungsungenauigkeiten auftreten, sind bei experimentellen Verfahren mehr oder weniger große Messfehler unvermeidbar. Liegen sowohl mathematische Modelle als auch Messergebnisse vor, so kann man versuchen, Hypothesen über die Art der Fehler zu bilden und das Modell anhand der Messergebnisse zu verbessern, dass sich eine bessere Übereinstimmung von Rechnung und Experiment ergibt.

4.4 Wie ist der Ablauf zur Entstehung eines rechnerinternen Modells?

- Der Modellbildner entwickelt eine gedankliche Vorstellung des zu untersuchenden Originals (z.B. eines realen technischen Objekts oder eines neuen Produkts) in Form eines mentalen Modells (Gedankenmodells), das anschließend zu einer Erfassung in eine formalisierte Informationsform mit Hilfe von Informationselementen und -strukturen gebracht wird.
- Dieses „Informationsmodell“ wird am Rechner implementiert (rechnerinternes Modell).

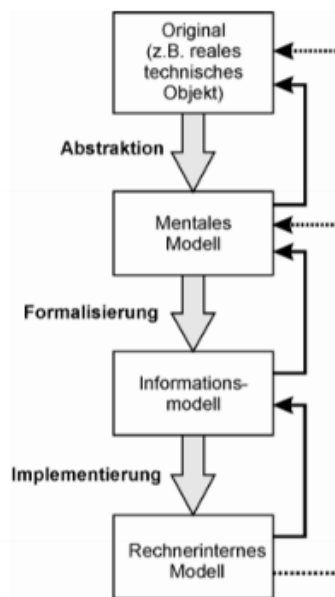


Abbildung 26: Ablauf zur Entstehung eines rechnerinternen Modells

4.5 Was ist bei der Interpretation von Ergebnissen wichtig?

Eine wichtige Aufgabe des Produktentwicklers (Versuchstechnikers, Berechnungsingenieurs) ist die Interpretation der Ergebnisse. Dazu muss er zwischen physikalischen Phänomenen und künstlichen Effekten (Artefakten), die z.B. von Messfehlern bzw. numerischen Lösungsverfahren herrühren, unterscheiden können. Dies erfordert Kenntnisse und Erfahrung sowohl über die untersuchten Fragestellungen als auch über die (z.B. versuchstechnischen bzw. numerischen) Verfahren, die zur Lösung des Modellproblems verwendet werden. Einfache, überschlägige Abschätzungen und Erfahrung sind unerlässlich, um die Ergebnisse (Messergebnisse bzw. Rechenergebnisse) zu überprüfen sowie zu bewerten und damit hohe Qualität der Simulation sicherzustellen.

4.6 Was bedeutet Simulation und wofür ist dies relevant?

In der VDI-Richtlinie 3633 mit dem Titel „Simulation von Logistik-, Materialfluss- und Produktionssystemen – Begriffsdefinitionen“ [VDI-3633] ist Simulation wie folgt definiert:

- Simulation ist ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden. Mit Hilfe der Simulation kann das zeitliche Ablaufverhalten komplexer Systeme untersucht werden (Simulationsmethode).

Simulationen werden dann durchgeführt wenn:

- kein reales System verfügbar ist (Entwurfsphase)
- das Experiment am realen System zu lange dauert

- das Experiment am realen System zu teuer ist
- das Experiment am realen System zu gefährlich ist (Flugzeug, Kraftwerk)
- die Zeitkonstanten des realen Systems zu groß sind (Klimamodelle)

5 Teil

beinhaltet folgende Foliensätze:

- Teil 5: Digitaler Zwilling

5.1 Erklären Sie in eigenen Worten den Begriff Digitaler Zwilling!

Ein Digitaler Zwilling ist ein virtuelles Abbild eines realen Objekts oder Systems. Er besteht aus dem realen Objekt, dem digitalen Modell und den Daten, die beide miteinander verbinden, um das reale Objekt in Echtzeit oder für Simulationen abzubilden.

5.2 Was ist ein Digitaler Schatten?

Ein Digitaler Schatten beschreibt die Echtzeitdarstellung von Betriebs-, Zustands- oder Prozessdaten eines realen Objekts oder Systems, ohne dass ein vollständiges Modell vorliegt. Er wird durch die Daten aus der realen Welt generiert.

Ein Digitaler Schatten ist eine Echtzeitdarstellung der Betriebs- oder Prozessdaten eines realen Objekts oder Systems. Im Gegensatz zum Digitalen Zwilling umfasst er kein vollständiges Modell, sondern nur die kontinuierlich erfassten Daten, die den aktuellen Zustand des Systems widerspiegeln. Diese Daten ermöglichen die Überwachung und Analyse des realen Objekts ohne Modellierung des gesamten Verhaltens.

5.3 Was bedeutet Functional Mock-Up Interface?

Das Functional Mock-Up Interface (FMI) ist eine standardisierte Schnittstelle, die den Austausch und die Integration von Simulationsmodellen zwischen verschiedenen Softwareumgebungen ermöglicht. Es erlaubt die Kosimulation, bei der Modelle in verschiedenen Tools gemeinsam ausgeführt werden, ohne dass das Modell in jedem Tool neu erstellt werden muss.