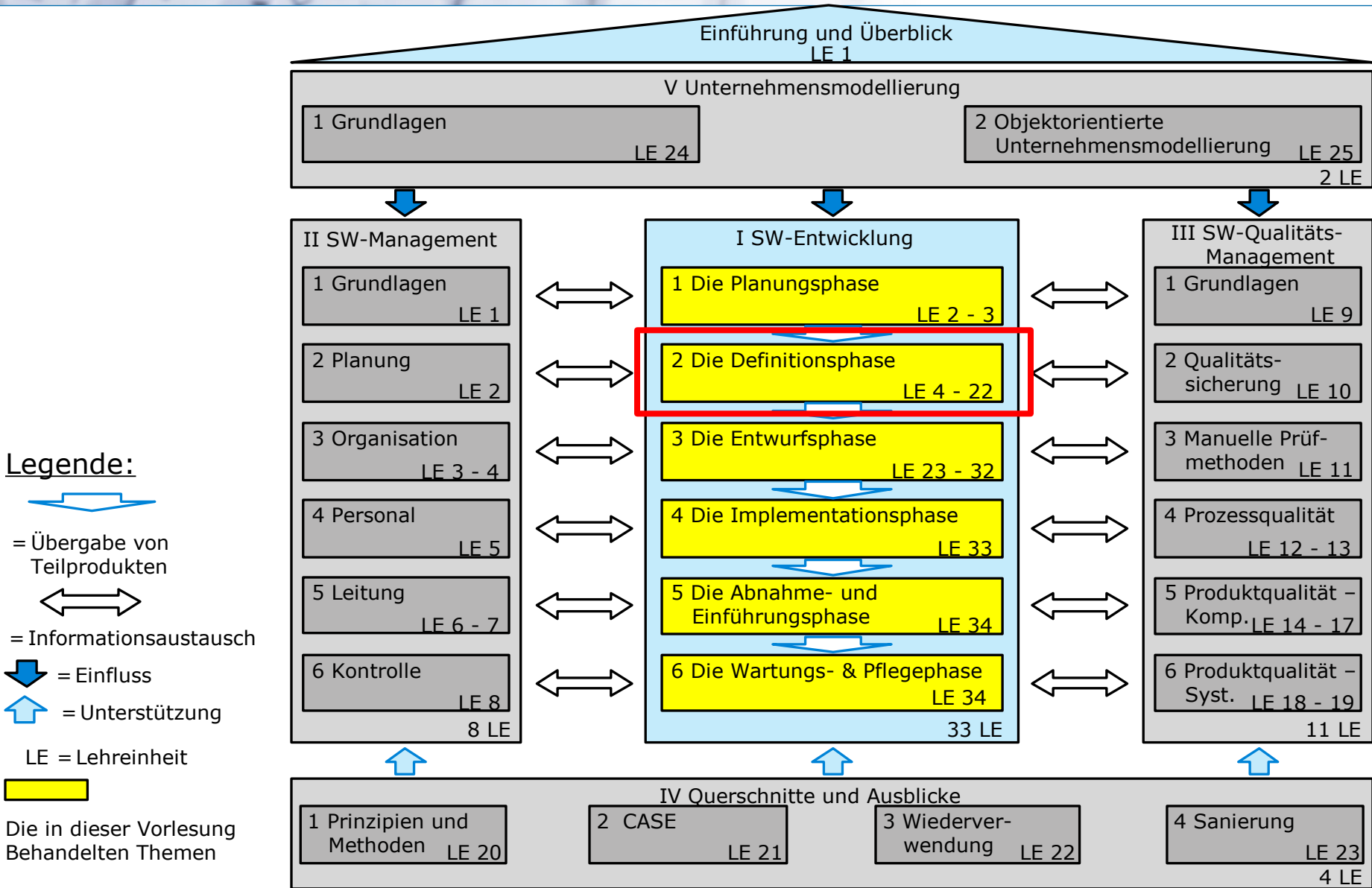


Vorlesung Softwaretechnik - Definitionsphase, Einführung -

Prof. Klaus-Peter Fährnich

Wintersemester 2008/2009



Überblick LE 4

- LE 4: Einführung
 - Lernziele
 - Einführung und Überblick
 - Modellierung
 - Verbale Anforderungen
 - Pflichtenheft
 - Zusammenfassung

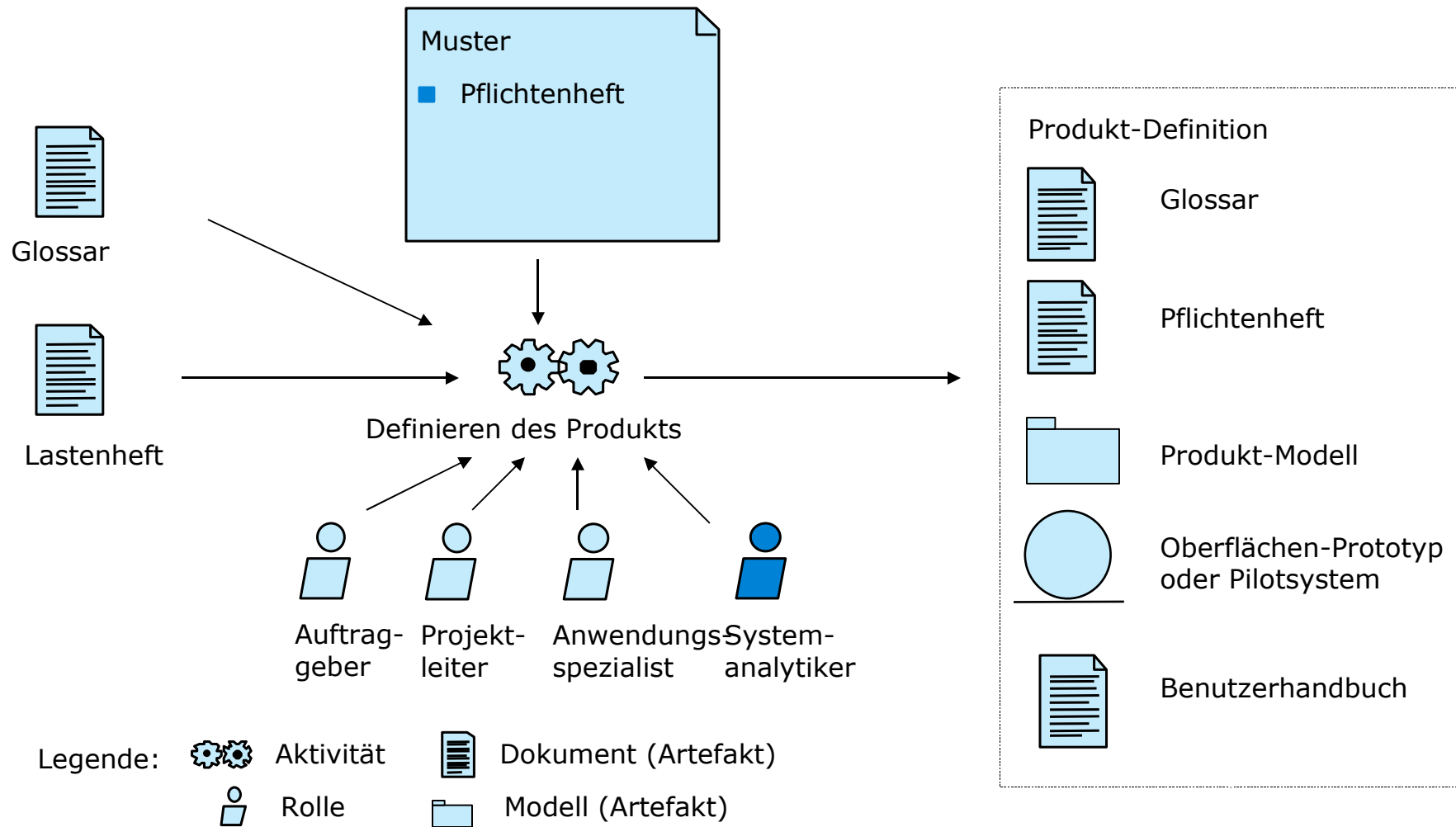
Lernziele

1. Definition der Qualitätsziele Vollständigkeit, Konsistenz und Eindeutigkeit sowie von Durchführbarkeit;
2. Komplexitätsarten Funktionen, Daten, Algorithmen, zeitabhängiges Verhalten, Systemumgebung und Benutzungsoberfläche bezogen auf eine Anwendung beschreiben können;
3. Basiskonzepte und kombinierte Methoden;
4. Zusammensetzung der kombinierten Methoden aus Basismethoden;
5. Vorgehensweise und Tätigkeiten beim Definitionsprozess;
6. Funktion eines Pflichtenheftes;
7. Erstellung eines Pflichtenheftes.

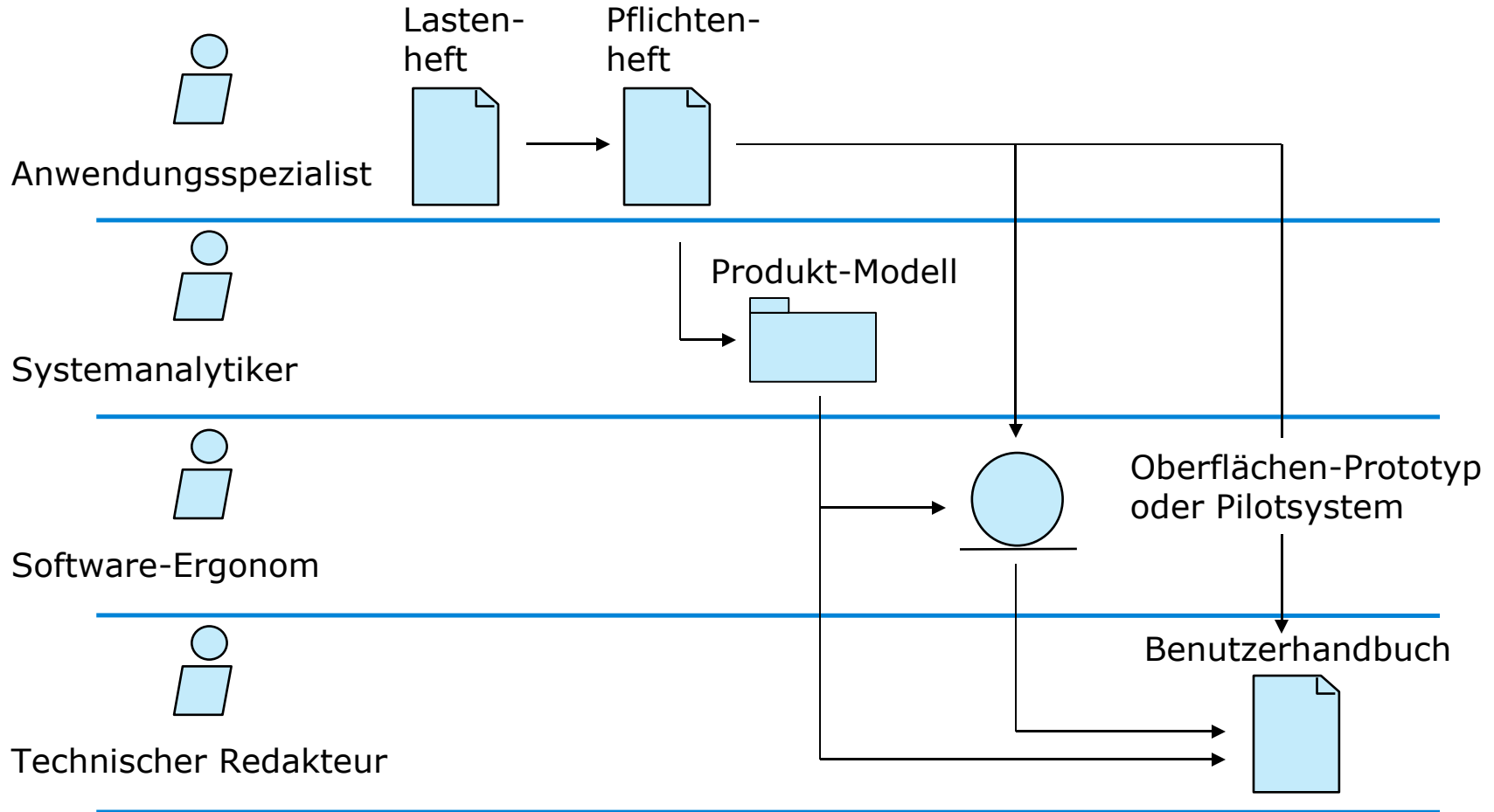
Definieren von Produkt-Anforderungen

- **Anforderungen (requirements):** Legen die qualitativen und quantitativen Eigenschaften eines Produkts aus der Sicht des Auftraggebers fest.
- **Systemanalyse (requirements engineering):** Systematische Vorgehensweise, um die Anforderungen in einem iterativen Prozess zu ermitteln.
- Der iterative Prozess „Definition des Produkts“ beinhaltet folgende Aktivitäten:
 - Anforderungen ermitteln und beschreiben;
 - Anforderungen als fachliche Lösung modellieren;
 - Anforderungen analysieren;
 - Anforderungen evtl. animieren, simulieren und ausführen;
 - Anforderungen verabschieden;
 - Ziel des Definitionsprozesses ist die Überführung der Anforderungen ans Produkt in einem **vollständigen, konsistenten** und **eindeutigen Produktmodell**.

Überblick Definitionsphase



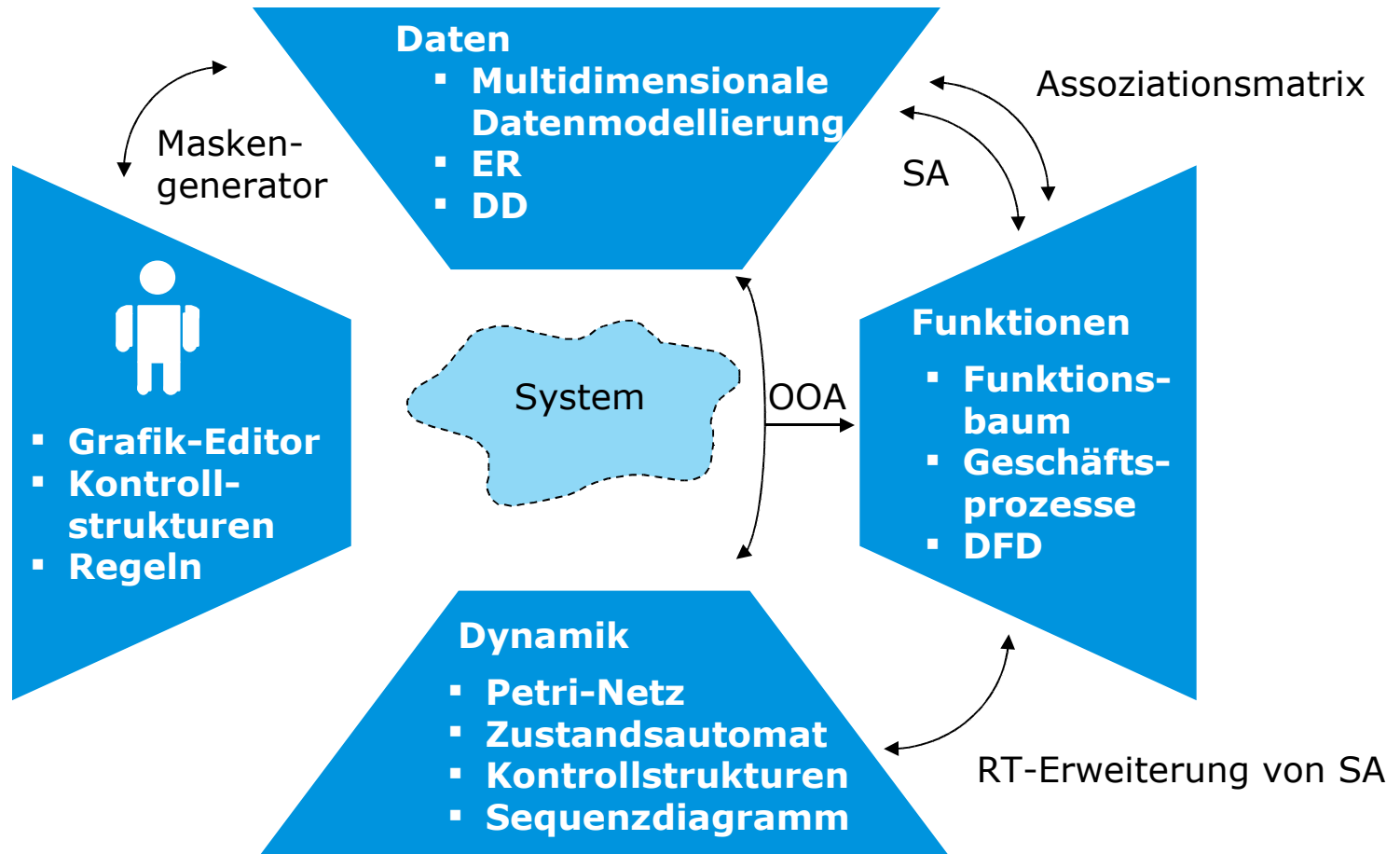
Aktivitäten in der Definitionsphase



Produkt-Modell

- In der Software-Technik ist ein **Modell** eine idealisierte und vereinfachte Darstellung eines Weltausschnitts mit dem Ziel dessen Eigenschaften besser studieren zu können.
- **Ist-Analyse:** Neu zu entwickelnde Software löst oft vorhandenes System ab. Ist-Analyse zeigt Schwachstellen des vorhandenen Systems. Diese können dann bei der Formulierung der neuen Anforderungen berücksichtigt werden.
- Anwendungsspezialist und Systemanalytiker kümmern sich aktiv um die Ermittlung der **Anforderungen**. Anforderungen bekommen sie schriftlich. Weitere Infos in der Regel durch Interviews und Befragungen.
- **Produktmodell**
 - **Vollständigkeit:** Beschreibt den Grad in dem das Produkt dem Benutzer alle notwendigen Funktionen und Daten selbst zur Verfügung stellt.
 - **Konsistenz:** Beschreibt den Grad in dem die definierten Anforderungen untereinander widerspruchsfrei sind
 - **Eindeutigkeit:** Beschreibt den Grad in dem die definierten Anforderungen genau eine Interpretation erlauben
- **Produkt-Definition** setzt sich meistens aus verschiedenen Artefakten zusammen. Ist ein juristisches Dokument. Mit ihrer Unterzeichnung sind Anforderungen ans Produkt verabschiedet.

Sichten und Methoden



Legende:



= Benutzungs-
oberfläche

ER = Entity Relationship

DD = Data Dictionary

DFD = Datenflussdiagramm

RT = Realtime Analysis

OOA = Object Oriented Analysis

SA = Structured Analysis

Software-Komplexität

- Anwendungen lassen sich nach **Komplexitätsarten** gliedern.
Wir unterscheiden 6 Komplexitätsarten:
 - Komplexität der Funktionen
 - Komplexität der Daten
 - Komplexität der Algorithmen
 - Komplexität des zeitabhängigen Verhalten
 - Komplexität der Systemumgebung
 - Komplexität der Benutzungsoberfläche

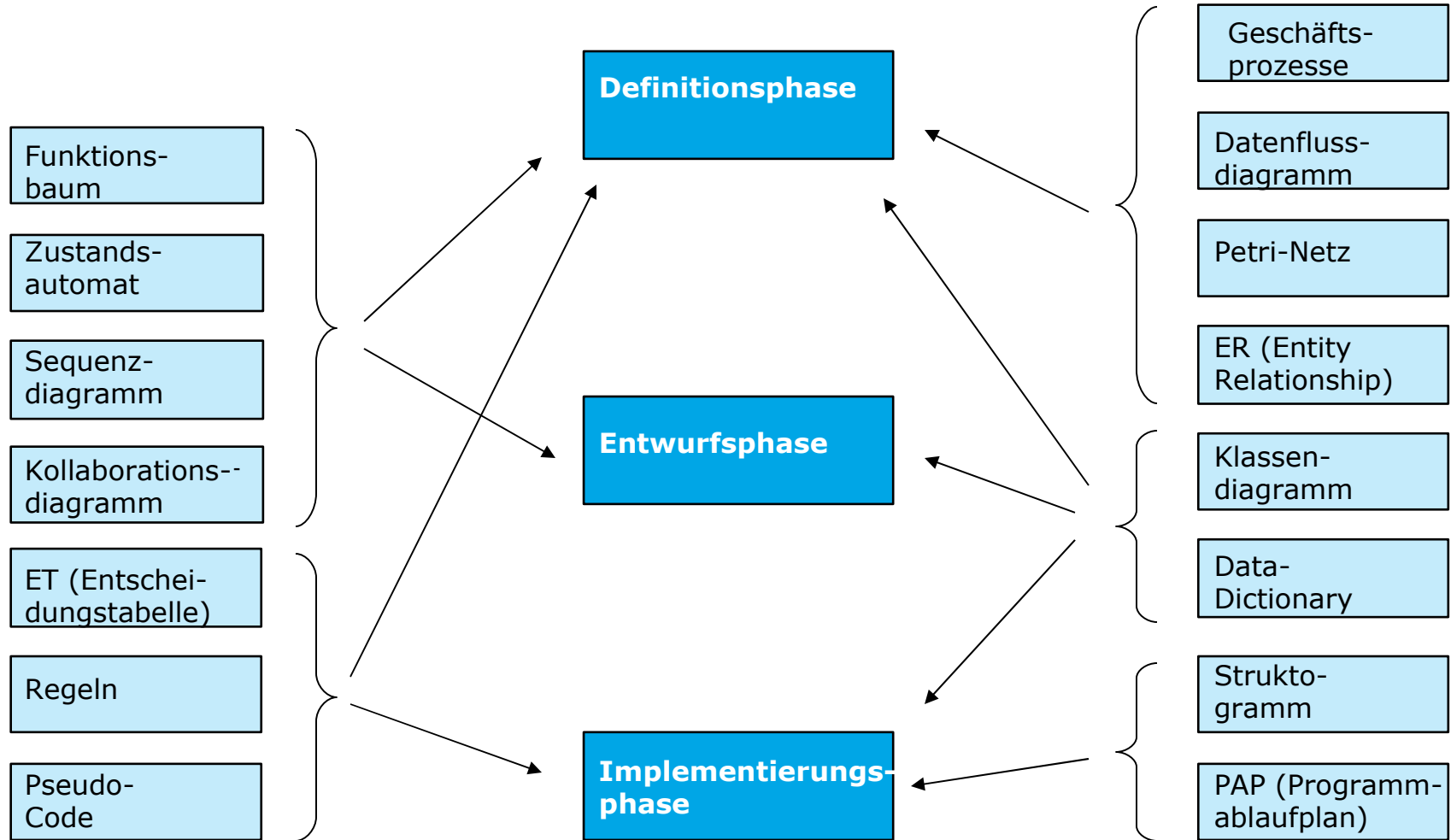
Software-Basiskonzepte (1)

- Zur Beschreibung des Produkt-Modells benutzt man **Basiskonzepte**. Ein Basiskonzept ist:
 - ein atomares Konzept,
 - konzeptionell langlebig,
 - phasenübergreifend verwendbar und
 - in unterschiedlichen Kontexten einsetzbar.
- Meisten Basiskonzepte recht alt (ungefähr 10)
- Abstraktionsniveau der Basiskonzepte unterschiedlich, je nachdem in welchen Phasen das Konzept eingesetzt wird.
- Voraussetzung für erfolgreiche Produkt-Definition ist der Einsatz geeigneter Basiskonzepte.
- Neue Entwicklungen haben dazu geführt, dass verschiedene Basiskonzepte in geeigneter Weise zu kombinierten Konzepten für die Systemanalyse zusammengesetzt wurden. Im Mittelpunkt steht heute:
 - OOA (Object Oriented Analysis)
- Als Stand der Technik sind anzusehen:
 - SA (Structured Analysis)
 - RT (Real Time Analysis)

Software-Basiskonzepte (2)

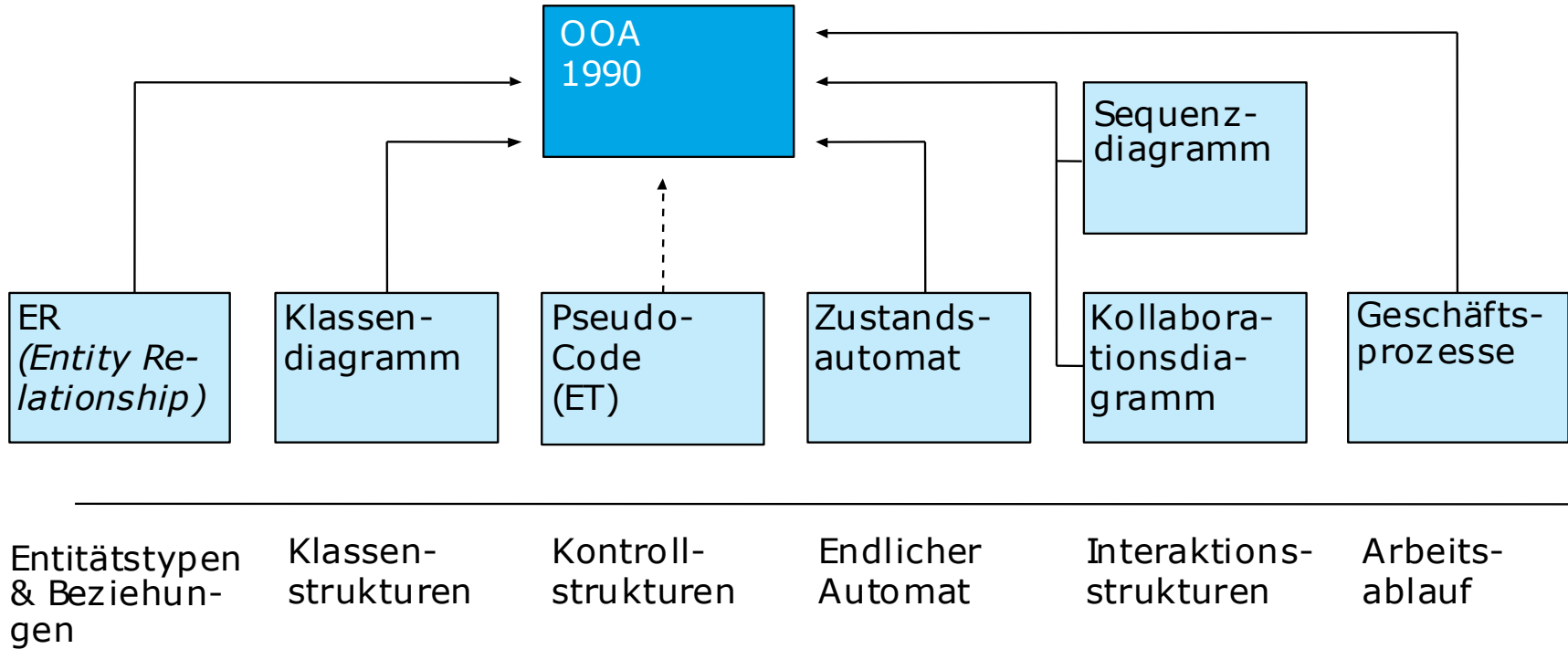
<p>Konzepte und Sichten</p> <p>↑ häufig verwendet</p> <p>↓ selten verwendet</p>										
Funktionsbaum	Geschäftsprozess (1987)	Daten-Flussdiagramm (1966)	Data Dictionary (1979)	ER (Entity Relationship) (1976)	Klassendiagramm (1980/90)	Pseudocode	Regeln	Zustands-Automat (1954)	Petri-Netz (1962)	Sequenzdiagramm (1987)
Funktionale Hierarchie	Arbeitsablauf	Informationsfluss	Datenstrukturen	Entitätstypen & Beziehungen	Klassenstrukturen	Kontrollstrukturen	wenn-dann Strukturen	Endlicher Automat	Nebenläufige Strukturen	Interaktionsstrukturen
Funktionale Sicht			Datenorientierte Sicht		Objektorientierte Sicht	Algorithmische Sicht	Regelbasierte Sicht	Zustandsorientierte Sicht		Szenario-basierte Sicht
LE5			LE8		LE6-7	LE9	LE9-10	LE11-12		LE7

Software-Basiskonzepte (3)



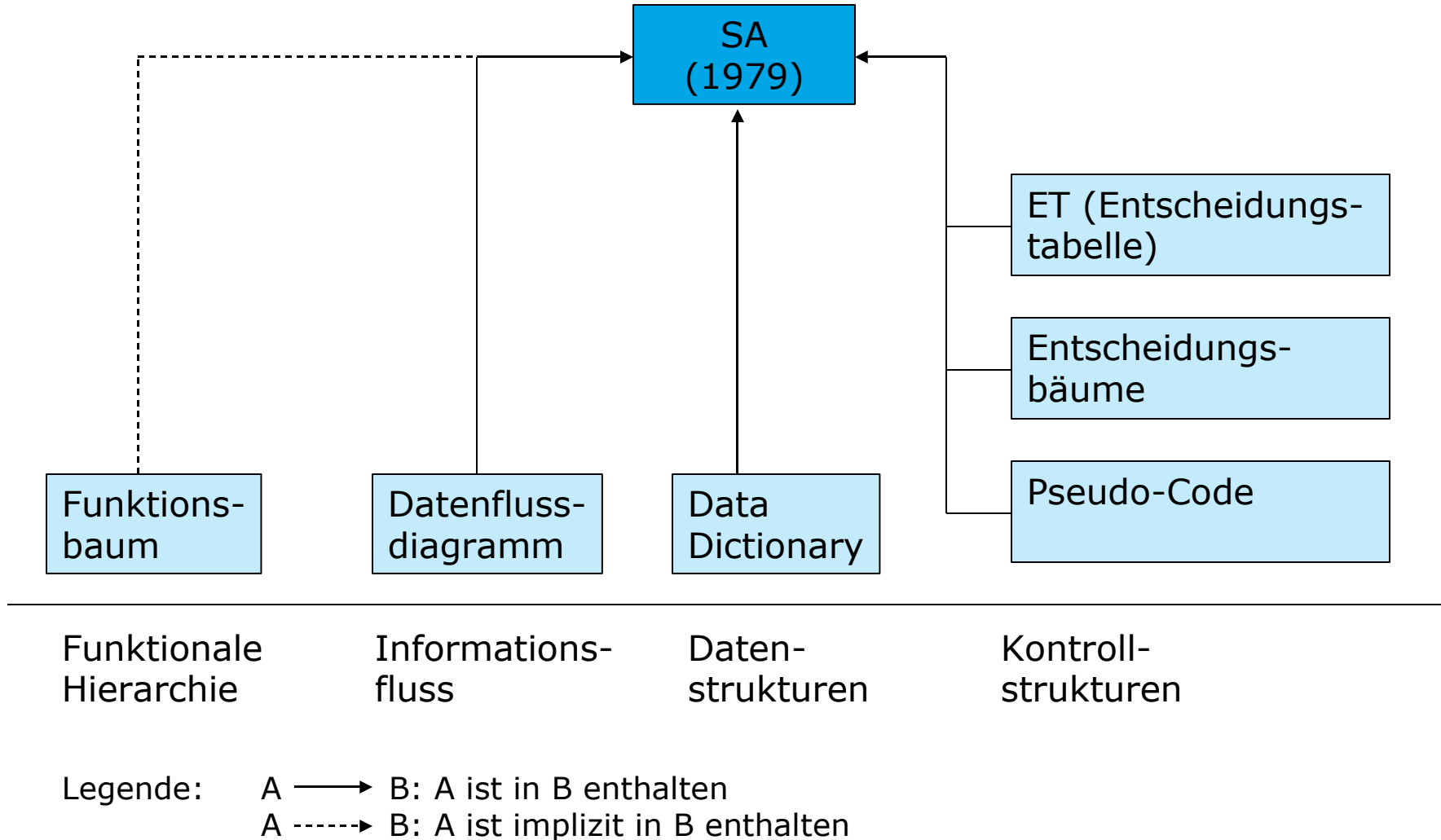
Legende: A → B: A wird in B eingesetzt

Object Oriented Analysis

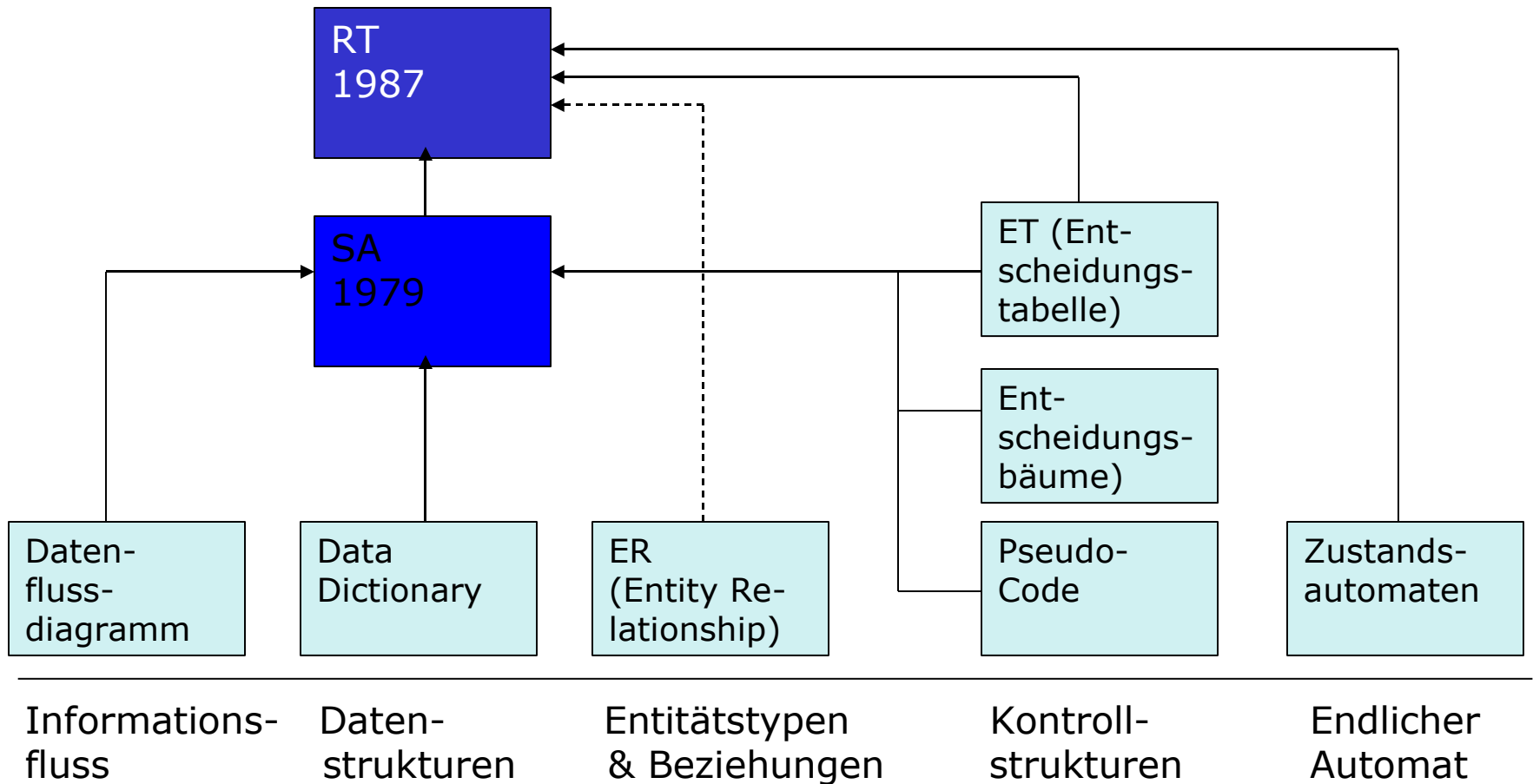


Legende: A → B: A ist in B enthalten
 A - - - - -> B: A ist implizit in B enthalten

Structured Analysis

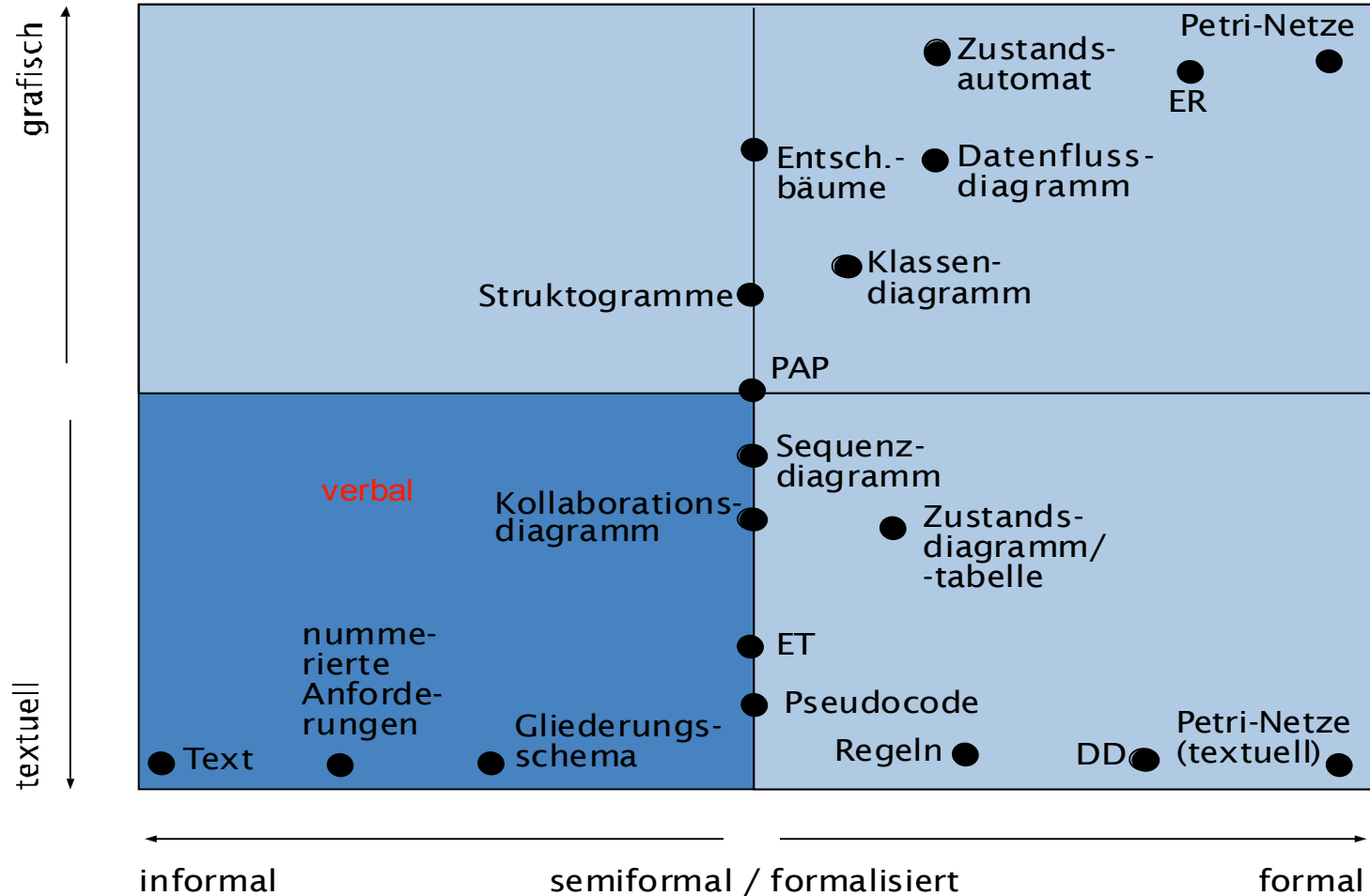


Real Time Analysis



Legende: A → B: A ist in B enthalten
A - - - - -> B: A ist implizit in B enthalten

Verbale Anforderungen (1)



Verbale Anforderungen (2)

- **Textuelle Beschreibungen** legen die Anforderungen in Form von natürlich-sprachlichen Texten fest.
- **Grafische Darstellungen** legen Anforderungen durch grafische Symbole und Linien zwischen den Symbolen fest. Diese werden durch textuelle Symbole bezeichnet.
- Unter **formaler Beschreibung** versteht man die Verwendung formaler, textueller Sprachen.
- **Semi-formalisierte Beschreibung** sind stärker formalisiert als die Umgangssprache aber nicht streng mathematisch wie eine formale Sprache.
- **Verbale Beschreibungen** sind nochmals gegliedert in:
 - Beschreibungen ohne festgelegte Regeln;
 - Beschreibungen mit nummerierten oder markierten Anforderungen;
 - Beschreibungen mit festgelegten Schemata.

Pflichtenheft

- **Pflichtenheft** ist detaillierte verbale Beschreibung der Anforderungen an ein neues Produkt.
- **Funktion** eines Pflichtenhefts
 - **Aufgabe:** Enthält eine Zusammenfassung aller fachlichen Anforderungen, die das zu entwickelnde Software-Produkt aus der Sicht des Auftraggebers erfüllen muss. Grundlage des Vertrages.
 - **Adressaten:** Auftraggeber, Auftragnehmer und Anwendungsspezialisten, Systemanalytiker, Entwerfer, Qualitätssicherer, Benutzerrepräsentant.
 - **Inhalt:** stellen in der Regel Konkretisierung und Detaillierung der Lastenheft-Inhalte dar.
 - **Form:** Vorgegebenes, standardisiertes, grobes Gliederungsschema mit festgelegten Inhalten.
 - **Sprache:** Detaillierte verbale Beschreibung mit Nummerierung einzelner Anforderungen. Nummerierung notwendig um sich in anderen Dokumenten und in späteren Phasen drauf beziehen zu können.
 - **Didaktik:** Das Gliederungsschema ist so aufgebaut, dass das Pflichtenheft gut lesbar ist, und eine leichte Einarbeitung erlauben.
 - **Zeitpunkt:** Erstes Dokument was nach Abschluss der Planungsphase erstellt wird.
 - **Umfang:** Die notwendigen Anforderungen müssen in ausreichender Detaillierung beschrieben werden. Beschreibung des was, nicht des wie.

Pflichtenheft – Gliederung (1)

1. Zielbestimmung
 1. Musskriterien
 2. Wunschkriterien
 3. Abgrenzungskriterien
2. Produkteinsatz
 1. Anwendungsbereiche
 2. Zielgruppen
 3. Betriebsbedingungen
3. Produktübersicht
4. Produktfunktionen
5. Produktdaten
6. Produktleistungen
7. Qualitätsanforderungen
8. Benutzungsoberfläche
9. Nichtfunktionale Anforderungen

Pflichtenheft – Gliederung (2)

10. Technische Produktumgebung

1. Software
2. Hardware
3. Orgware
4. Produkt-Schnittstellen

11. Spezielle Anforderungen

1. Software
2. Hardware
3. Orgware
4. Entwicklungs-Schnittstellen

12. Gliederung in Teilprodukte

13. Ergänzungen

Pflichtenheft – Beispiel (1)

Pflichtenheft HIWI-Verwaltung

1 Zielbestimmung

- Das Programm HIWI-Verwaltung soll einen Lehrstuhl in die Lage versetzen, die beschäftigten Hilfsassistenten rechnergestützt zu verwalten.

1.1 Musskriterien

- Verwalten von HIWI-Daten
- Erstellen einer Liste aller Hilfsassistenten
- Ausgabe aller Daten eines Hilfsassistenten

1.2 Wunschkriterien

- Ausdrucken von Arbeitszeugnissen
- Automatische Generierung einer Verlängerung der HIWI-Verträge um drei Monate zum Quartalsende
- Berechnung der benötigten drei Gehälter für alle HIWIs bis zum Jahresende
- Serienanschreiben zwecks Einladung zur Weihnachtsfeier
- Gezielte Abfragen mit einer Endbenutzersprache

Pflichtenheft – Beispiel (2)

1.3 Abgrenzungskriterien

- Software ist nicht netzwerkfähig
- keine Überprüfung der Arbeits-/Urlaubszeiten der HIWIs

2 Produkteinsatz

- Das Produkt wird im Sekretariat eines Lehrstuhls zur Verwaltung der Hilfsassistenten eingesetzt

2.1 Anwendungsbereich

- Hilfsassistentenverwaltung
- Abfragen (kommerzieller Anwendungsbereich)

2.2 Zielgruppen

- Mitarbeiter/-in eines Lehrstuhls (Sekretär/in)

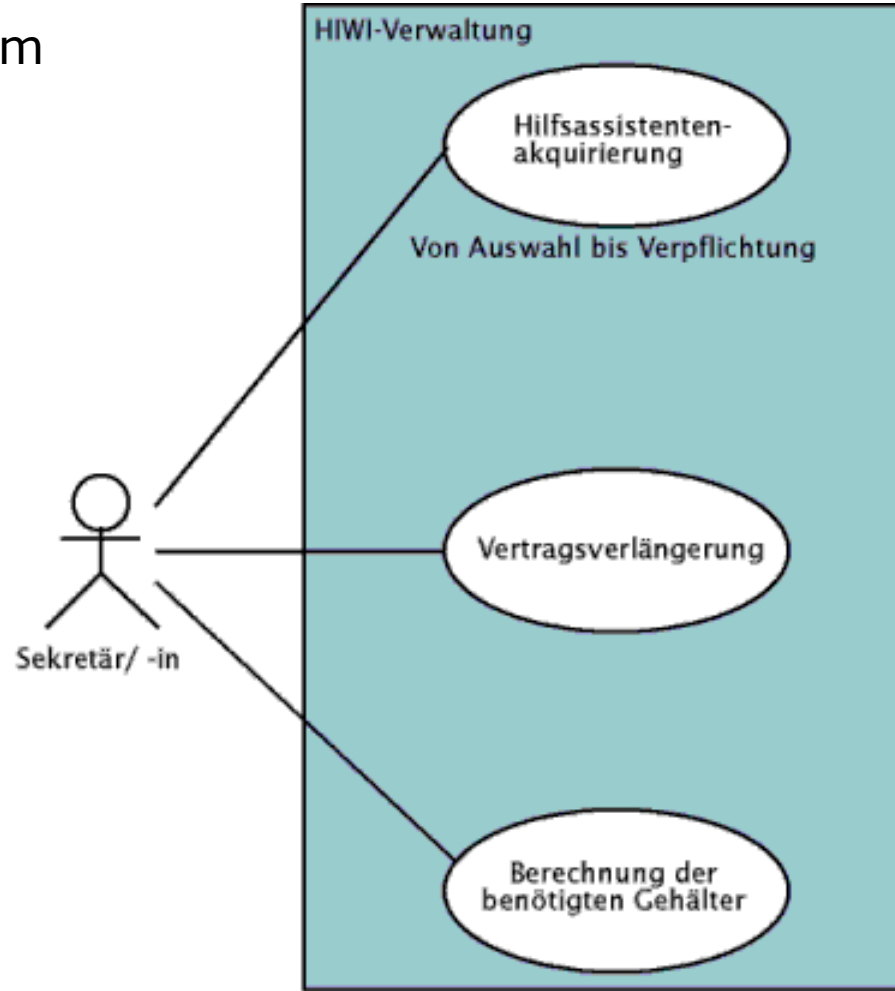
2.3 Betriebsbedingungen

- Büroumgebung

Pflichtenheft – Beispiel (3)

3 Produktübersicht

- Umweltdiagramm



Pflichtenheft – Beispiel (4)

4 Produktfunktionen

4.1 Geschäftsprozesse

- /F10/ **Geschäftsprozess**: Hilfsassistentenakquirierung: Von Auswahl bis Verpflichtung
Akteur: Sekretärin
Beschreibung: Für den Lehrstuhl neue Hilfsassistenten suchen und als freie Mitarbeiter verpflichten. Dabei Überprüfung, ob er mehr als 19 h/Woche an der Universität als HIWI beschäftigt ist. (HIWIs dürfen höchstens 19 h/Woche an der Universität arbeiten.)
- /F20/ **Geschäftsprozess**: Vertragsverlängerung
Akteur: Sekretärin
Beschreibung: Zum Quartalsende werden alle auslaufenden HIWI-Verträge automatisch um 3 Monate verlängert und ausgedruckt.

Pflichtenheft – Beispiel (5)

- /F30/ **Geschäftsprozess**: Berechnung der benötigten Gehälter
Akteur: Sekretärin
Beschreibung: Zum aktuellen Zeitpunkt wird die Summe der bis zum Jahresende benötigten Gehälter für alle HIWIs berechnet

4.2 Listen

- /F40/ Ausgabe (Drucker + Bildschirm) einer alphabetischen Liste aller Hilfsassistenten mit folgenden Daten: Name, Vorname, Matrikel-Nr., Studienadresse, Geburtsdatum
- /FW50/ Das Programm generiert Steuerdateien mit den Adressen aller Hilfsassistenten, um mit Word Serienbriefe an alle HIWIs zu schicken.

4.3 Berichte

- /F60/ Pro HIWI Ausgabe sämtlicher Daten eines Hilfsassistenten
- /FW70/ Ausdrucken eines Arbeitszeugnisses mit Überprüfung, ob
 1. die Beschäftigungszeit beendet ist
 2. noch kein Zeugnis gedruckt worden ist
- /FW80/ Suchoperationen auf allen Datenfeldern

Pflichtenheft – Beispiel (6)

5 Produktdaten

- /D10/ Personendaten: Name, Vorname, Matrikel-Nr., Geburtsdatum, Heimatadresse, Heimattelefon-Nr., Studienadresse, Studientelefon-Nr.
- /D20/ Studiendaten: (können pro Student mehrere sein) Studienfach, Semesteranzahl, Vordiplomnote + Datum, Diplomnote + Datum
- /D30/ Beschäftigungsdaten: (können pro Student mehrere sein) Beginndatum, Endedatum, Stundenzahl pro Woche, Tätigkeitsgebiete, benutzte Software-Systeme, Arbeitszeugnisdatum

6 Produktleistungen

- /L10/ Die Geschäftsprozess-Funktionen außer /FGW30/, die Listen-Funktionen außer /FLW20/ sowie die Report-Funktionen dürfen nicht mehr als 2 Sekunden Antwortzeit benötigen.
- /L20/ Die Funktionen /FGW30/ und /FLW20/ dürfen nicht mehr als 5 Minuten benötigen (incl. Drucken). Der Arbeitsfortschritt wird auf dem Bildschirm angezeigt.
- /L30/ Es müssen maximal 1.000 Hilfsassistenten verwaltet werden können.

Pflichtenheft – Beispiel (7)**7 Qualitätszielbestimmung**

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		X		
Richtigkeit		X		
Interoperabilität				X
Ordnungsmäßigkeit		X		
Sicherheit		X		
Zuverlässigkeit				
Reife		X		
Fehlertoleranz			X	
Wiederherstellbarkeit			X	
Benutzbarkeit				
Verständlichkeit		X		
Erlernbarkeit		X		
Bedienbarkeit	X			

Pflichtenheft – Beispiel (8)

Produktqualität	sehr gut	gut	normal	nicht relevant
Effizienz				
Zeitverhalten			X	
Verbrauchsverhalten				X
Änderbarkeit				
Analysierbarkeit				X
Modifizierbarkeit			X	
Stabilität			X	
Prüfbarkeit			X	
Übertragbarkeit				
Anpassbarkeit				X
Installierarbeit				X
Konformität				X
Austauschbarkeit				X

Pflichtenheft – Beispiel (9)

8 Benutzungsschnittstelle

- /B10/ Standardmäßig ist eine menüorientierte Bedienung vorzusehen.
- /B20/ Die Bedienungsoberfläche ist auf Mausbedienung auszuliegen; eine Bedienung ohne Maus muss aber auch möglich sein.
- /B30/ DIN 66234, Teil 8 ist zu beachten.
- /B40/ Fensterlayout, Dialogstruktur und Mausbedienung entsprechen dem Windows-Gestaltungs-Regelwerk (style guide)
- /B50/ Sämtliche Daten sind passwortgeschützt und dürfen nur von autorisierten Mitarbeitern des Lehrstuhls bearbeitet werden.

9 Nichtfunktionale Anforderungen

-

10 Technische Produktumgebung

- Das Produkt läuft auf einem Arbeitsplatzrechner mit graphischer Benutzungsoberfläche.

Pflichtenheft – Beispiel (10)

10.1 Software

- Betriebssystem: Windows 95

10.2 Hardware

- PC

10.3 Orgware

- Netzwerkverbindung zum Drucker

10.4 Produkt-Schnittstellen

- Für die Verlängerungen der Arbeitsverträge wird eine Word-Serienbrief-Datei erzeugt.

11 Spezielle Anforderungen an die Entwicklungs-Umgebung

-

12 Gliederung in Teilprodukte

-

13 Ergänzungen

-

Zusammenfassung

- Soll ein neues Software-Produkt erstellt werden, dann müssen die **Anforderungen** an dieses Produkt in der **Definitionsphase** von den Anwendungs-Spezialisten und den Systemanalytikern in Zusammenarbeit mit dem Auftraggeber und den Benutzer-Repräsentanten in Form einer **Produkt-Definition** beschrieben werden.
- Die Systematische Ermittlung, Beschreibung, Modellierung und Analyse der Anforderungen unter Einsatz geeigneter Methoden und Werkzeuge bezeichnet man als **Systemanalyse** (requirement engineering).
- Eine **Produkt-Definition** besteht aus mehreren Dokumenten. Ein Dokument davon ist meist das verbal beschriebenes **Pflichtenheft**. Auf der Grundlage des Pflichtenheftes kann ein **Produkt-Modell** erstellt werden.
- Anhand des Produkt-Modells wird ein Konzept für die **Benutzungsoberfläche** erstellt und in Form eines **Oberfläche-Prototyps** oder eines **Pilot-Systems** umgesetzt.
- Unter Berücksichtigung der bisherigen Informationen wird dann ein **Benutzerhandbuch** angefertigt.

Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg, 2000.
-
- Dorfmann M., Thayer RH., Standards, Guidelines, and Examples on System and Software Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California, 1990
 - IEEE Standards Software Engineering, 1999 Edition, Volume Four: Resource and Technique Standards, N.Y., IEEE, 1999
 - Hesse W. et al., Terminologie der Softwaretechnik, in: Informatik Spektrum 17, 1994, S.39-47
 - Schefe P., Softwaretechnik und Erkenntnistheorie, in: Informatik Spektrum 22, 1999, S. 122-135