Wichtige Termine:

Klausur: 21.01.2009 | Zeit? | Ort?

Auftakttreffen: Softwaretechnik-Praktikum im SS 2010

??.02.20010 | Zeit? | Ort?

Lehrveranstaltungsevaluation: bis 15.02.20010

Vorlesung Softwaretechnik
- Zusammenfassung -

Prof. Dr.-Ing. habil. Klaus-Peter Fähnrich

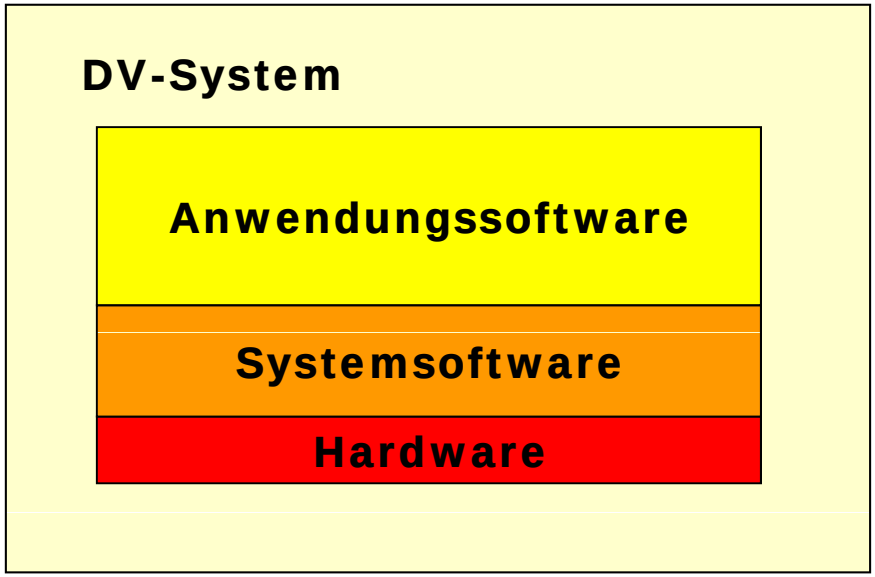
Wintersemester 2009/2010

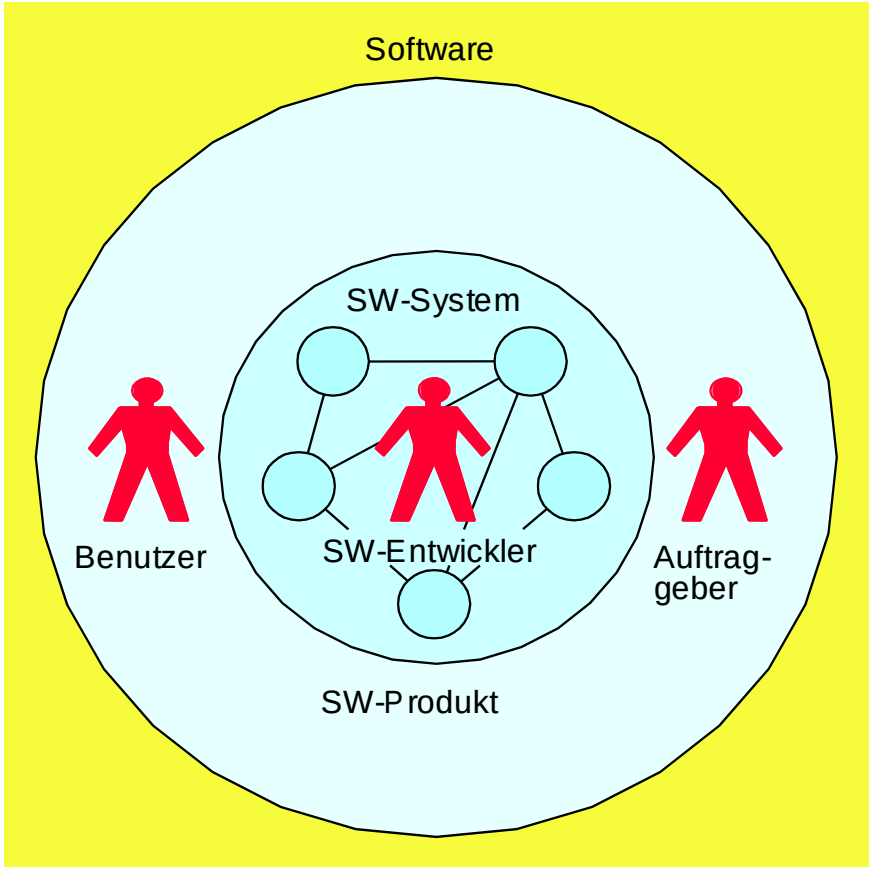
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Software-Definitionen

Software-Technik

Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen.





UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Vorgehensmodell

- Definition **Vorgehensmodell**
  - Ein Vorgehensmodell definiert einen allgemeinen Rahmen für den organisatorischen Prozess der Softwareerstellung
  - **Vorgehensmodelle** legen fest:
    - durchzuführende Aktivitäten
    - Reihenfolge des Arbeitsablaufs (Entwicklungsstufen, Phasen)
    - Definition der Teilprodukte / Ergebnisse (Inhalt, Layout)
    - Fertigstellungskriterien
    - Verantwortlichkeiten und Kompetenzen
    - Notwendige Mitarbeiterqualifikationen
    - Anzuwendende Standards, Richtlinien, Methoden und Werkzeuge
  - Vorgehensmodelle, die die Strukturierung in Phasen besonders betonen, nennt man auch Phasenmodelle

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Warum Vorgehensmodelle ?

- Ziel ist die Kontrolle von Zeit, Budget und Qualität der Ergebnisse
- Planbarkeit von Softwareprojekten durch definierte, **strukturierte** und standardisierte Vorgehensweise
- Optimierung des Entwicklungsprozesses
- Verbesserung der Kommunikation innerhalb des Projekts und nach außen
- Automatisierungsmöglichkeiten durch Werkzeuge

Wasserfallmodell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Merkmale

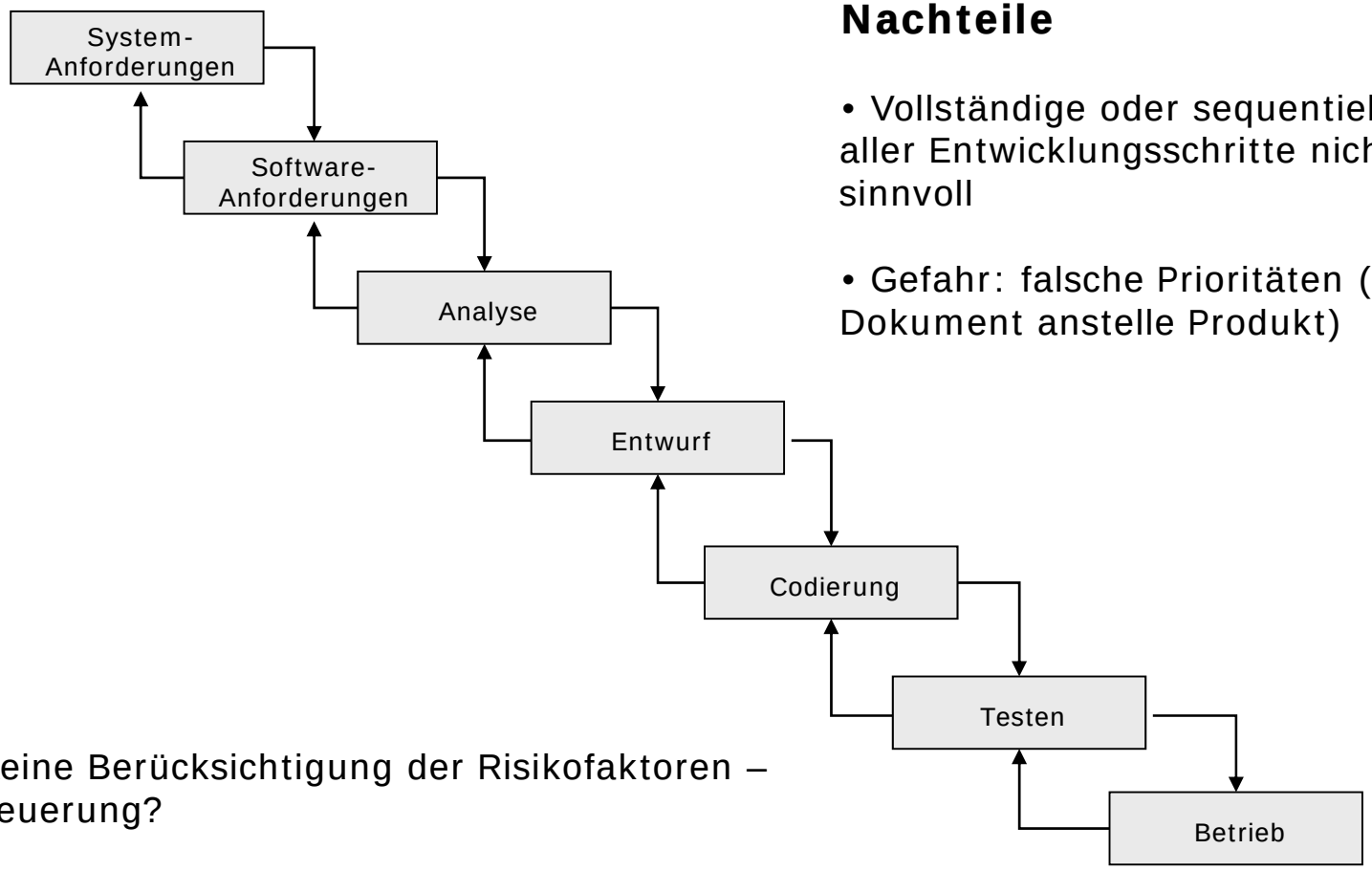
- Weit verbreitet in Varianten in vielen Modellen!
- Mehrere hintereinander liegende ***abgeschlossene Phasen***
- Jede Aktivität ist in der richtigen Reihenfolge und in der vollen Breite vollständig durchzuführen
- Am Ende jeder Aktivität steht ein fertig gestelltes Dokument
- Sequentieller Entwicklungsablauf
- Top-down-Vorgehen

Vorteile

- Einfach, verständlich
- Leichte Fortschrittkontrolle, benötigt wenig Managementaufwand
- zentrales Prinzip: Trenne WAS vom WIE

Wasserfallmodell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Nachteile

• Vollständige oder sequentielle Abwicklung aller Entwicklungsschritte nicht immer sinnvoll

• Gefahr: falsche Prioritäten (Fokus auf Dokument anstelle Produkt)

•Keine Berücksichtigung der Risikofaktoren – Steuerung?

•Berücksichtigung anderer Kriterien (z.B. leicht änderbare Software) schwach ausgeprägt. Fokus sind phasenorientierte Dokumente.

Quelle: Balzert, H.; Lehrbuch der Software-Technik

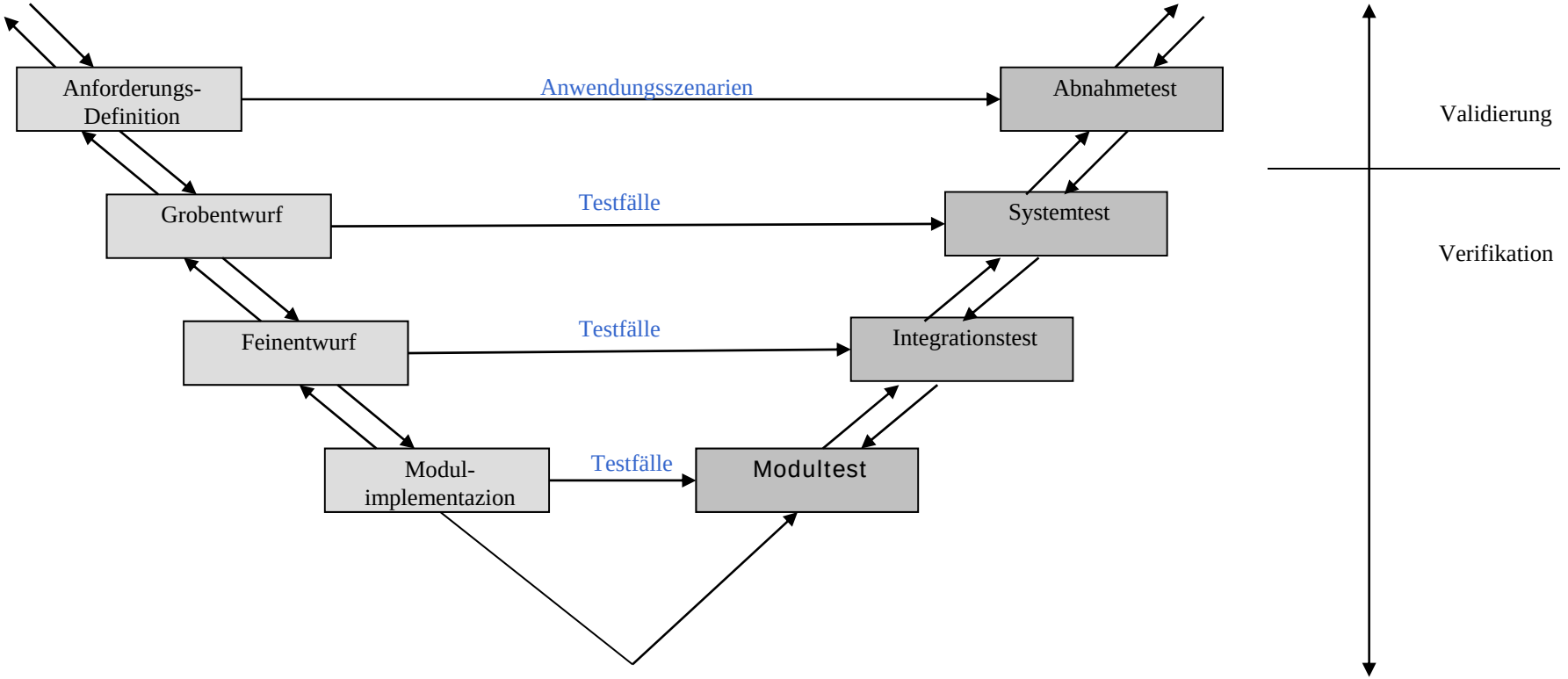
V-Modell nach Boehm

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Erweiterung des Wasserfall-Modells durch Integration einer expliziten Qualitätssicherung
- Verifikation und Validation der Teilprodukte sind Bestandteile des V-Modells

  - **Verifikation** - Überprüfung der Übereinstimmung zwischen einem Software-Produkt und seiner Spezifikation - „Wird ein korrektes Produkt entwickelt?“

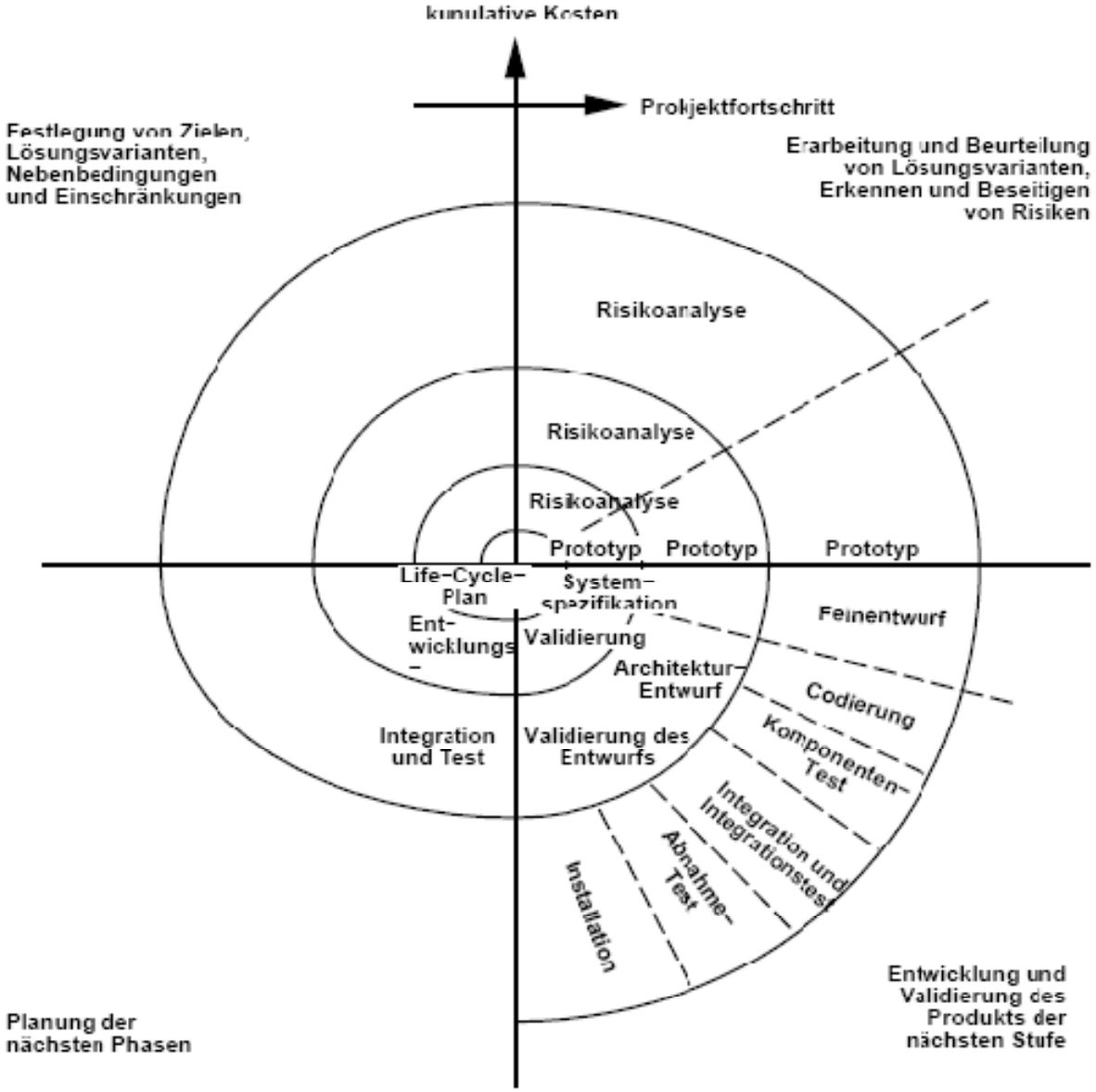
  - **Validation** - Eignung bzw. der Wert eines Produktes bezogen auf seinen Einsatzzweck - „Wird das richtige Produkt entwickelt?“



Quelle: Balzert, H.; Lehrbuch der Software-Technik

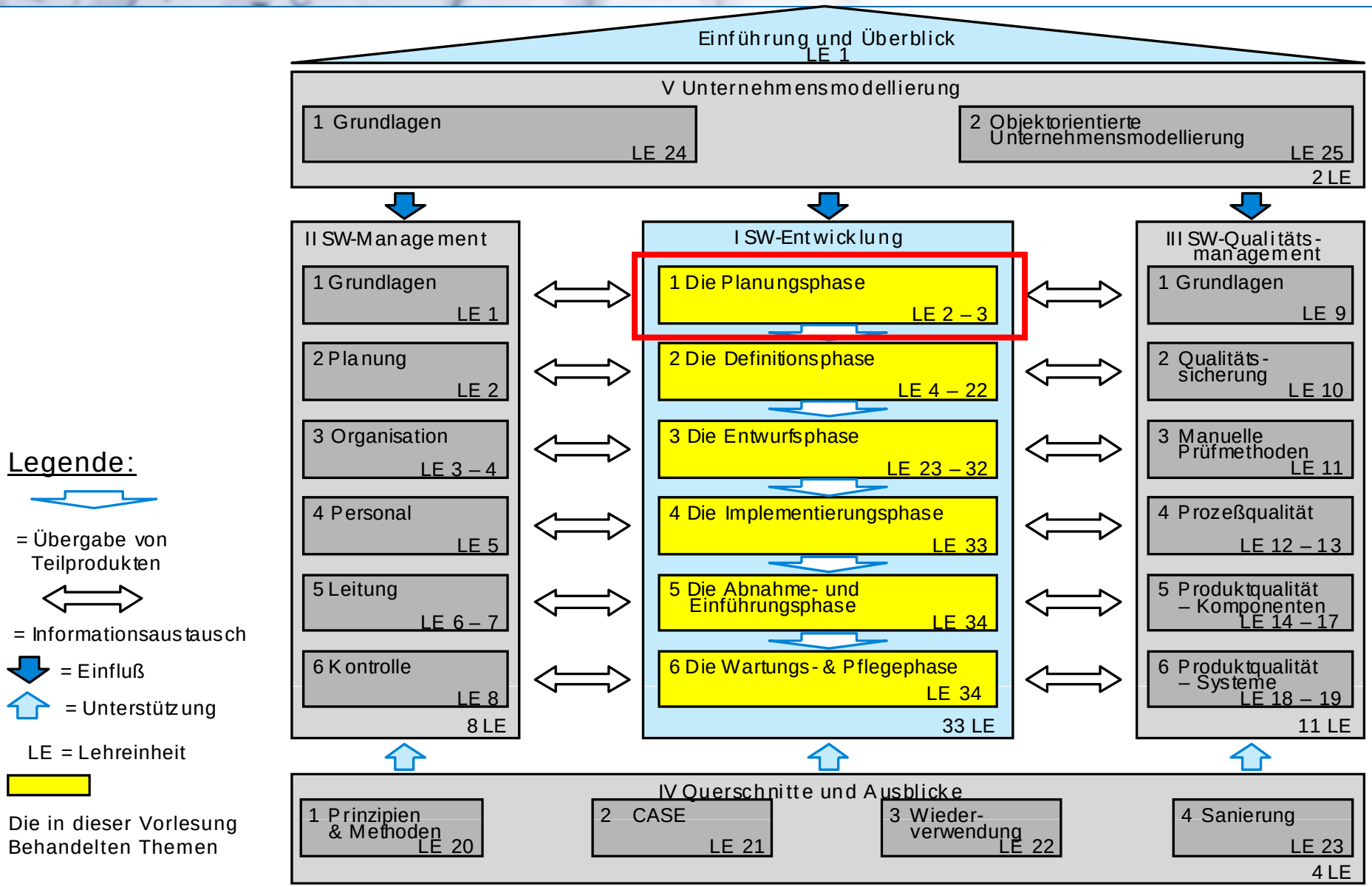
Spiralmodell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Planungsphase

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Voruntersuchung bzw. Durchführbarkeitsuntersuchung

- Zeigen der fachlichen, ökonomischen und personelle Durchführbarkeit
- Am Ende der Planungsphase steht die Entscheidung über die weitere Vorgehensweise

Aktivität Planen des Produktes beinhaltet :

- Auswählen des Produktes
- Voruntersuchung des Produkts
- Durchführbarkeitsuntersuchung
- Ergebnisse dieser Tätigkeiten:
  - Lastenheft:

    Enthält die Zusammenfassung aller fachlichen Basisanforderungen, die das zu entwickelnde Software-Produkt aus Sicht des Auftraggebers erfüllen muss.
  - Glossar:

    Definiert und erläutert Begriffe, um eine einheitliche Terminologie sicherzustellen.

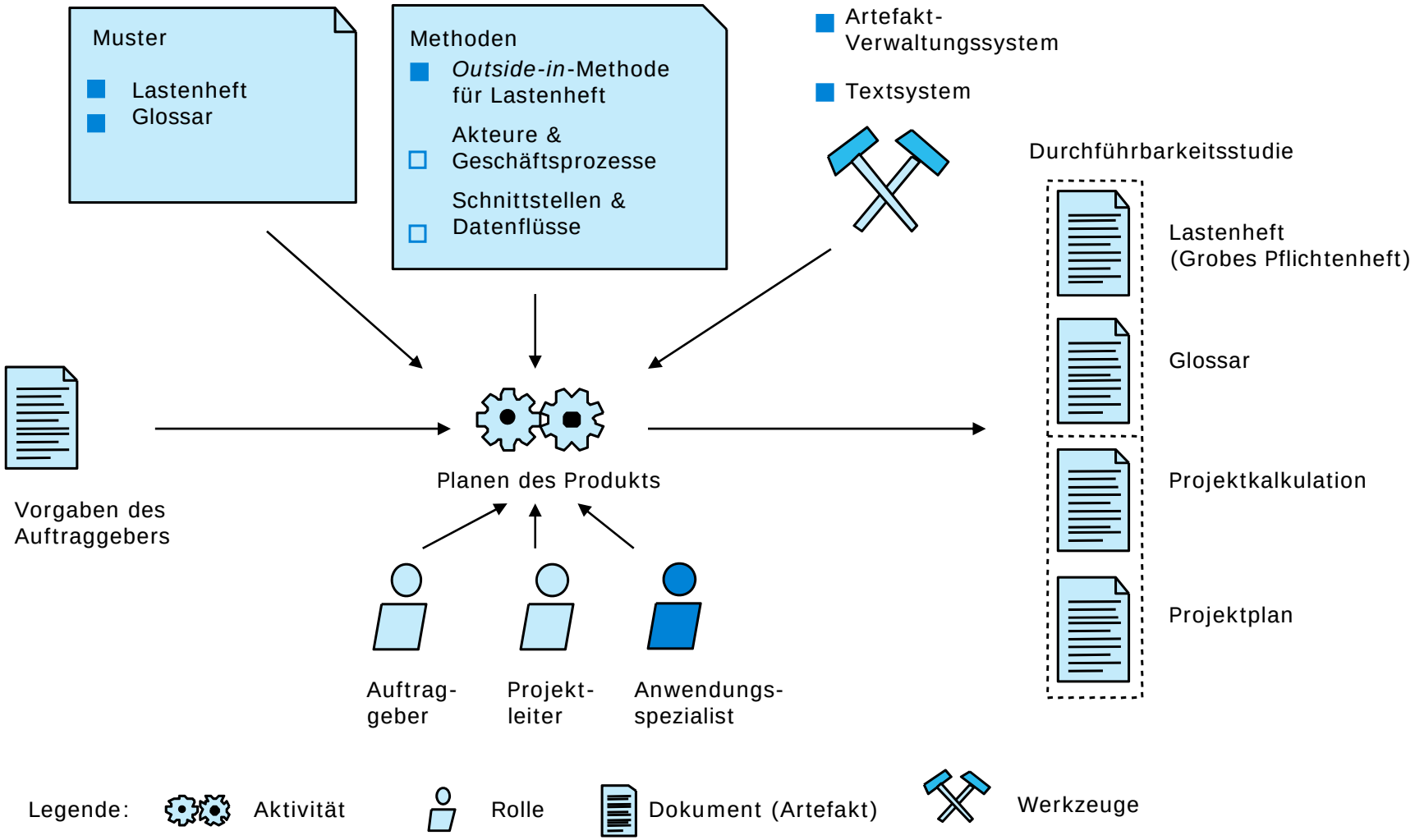
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Gliederung und Übersicht Lastenheft

- **Zielbestimmung**: Hier wird beschrieben, welche Ziele die durch den Einsatz des Produktes erreicht werden sollen.
- **Produkteinsatz**: Es wird festgelegt für welche Anwendungsbereiche und für welche Zielgruppen das Produkt vorgesehen ist.
- **Produktübersicht**: Gibt einen (meist grafischen) Überblick über die Produktumgebung, z. B. durch ein Umweltdiagramm.
- **Produktfunktionen**: Hauptfunktionen des Produkts aus Auftraggeber-sicht sind auf oberster Abstraktionsebene zu beschreiben. Funktionalität kann mit Hilfe von Akteuren und Geschäftsprozessen oder Schnittstellen und Datenflüssen systematisch ermittelt werden.
- **Produktdaten**: Die langfristig zu speichernden Hauptdaten und deren voraussichtlicher Umfang (Mengengerüst) sind aus Benutzersicht auszuführen (/LDnn/).
- **Produktleistungen**: Werden Leistungsanforderungen bzgl. Zeit oder Genauigkeit gestellt, dann werden sie hier aufgeführt und mit /LLnn/ markiert.
- **Qualitätsanforderungen**: Die wichtigsten Qualitätsanforderungen und die jeweils geforderte Qualitätsstufe sind hier auszuführen. (Benutzbarkeit,Effizienz,)
- **Ergänzungen**: Hier werden Ergänzungen oder spezielle Anforderungen beschrieben.

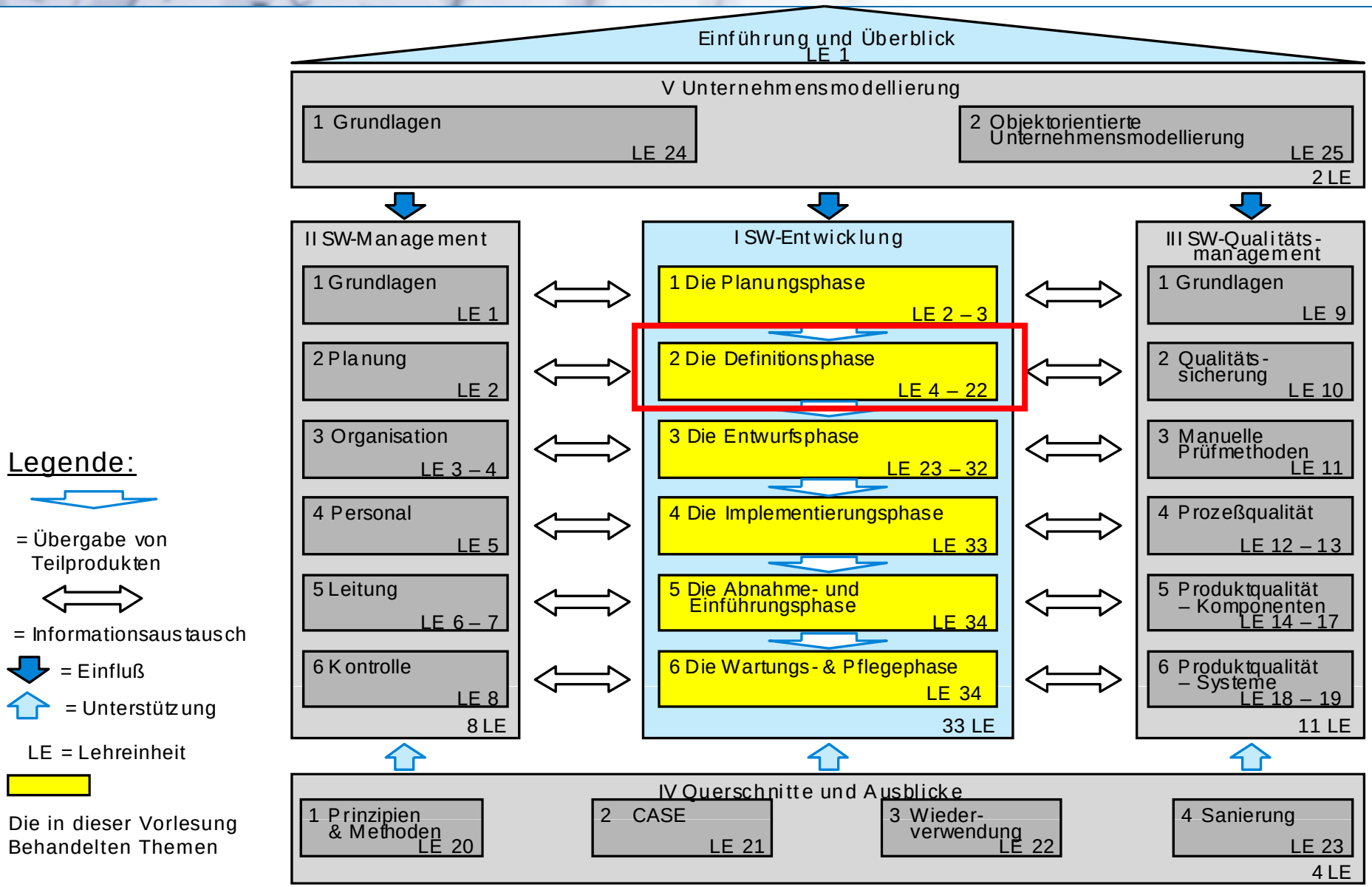
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Übersicht Planungsphase



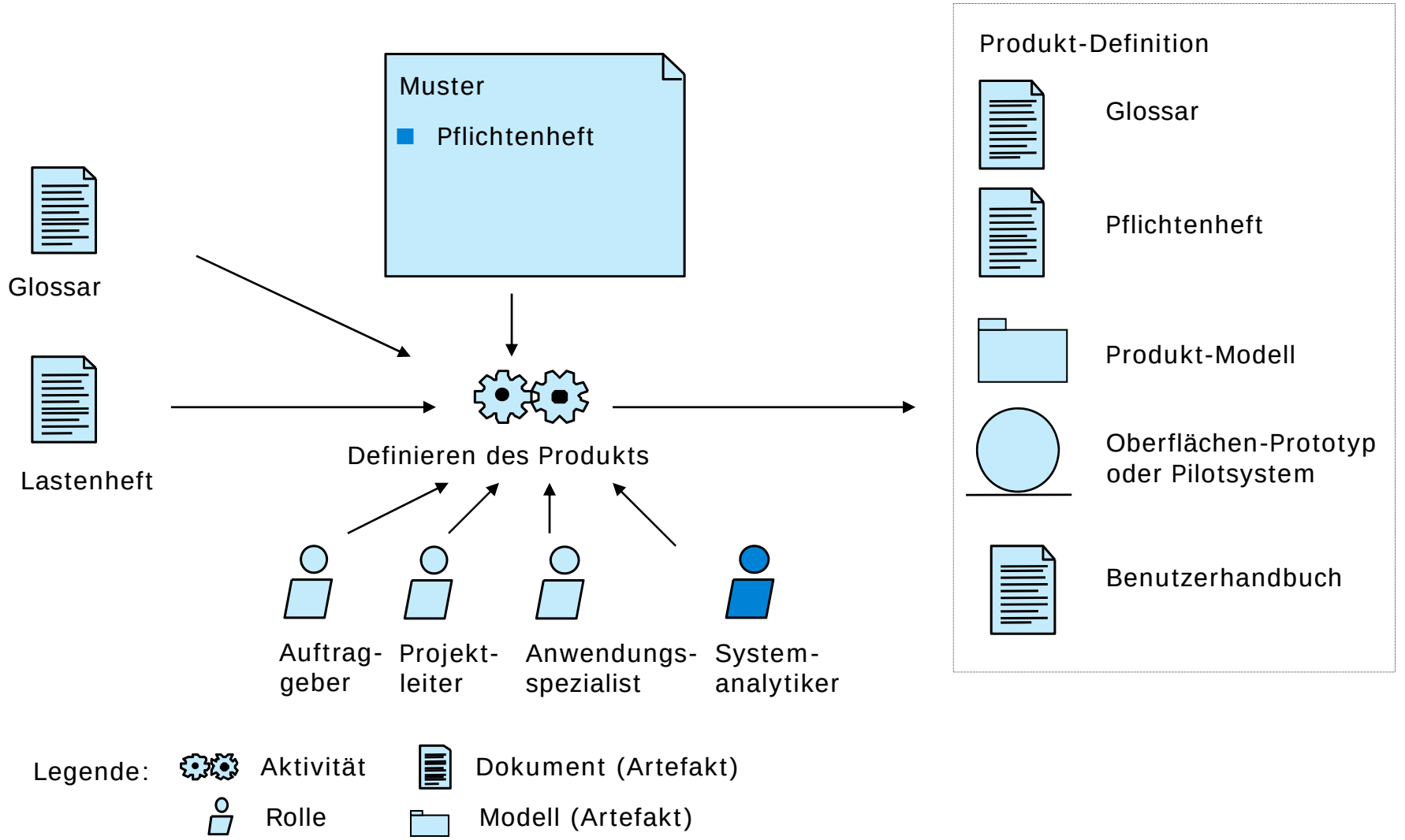
Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



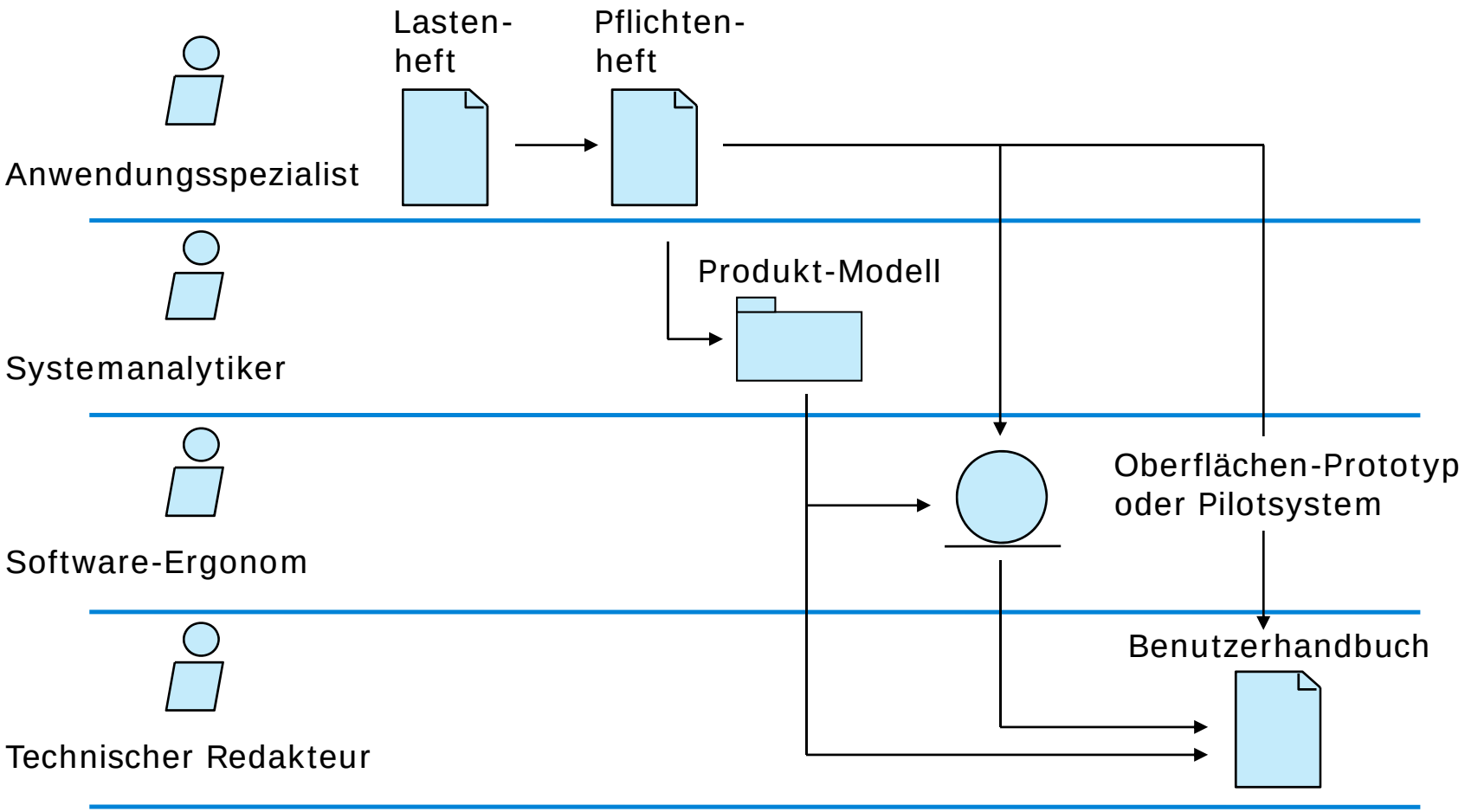
Übersicht Definitionsphase

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Aktivitäten der Definitionsphase



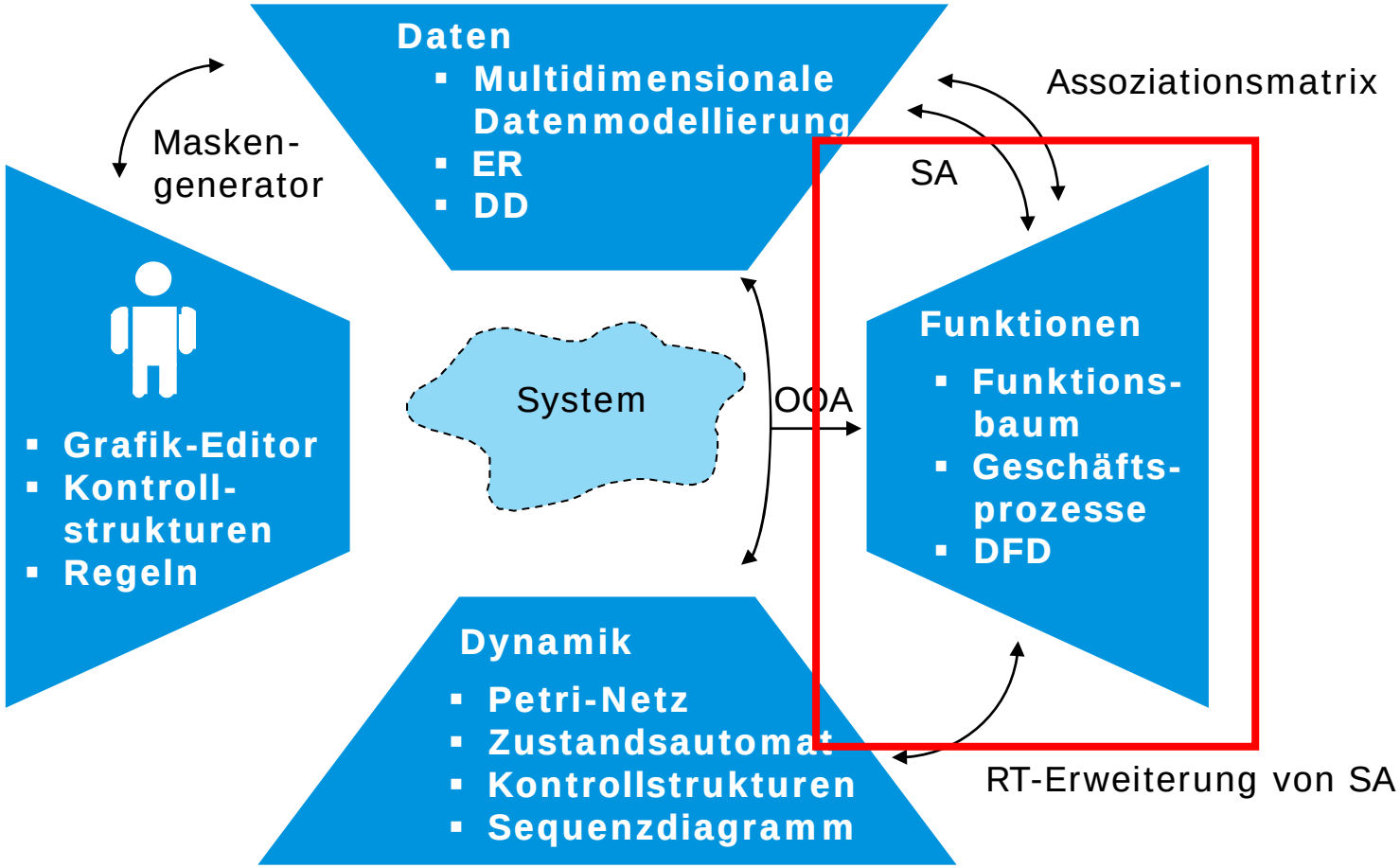
Gliederung Pflichtenheft

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

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
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

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Sichten und ihre Methoden



Legende:

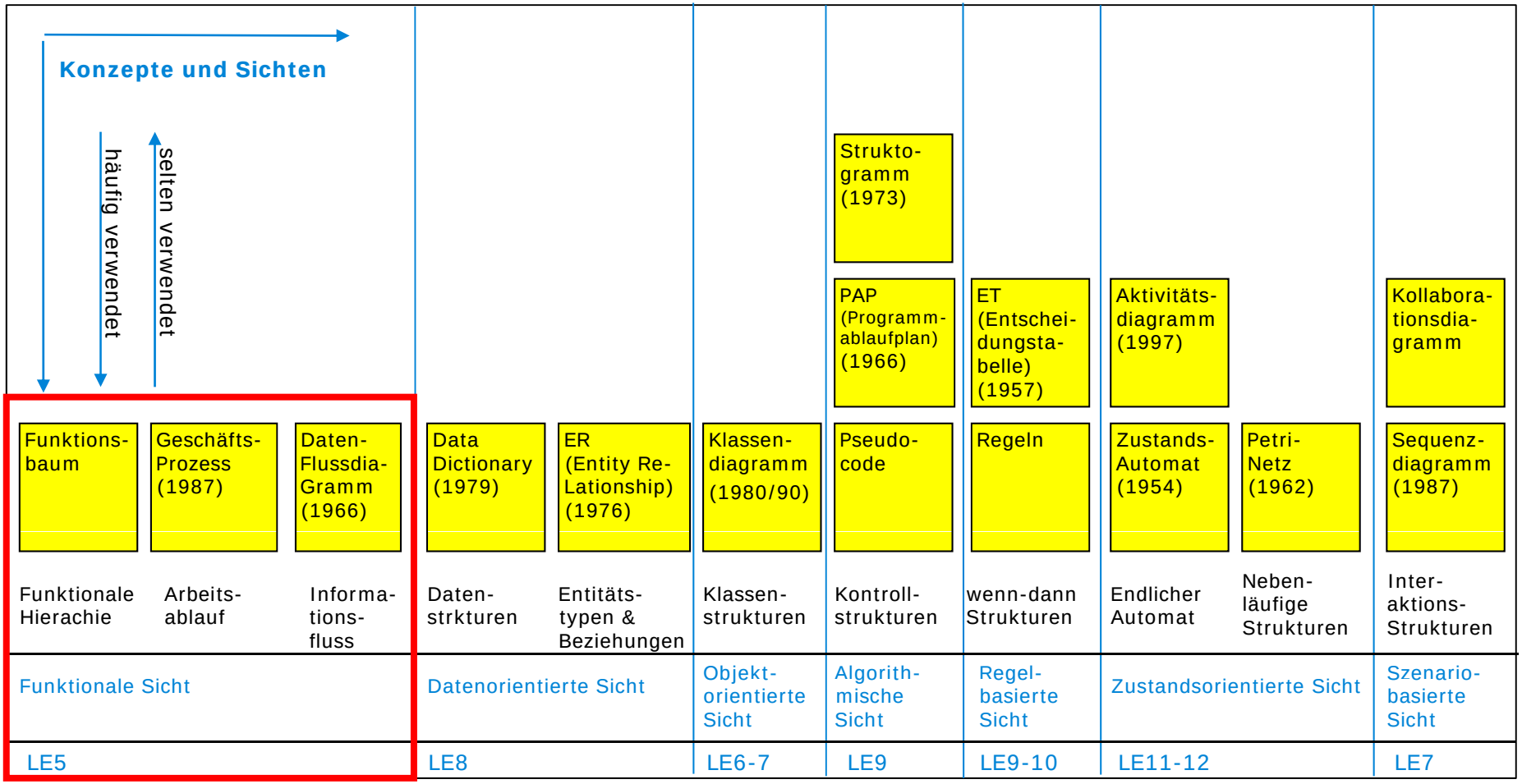
= Benutzungs-oberfläche

ER = Entity Relationship
DD = Data Dictionary
DFD = Datenflussdiagramm

RT = Realtime Analysis
OOA = Object Oriented Analysis
SA = Structured Analysis

Funktionale Sicht

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



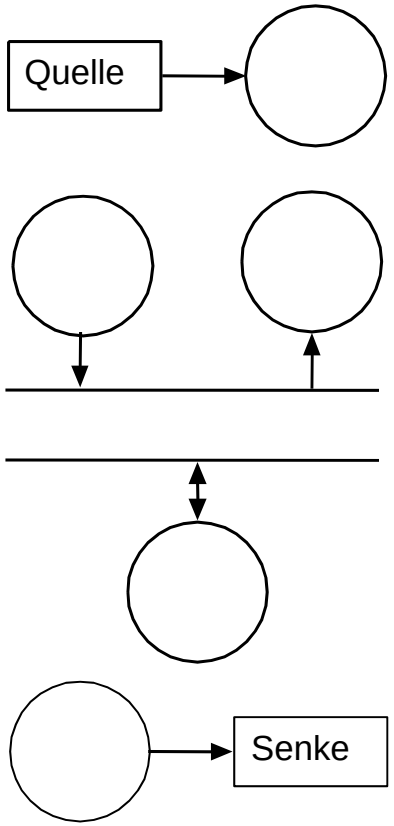
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Geschäftsprozesse

- Geschäftsprozesse im Großen
  - Ablauforganisation eines Unternehmens
  - Unternehmensprozess (business process)
    - Besteht aus einer Anzahl von unternehmensinternen Aktivitäten, die durchgeführt werden, um die Wünsche eines Kunden zu befriedigen
- Geschäftsprozesse im Kleinen
  - Funktionalität eines Produkts
  - Definiert einen Arbeitsablauf, der mit Hilfe von Software durchgeführt wird, aber manuelle und organisatorische Anteile besitzen kann
  - use case
    - Teil eines Geschäftsprozesses, der die Benutzerkommunikation mit dem Software-System beschreibt
  - Geschäftsprozess (Arbeitsablauf)
    - besteht aus mehreren zusammenhängenden Aufgaben, die von einem Akteur durchgeführt werden, um ein Ziel zu erreichen bzw. ein gewünschtes Ergebnis zu erstellen.
- Akteure
  - Rollen von Menschen oder Systeme, insbesondere Computersysteme, die als externe Beteiligte mit einem Unternehmen (Kunden) oder einem Software-Produkt (Benutzer) kommunizieren und Daten austauschen; stets außerhalb des Systems

Datenflussdiagramm

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Ein DFD (data flow diagram) beschreibt die Wege von Daten bzw. Informationen zwischen Funktionen, Speichern und Schnittstellen und die Transformation der Daten bzw. Informationen durch Funktionen. DFDs stellen also die „Daten-Pipeline“ dar.

Grundidee eines DFD besteht darin, dass man sich vorstellt, das zu entwickelnde System läuft bereits. Man macht sich keine Gedanken darüber, wie das System initialisiert und terminiert wird. Man konzentriert sich darauf, welche Informationen von wo nach wo durch das System fließen.

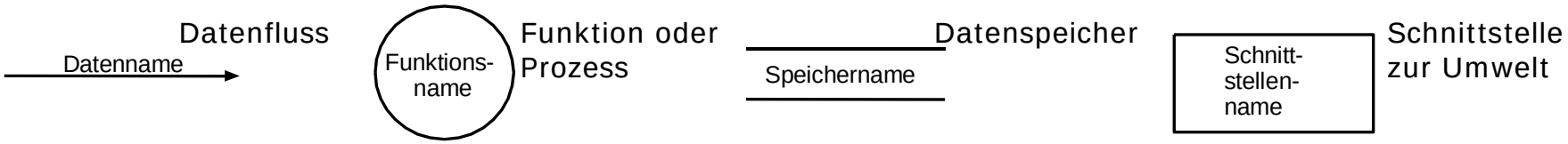
System hat **Schnittstellen** mit seiner Umwelt. Schnittstellen können Datenquellen und -senken sein

Umwelt besteht für das System aus Informations-quellen und Informationssenken.

Speicher sind Hilfsmittel zur Ablage von Informationen. Informationen können hineinfließen oder herausgelesen werden (siehe Pfeilrichtungen).

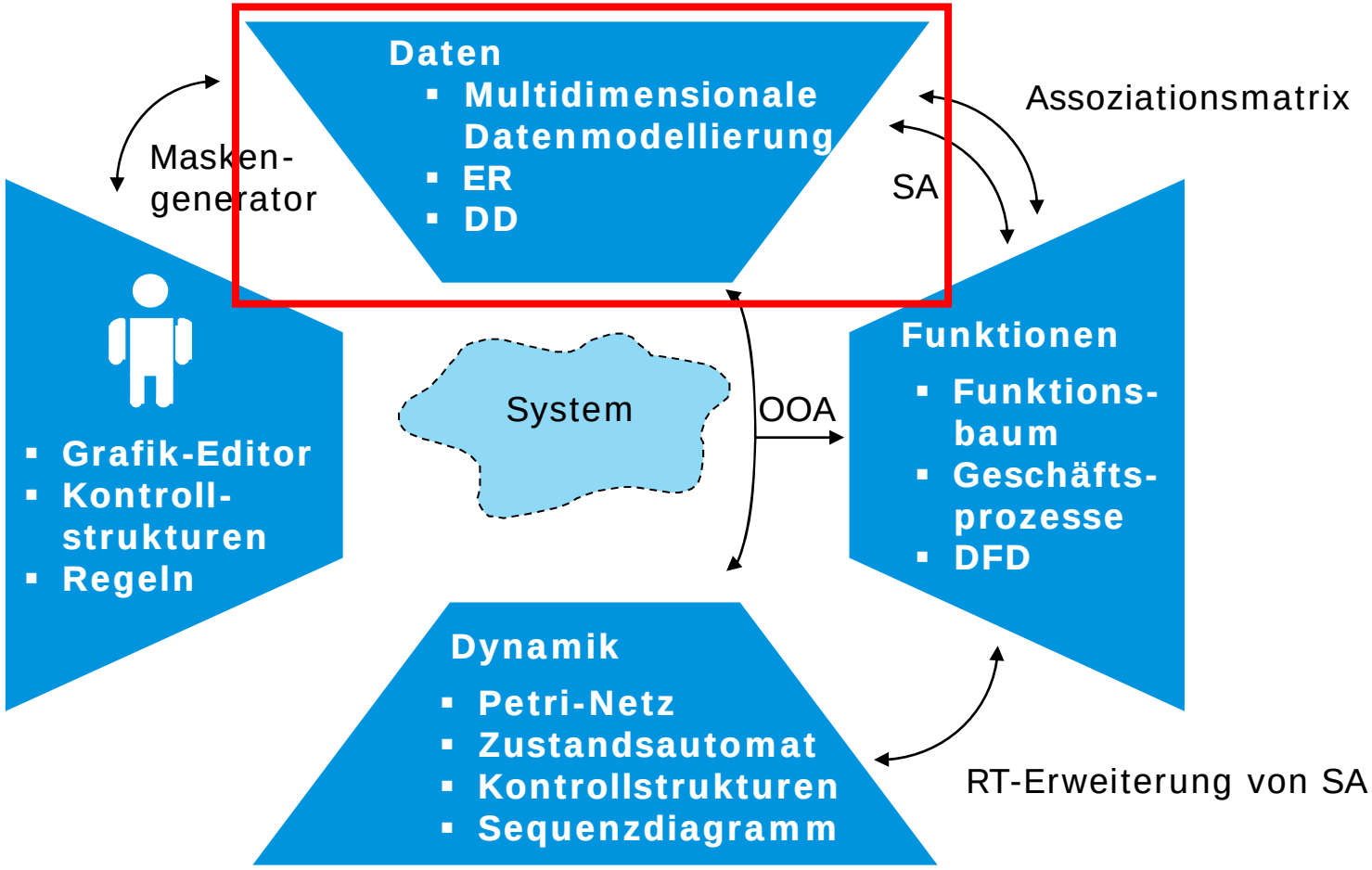
Funktionen transformieren den ankommenden Datenfluss in den abgehenden.

Legende:



Sichten und ihre Methoden

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende:

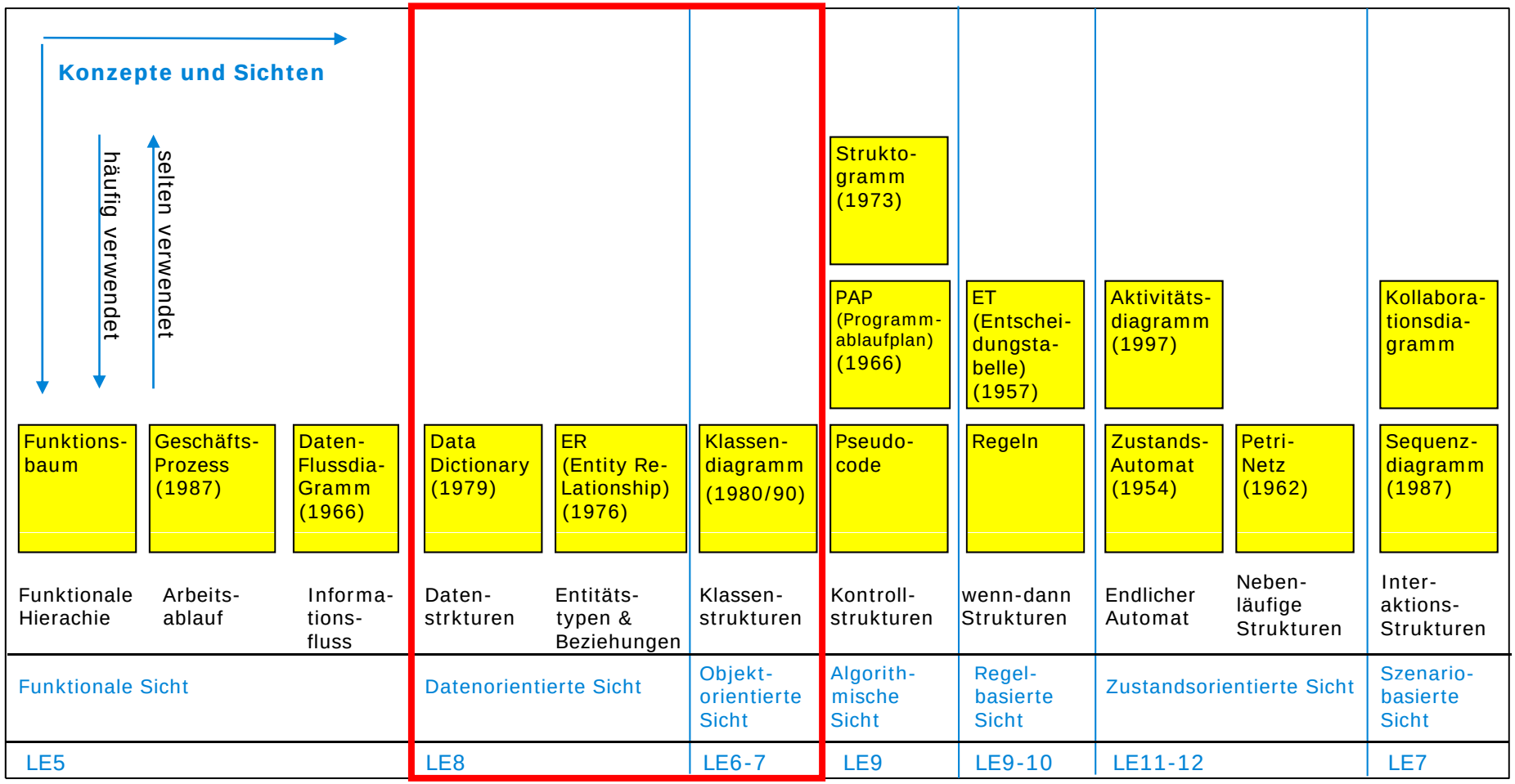
= Benutzungs-oberfläche

ER = Entity Relationship
DD = Data Dictionary
DFD = Datenflussdiagramm

RT = Realtime Analysis
OOA = Object Oriented Analysis
SA = Structured Analysis

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Datenorientierte- und objektorientierte Sicht



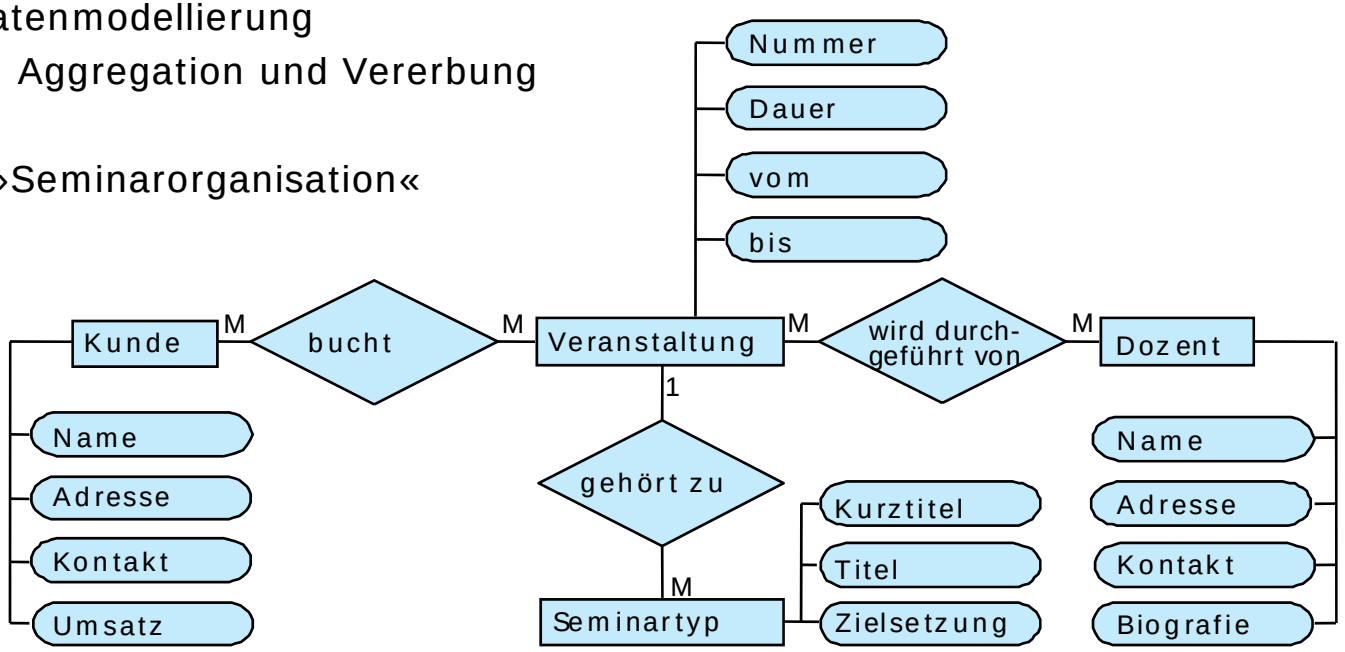
Entity-Relationship-Modell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Entity-Relationship-Modell (ER-Modell)
  - 1976 von P. Chen zur Datenmodellierung entwickelt
- Ziel
  - Beschreibung der permanent gespeicherten Daten und ihre Beziehungen untereinander.
  - Die Analyse der Information erfolgt aus fachlogischer Sicht.

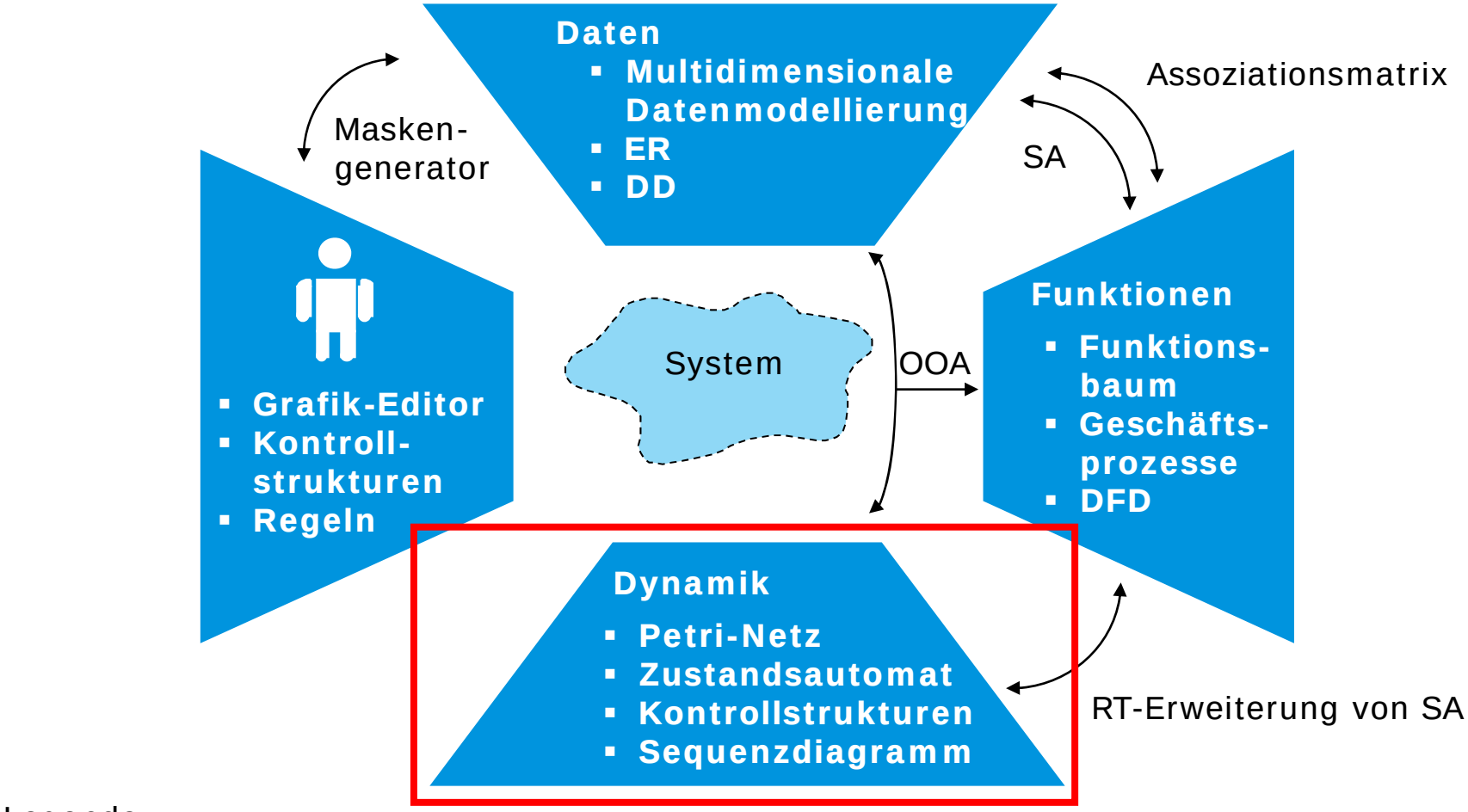
- Ergebnis
  - Konzeptionelles Modell, das gegen Veränderungen der Funktionalität weitgehend stabil ist.
  - Semantische Datenmodellierung
  - Erweiterung um Aggregation und Vererbung

- Beispiel: Fallstudie »Seminarorganisation« (Chen-Notation)



Sichten und ihre Methoden

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende:

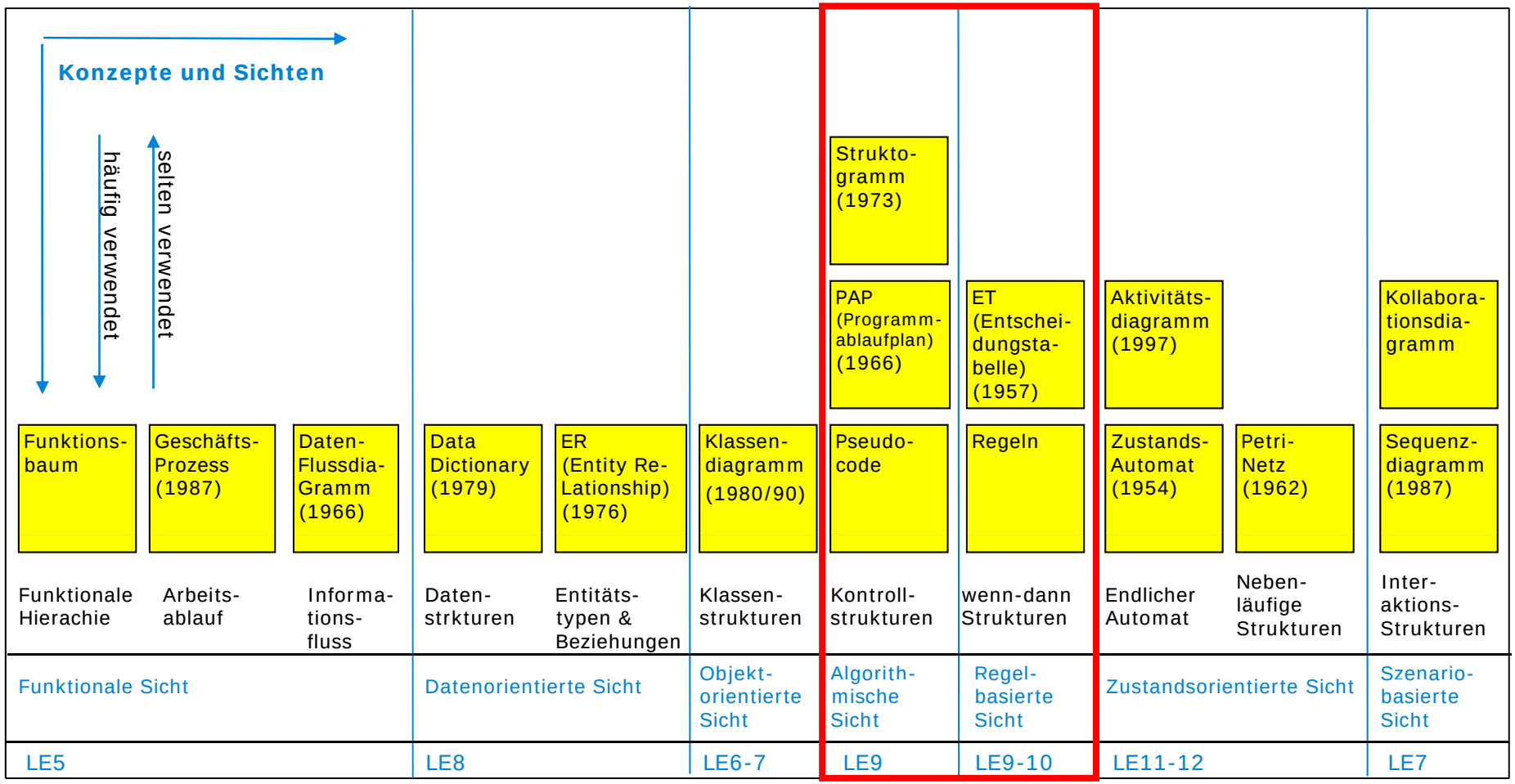
= Benutzungs-oberfläche

ER = Entity Relationship
DD = Data Dictionary
DFD = Datenflussdiagramm

RT = Realtime Analysis
OOA = Object Oriented Analysis
SA = Structured Analysis

Algorithmische und regelbasierte Sicht

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Kontrollstrukturen / Entscheidungstabelle

- Kontrollstrukturen legen innerhalb eines Algorithmus fest, in welcher Reihenfolge, ob und wie oft Anweisungen ausgeführt werden sollen. Die strukturierte Programmierung erlaubt nur lineare Kontrollstrukturen. Es lassen sich vier verschiedene Typen unterscheiden: die Sequenz, die Auswahl, die Wiederholung und der Aufruf. Alle Typen lassen sich beliebig miteinander kombinieren und ineinander schachteln.
- Eine textuelle Darstellungsform (Pseudo-Code) und zwei grafische (Struktogramme und PAP).
- Entscheidungstabellen erlauben eine tabellarische oder grafische Darstellung (Entscheidungsbäume) von durchzuführenden Aktionen in Abhängigkeit von Bedingungen.
- Tritt zu einer Zeit genau eine Bedingungskonstellation auf, dann handelt es sich um eine Eintreffer-Tabelle.
- Begrenzte Entscheidungstabelle liegt vor, wenn als Bedingungsanzeiger nur J,N und als Aktionszeiger nur X verwendet werden.
- Erfordert eine Problembeschreibung eine Kombination von Entscheidungs-tabellen (möglich sind Sequenz, Verzweigung, Schleife und Schachtelung), dann geschieht dies durch einen Entscheidungstabellen-Verbund.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Regeln

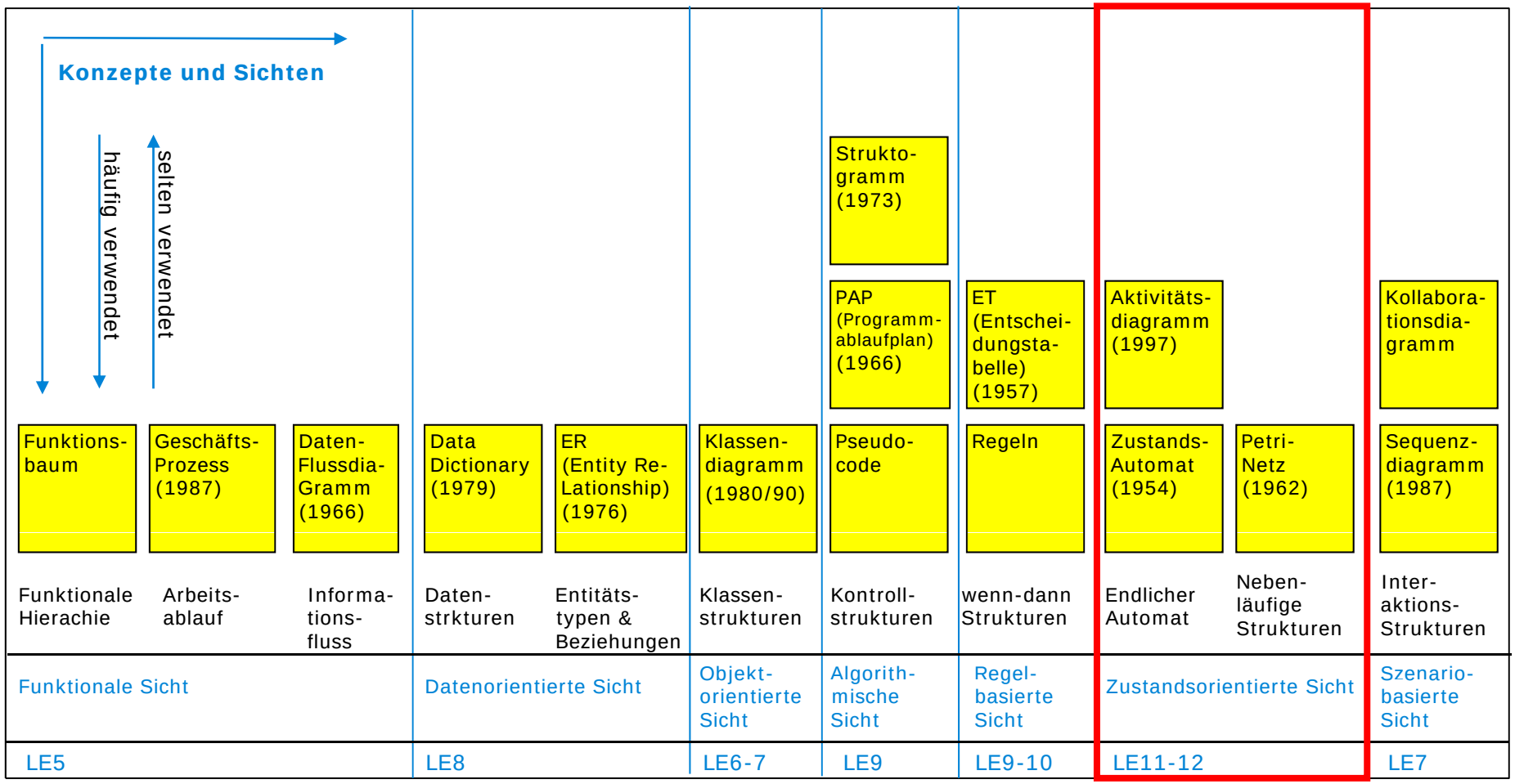
- Eine Regel besteht aus einer Vorbedingung und einer Aktion
  - wenn Vorbedingung dann Aktion
  - Vorbedingung beschreibt eine Situation, in der die Aktion ausgeführt werden soll.

- Zwei Typen von Aktionen
  - Implikationen oder Deduktionen, mit denen der Wahrheitsgehalt einer Feststellung hergeleitet wird.
  - Handlungen, mit denen ein Zustand verändert wird.

- Regeln vergleichbar mit denen in Entscheidungstabellen
  - Regeln stehen hier jeweils für sich,
  - ET sind in einen festen Kontext eingebettet.
  - Bei den Aktionen handelt es sich um Handlungen.
  - Lassen sich Anforderungen an ein neues System noch nicht als Entscheidungstabellen oder Bäume strukturieren, dann als Menge einzelner Regeln formulieren.

Zustandsorientierte und szenariobasierte Sicht

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

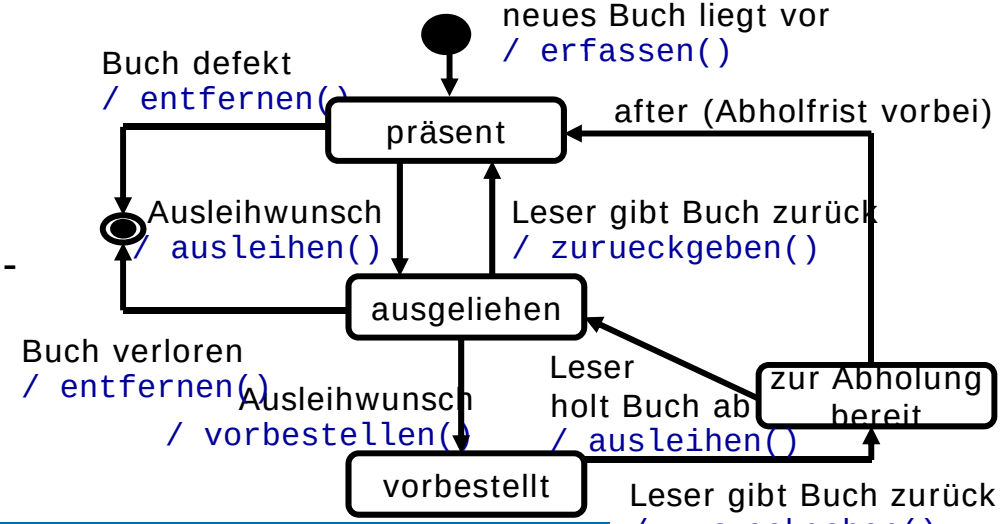
Zustandsautomaten

Viele Systeme und Geräte zeigen ein Verhalten, das von der bis dahin durchlaufenen Historie abhängt. Das aktuelle Verhalten wird durch den internen **Zustand** bestimmt, der durch vorausgegangene Eingaben oder **Ereignisse** erreicht worden ist. Zur Modellierung solcher Systeme eignen sich **Zustandsautomaten (finite state machine)**, auch **endliche Automaten (finite automation, sequential machine)** genannt.

Gegenüber einem allgemeinen Automaten besitzen endlich Automaten nur eine endliche Zahl von Zuständen. Bei deterministischen endlichen Automaten gibt es zu einer Eingabe von einem Zustand aus höchstens einen **Zustandsübergang (transition)** in einen anderen Zustand, während bei nichtdeterministischen endlichen Automaten mehrere Übergänge für dieselbe Eingabe möglich sind.

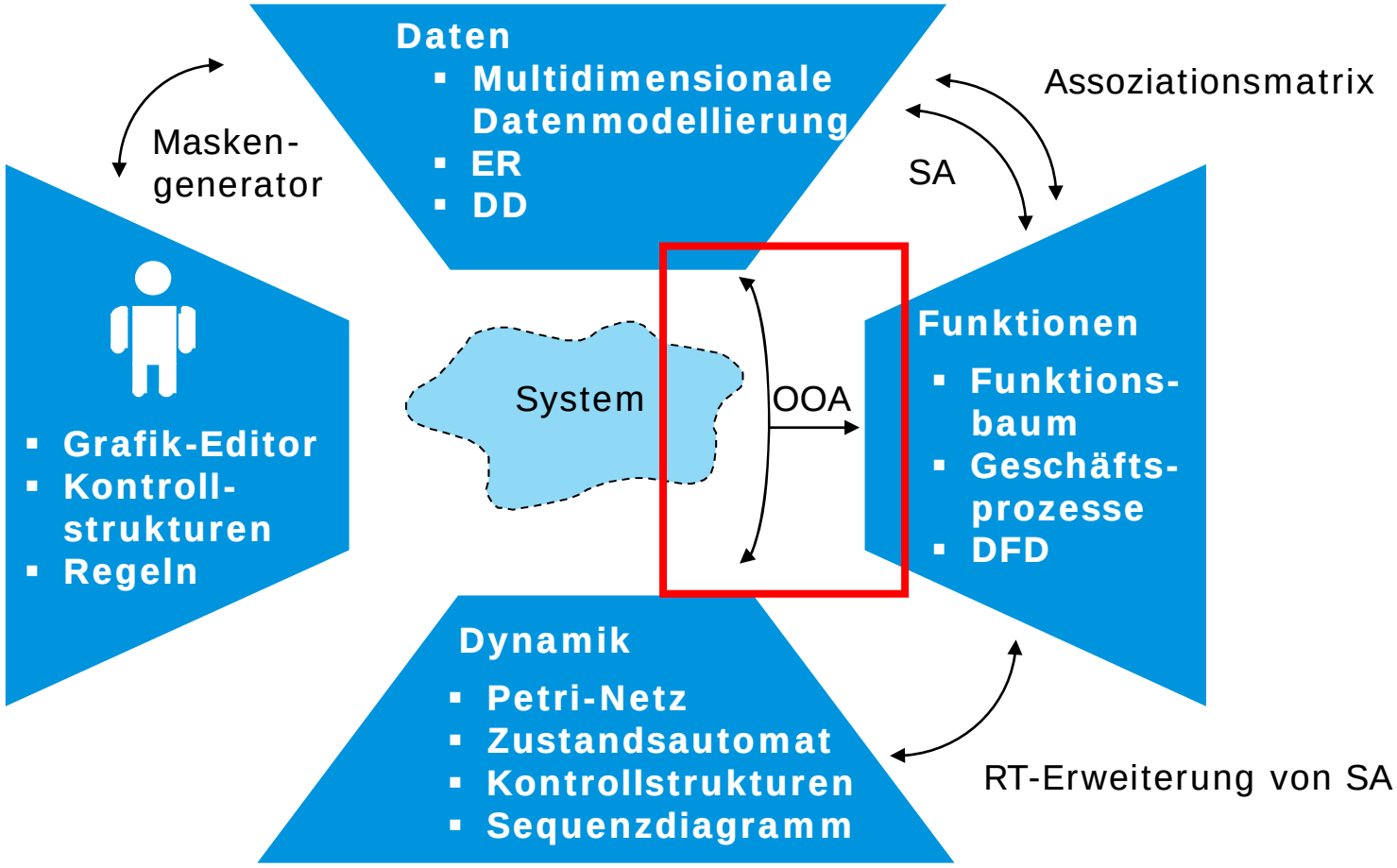
Zustandsautomaten können als **Zustandsdiagramm**,**Zustands-tabelle** oder **Zustandsmatrix** dargestellt werden.

Ziel beim Aufstellen eines Zustands-automaten ist es, mit möglichst wenig Zuständen auszukommen.



Sichten und ihre Methoden

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



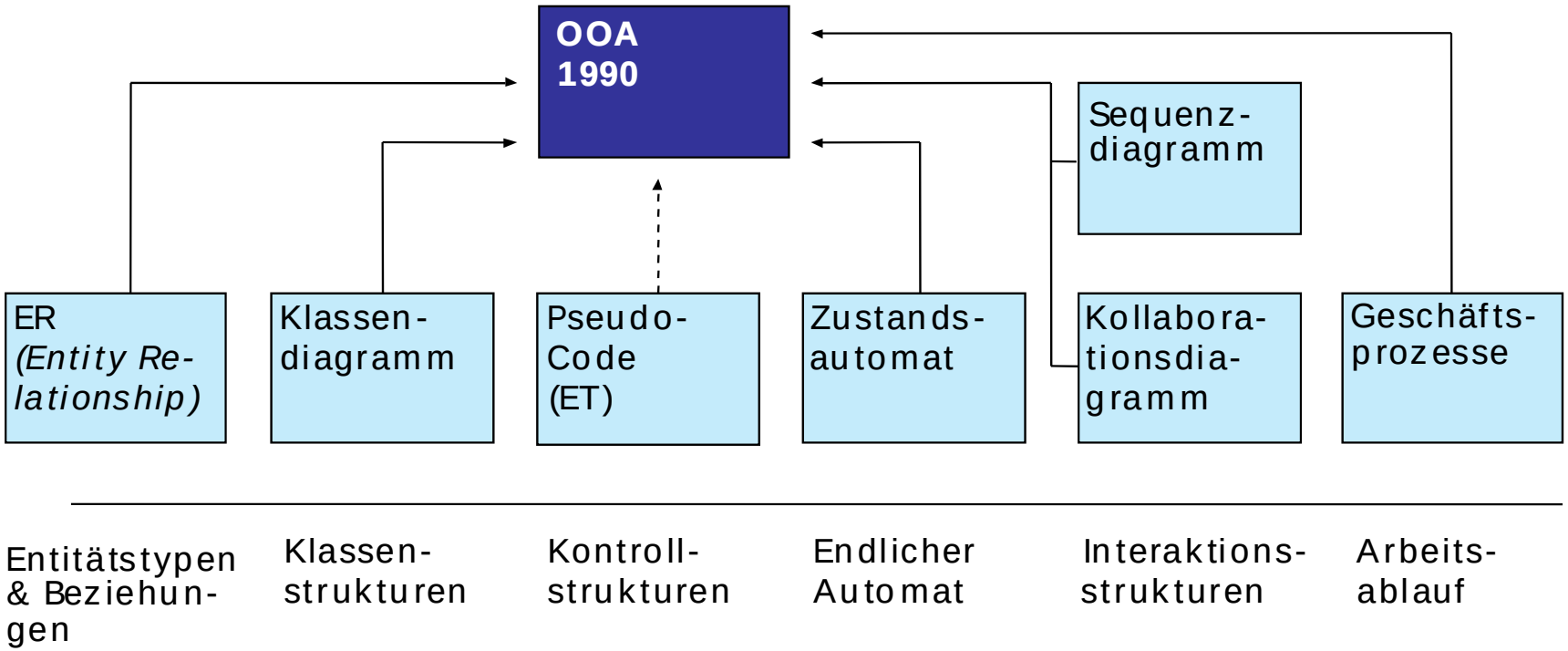
Legende:

= Benutzungs-oberfläche

ER = Entity Relationship
DD = Data Dictionary
DFD = Datenflussdiagramm

RT = Realtime Analysis
OOA = Object Oriented Analysis
SA = Structured Analysis

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende: A ⟶ B: A ist in B enthalten
A ⇢ B: A ist implizit in B enthalten

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Übersicht OOA

- Die OOA verwendet die Basiskonzepte der Objektorientierung - Objekt, Klasse, Attribute, Operationen – ergänzt um die statischen Konzepte Assoziation, Vererbung, Paket und die dynamischen Konzepte Botschaft, Geschäftsprozess, Zustandsautomat und Szenario um die fachliche Problemlösung zu modellieren. Dabei entstehen ein statisches und ein dynamisches Modell, die sich gegenseitig beeinflussen und ausbalanciert sein sollen.

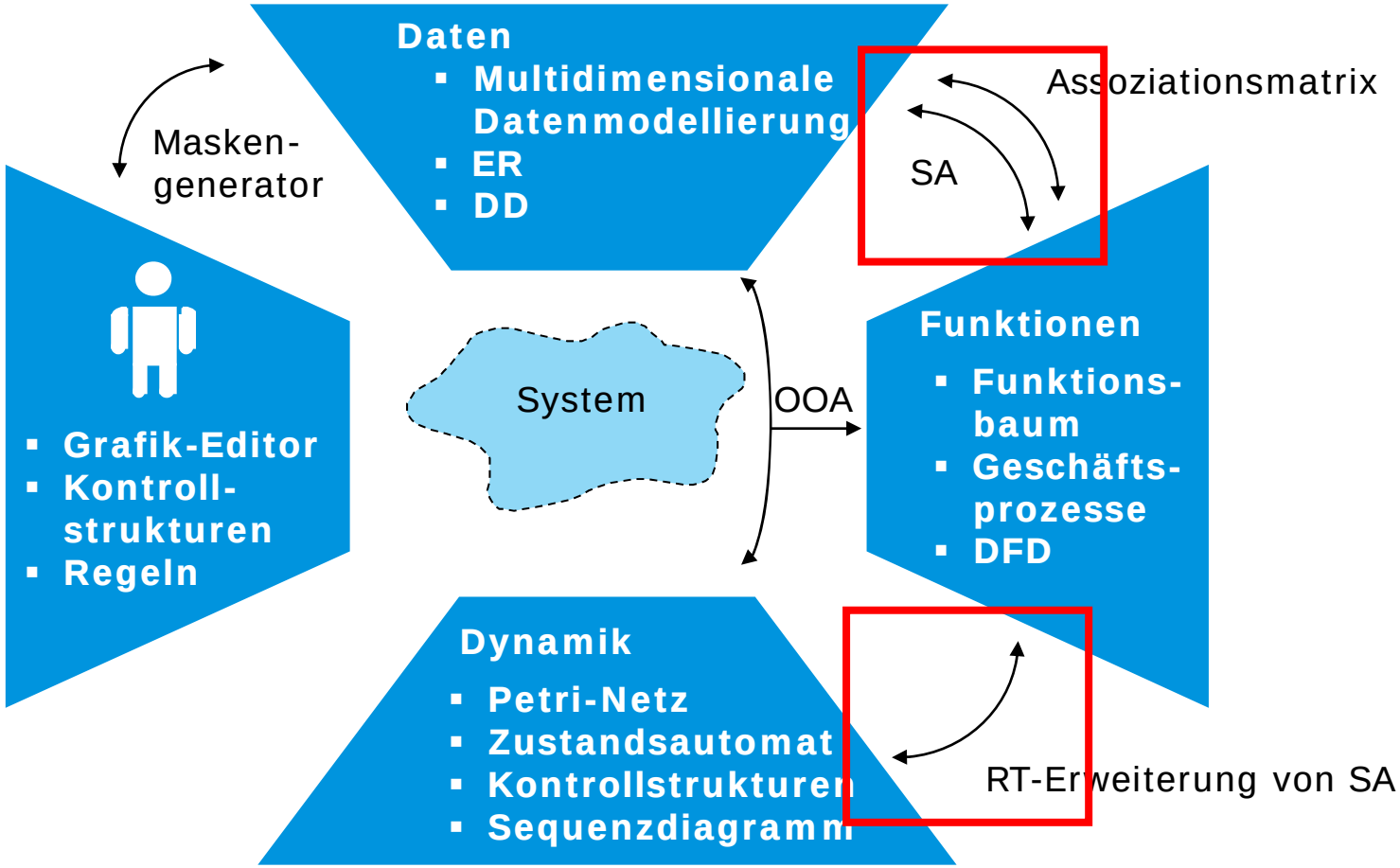
- Muster (patterns) sind bewährte generische Lösungen für immer wiederkehrende Probleme, die in bestimmten Situationen auftreten. OOA-Muster werden für die Analyse und Konstruktion von OOA-Modellen benutzt. Wichtige OOA-Muster sind: Liste, Exemplartyp, Baugruppe, Stückliste, Koordinator, Rollen, wechselnde Rollen, Historie, Gruppe und Gruppenhistorie.

- Das Erstellen eines OOA-Modells auf der Grundlage schriftlicher Informationen, z. B. eines Pflichtenheftes, oder mündlicher Informationen, z. B. durch Interviews, gehört zu den schwierigsten Aufgaben der Software-Technik.

- Es handelt sich um einen iterativen Vorgang. Beschrieben wurde ein Makroprozess, der die Gleichgewichtigkeit von statischem und dynamischem Modell berücksichtigt. Für jedes objektorientierte Konzept stehen einheitlich aufgebaute Checklisten zur Verfügung, die die Konstruktion und Analyse unterstützen.

Sichten und Methoden

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende:

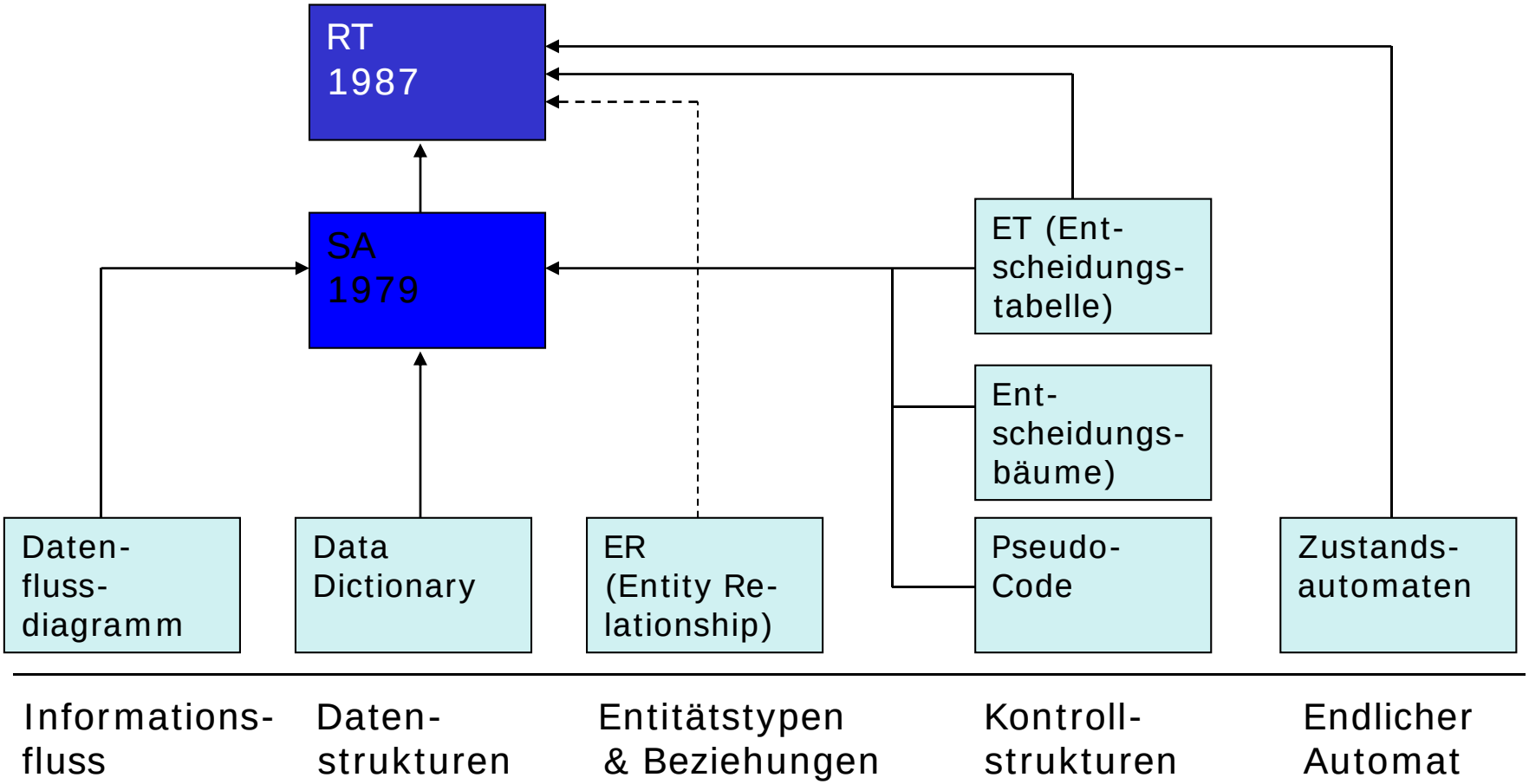
= Benutzungs-oberfläche

ER = Entity Relationship
DD = Data Dictionary
DFD = Datenflussdiagramm

RT = Realtime Analysis
OOA = Object Oriented Analysis
SA = Structured Analysis

Strukturierte Analyse / Realtime-Analyse

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende: A ⟶ B: A ist in B enthalten
A ⇢ B: A ist implizit in B enthalten

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

SA / RT

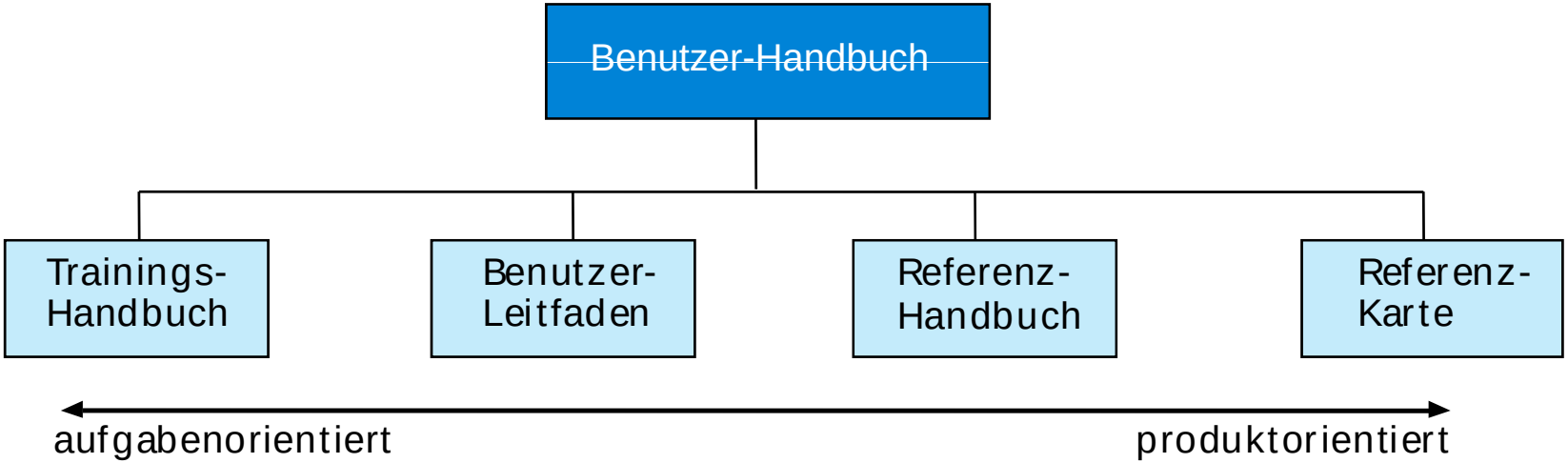
- Die Strukturierte Analyse (SA, structured analysis) besteht aus einem Hierarchiemodell, das die einzelnen DFD als Baum anordnet. Wurzel des Baumes ist das Kontextdiagramm. Blätter des Baumes sind DFD, die nicht weiter verfeinert werden können.
- Prozesse dieser DFD werden durch Mini-Spezifikationen (MiniSpecs) beschrieben. Innerhalb eines SA-Modells, d. h. von der Baumwurzel bis zu den Baumblättern, muss die Datenintegrität (balancing) sichergestellt sein, d. h., die DFD müssen zwischen Kind- und Elterndiagramm ausbalanciert sein.
- Real-Time Analysis (RT) erweitert SA um die Möglichkeit, Prozesse zu aktivieren und zu deaktivieren. Außerdem können Zeitspezifikationen beschrieben werden. Um dies zu ermöglichen, gibt es neben den Datenflüssen auch Kontrollflüsse, die Ereignisse repräsentieren. DFD werden zu Flussdiagrammen verallgemeinert, die zusätzlich Kontrollflüsse enthalten können. Die Prozesssteuerung der Prozesse,die sich auf einem Flussdiagramm befinden, erfolgt durch einen zugeordnete Kontrollflussspezifikation (Cspec). Die in der Kontrollflussspezifikation verwendeten Ein- und Ausgabensignale werden durch eine Balken-Notation (bar) im zugeordneten Flussdiagramm aufgeführt. Elementare Prozesse werden durch eine Prozess-Spezifikation (PSpec) beschrieben (anderer Name für Mini-Spezifikationen). Alle Flüsse und Speicher werden im Requirements Dictionary (RD) definiert (anderer Name für Data Dictionary).

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Handbuchtypen

- Zu jedem Software-Produkt gehört eine adäquate, vollständige und fehlerfreie Dokumentation. Die für den Endbenutzer bzw. Anwender des Software-Produktes bestimmte Dokumentation bezeichnet man als Benutzer-Handbuch. Es gibt verschiedene Handbuchtypen in Abhängigkeit davon, ob eine produktorientierte Gliederung – Referenz-Handbuch, Referenz-Karte (quick reference) – oder eine aufgabenorientierte Gliederung –Trainings-Handbuch (tutorial), Benutzer-Leitfaden (user guide) – im Vordergrund steht.

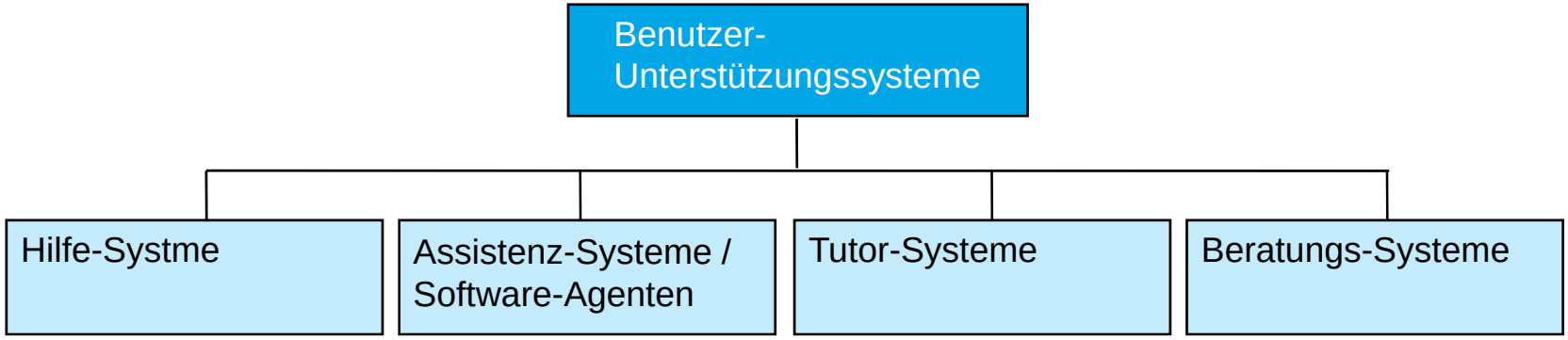
- Wichtige Gestaltungsziele für den Benutzer-Leitfaden sind die leichte Navigation, das leichte Erlernen und gute Lesbarkeit.



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Benutzer-Unterstützungssysteme

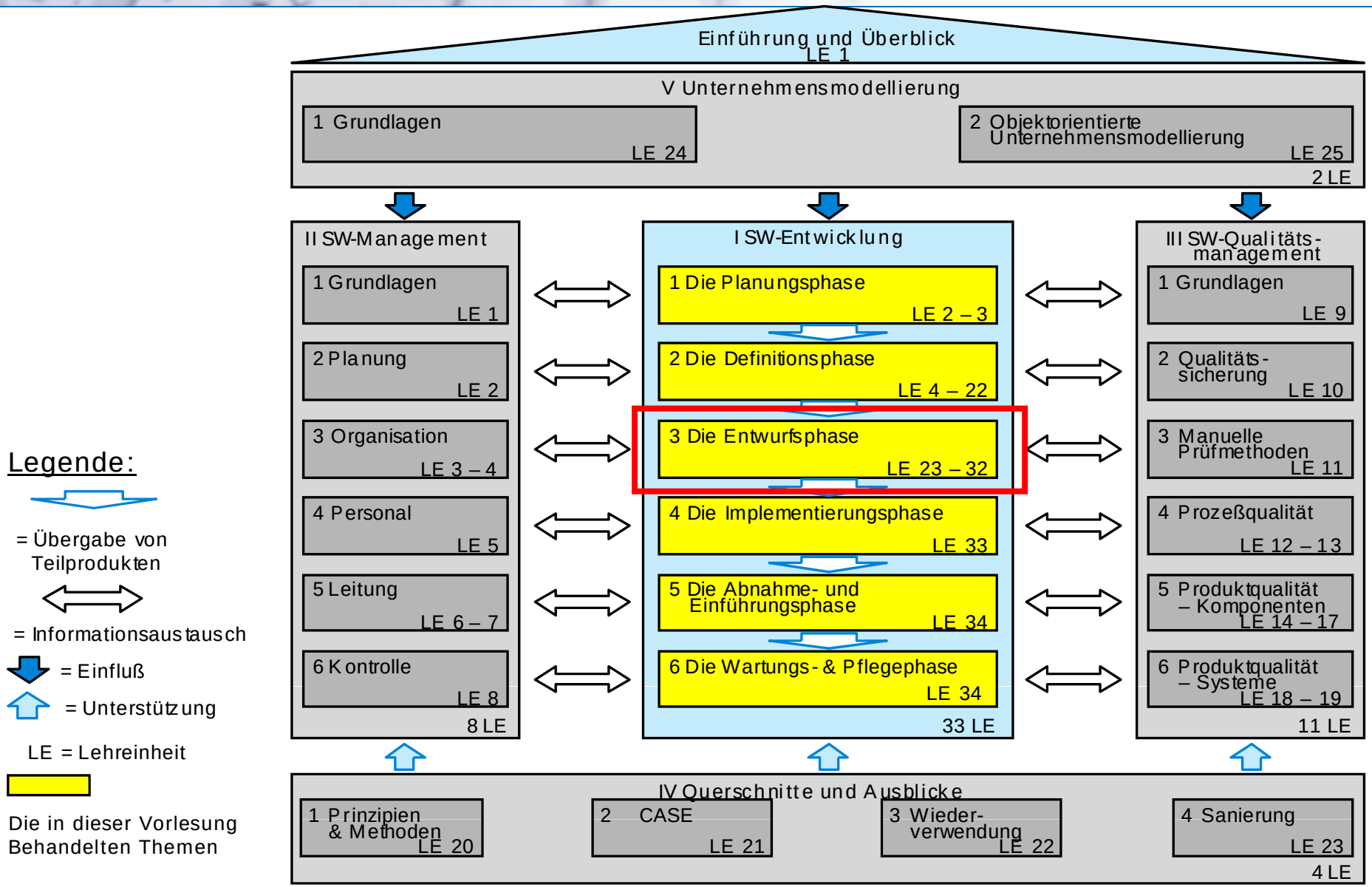
- Gute Benutzer-Unterstützungssysteme (user support system) beschleunigen die Einarbeitung, reduzieren die Schulungs- und Trainingskosten, erleichtern den Umgang mit den Software-Systemen, unterstützen den Benutzer bei der Problemlösung und übernehmen die Erledigung von „Lehrlings-Arbeiten“. Dem Benutzer bieten sie eine neue Qualität bei der Arbeit mit Software-Systemen. Hilfesysteme gehören zu den Standardleistungen eines Software-Systems. Tutorsysteme sind nur vereinzelt anzutreffen. Beratungssysteme (advisory systems) befinden sich noch im Prototypen-Stadium. Assistenz-systeme sind bereits vereinzelt im Einsatz, wobei manche in Form von Software-Agenten realisiert sind.



- Alle Systeme überlappen sich in Teilbereichen. Anzustreben sind daher integrierte Unterstützungssysteme.

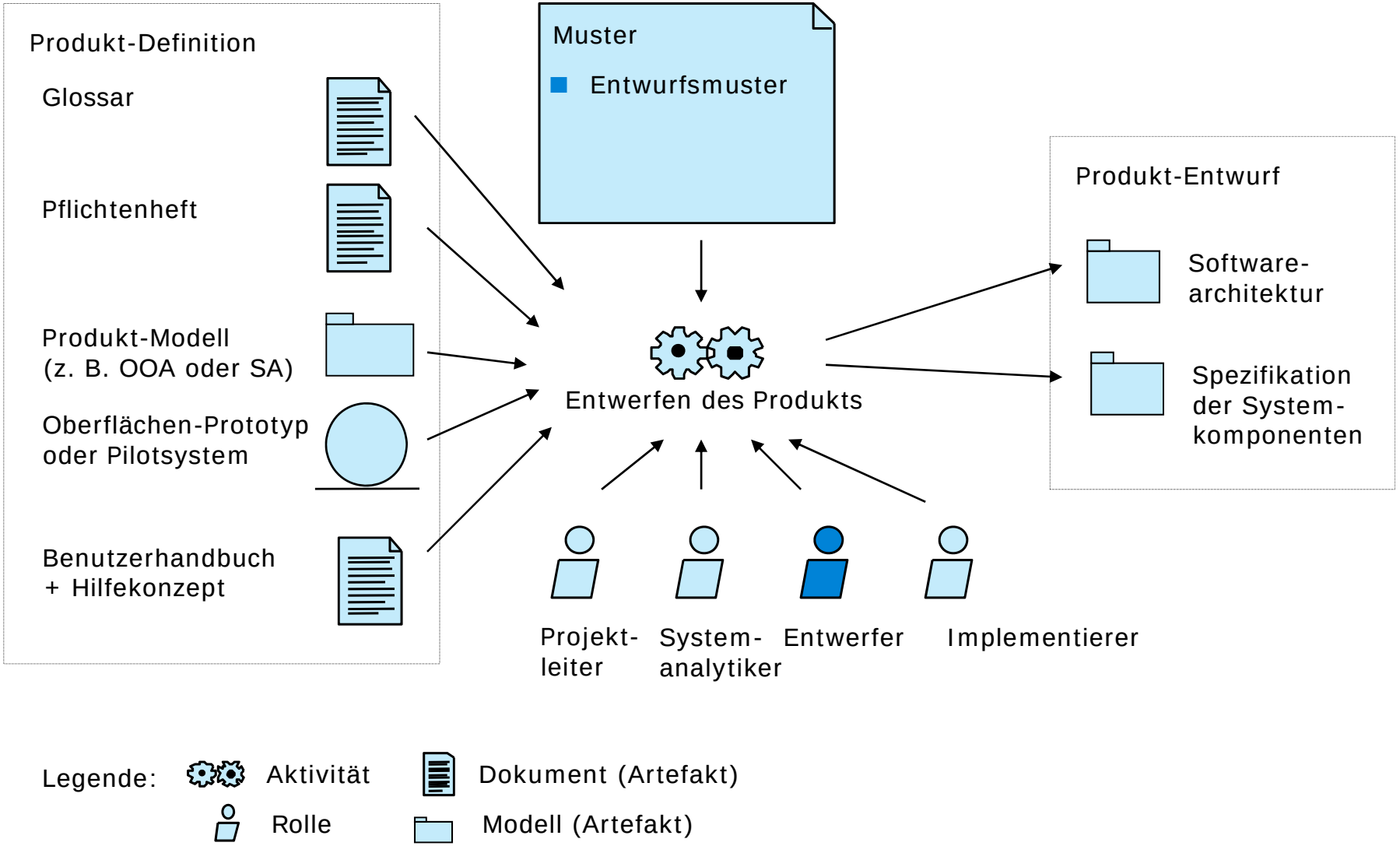
Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



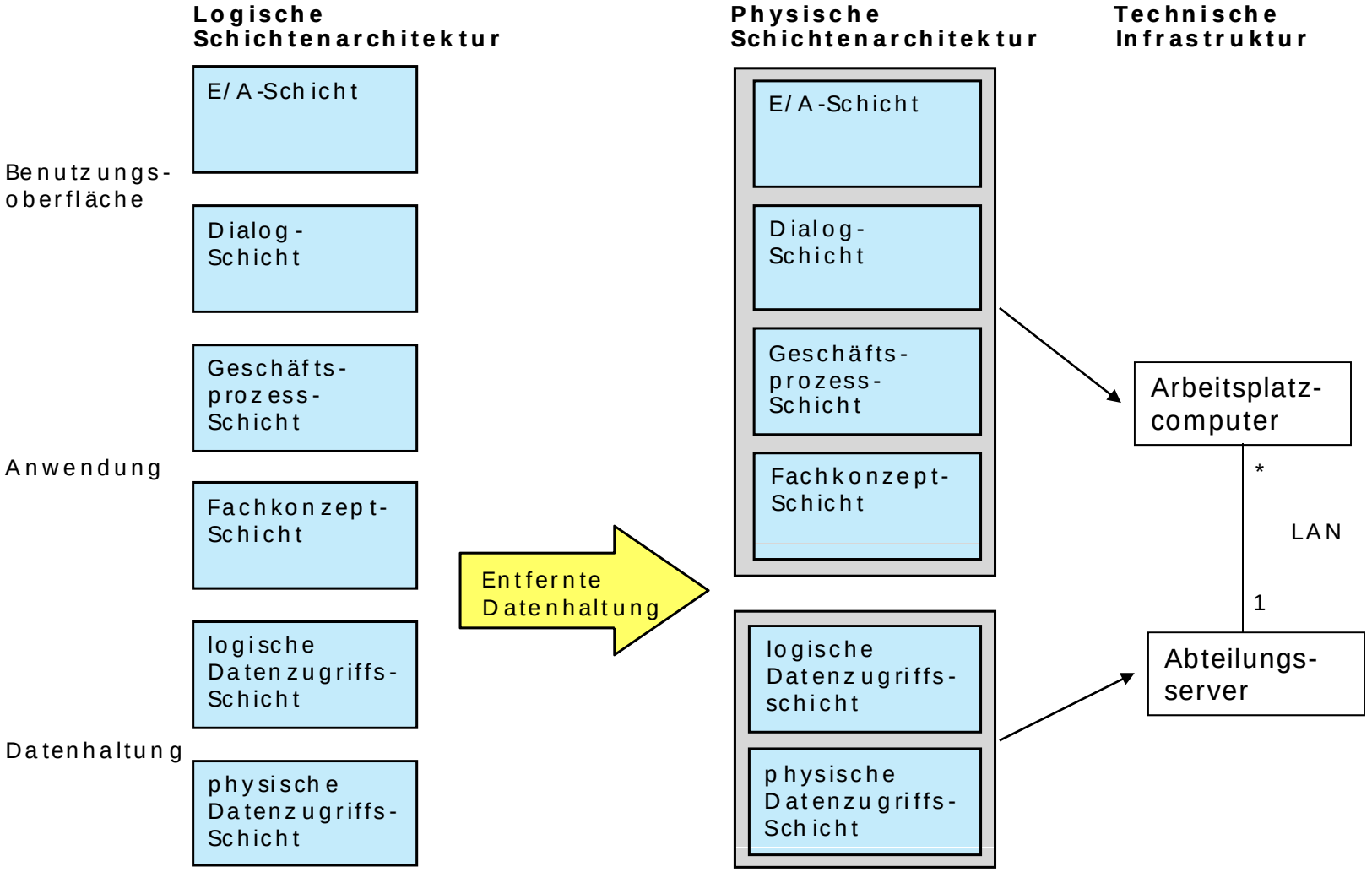
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Übersicht Entwurfsphase



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

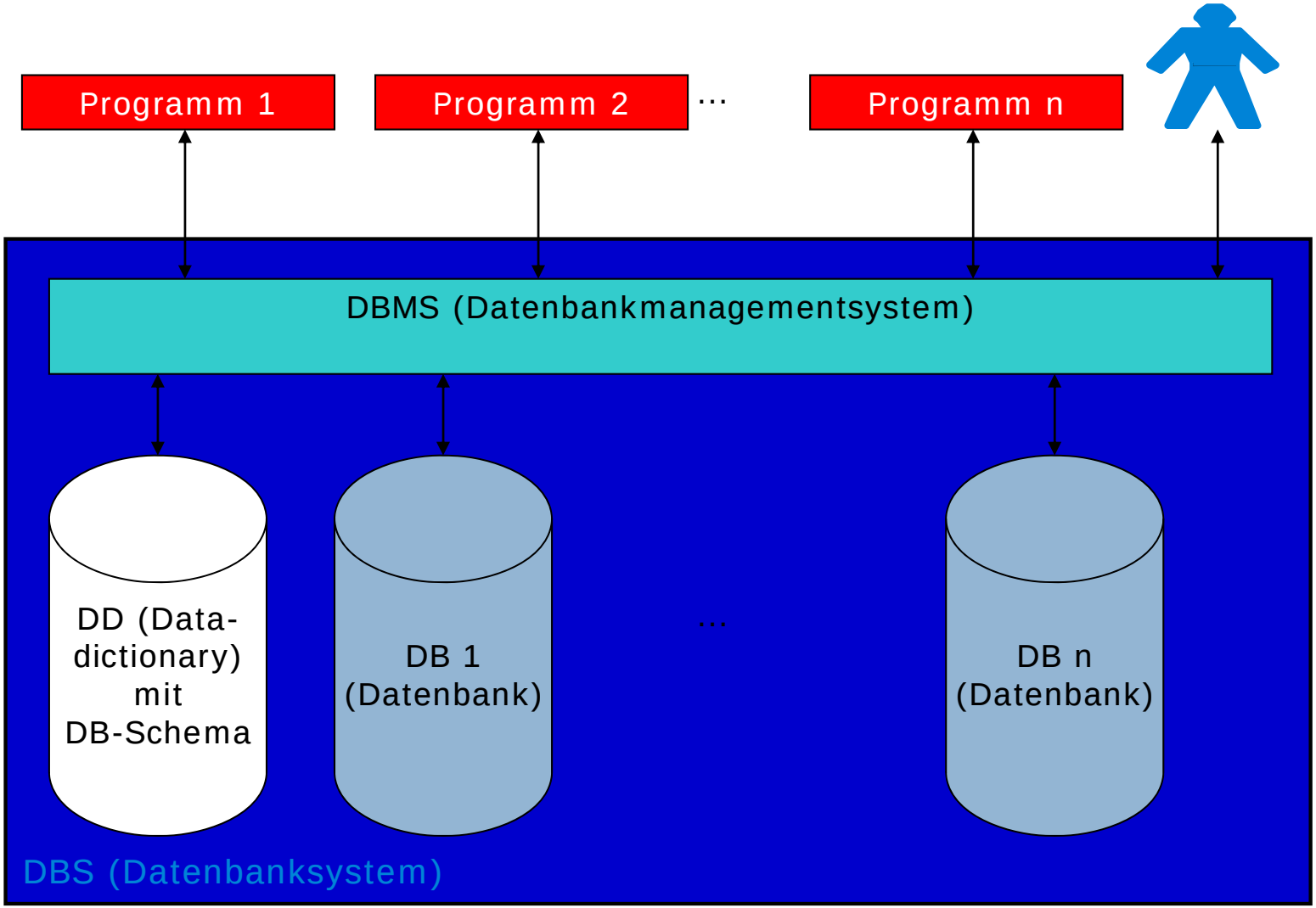
Schichtenverteilung



- Logische Schicht
  - Um eine Anwendung verteilen zu können, muss sie in Schichten strukturiert sein.
  - Die Schichten einer Anwendung bezeichnet man auch als logische Software-Schichten.
- Physische Schicht
  - Logische Schichten stellen die Modularisierungseinheiten einer Anwendung dar.
  - Physische Schichten repräsentieren die Einheiten, die auf unterschiedliche Computerklassen (Arbeitsplatzcomputer, Abteilungsserver, Datenbankserver, Zentralcomputer) verteilt werden.
  - Die 3-Schichten-Architektur kann zu einer n-Schichten-Struktur verfeinert werden.
- Verteilungsmuster
  - Logische Schichten können mit Hilfe von Verteilungsmustern auf physische Schichten abgebildet werden.
  - Die Verteilungsmuster hängen davon ab, ob
    - eine Client/Server-Architektur oder
    - eine Web-Architektur zugrunde gelegt wird.

Konzept eines Datenbanksystems

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Relationales Datenmodell

- Darstellung von Entitäten, ihren Eigenschaften und Beziehungen untereinander in Relationen.
- Jede Relation kann als Tabelle dargestellt werden. Die Spalten tragen die Namen der Attribute. In den Zeilen sind die Elemente (Tupel) aufgeführt.

Tabellenname — Schlüsselattribut (unterstrichen)

Firma	Kurzname	Name	Adresse	Kurzmitteilung
	Softtech	Softtech GmbH	Bochum	–
	Innosoft	Innovation & Software	Dortmund	Beiliegend erhalten Sie unseren neuesten…
	…	…	…	…

Tabelle bzw. Schema

← Attribute
← Tupel (Inhalt)

Objektorientiertes Datenmodell

- Objekte werden unverändert gespeichert. Sie werden nicht in Tabellen transformiert
- Schema-Definitionssprache
  - Objekt-Definitionssprache ODL (Object Definition Language) der ODMG (ODMG = Object Database Management Group)
- Jede Klasse des OOA-Modells wird in ODL durch eine Schnittstellendeklaration (interface declaration) beschrieben.

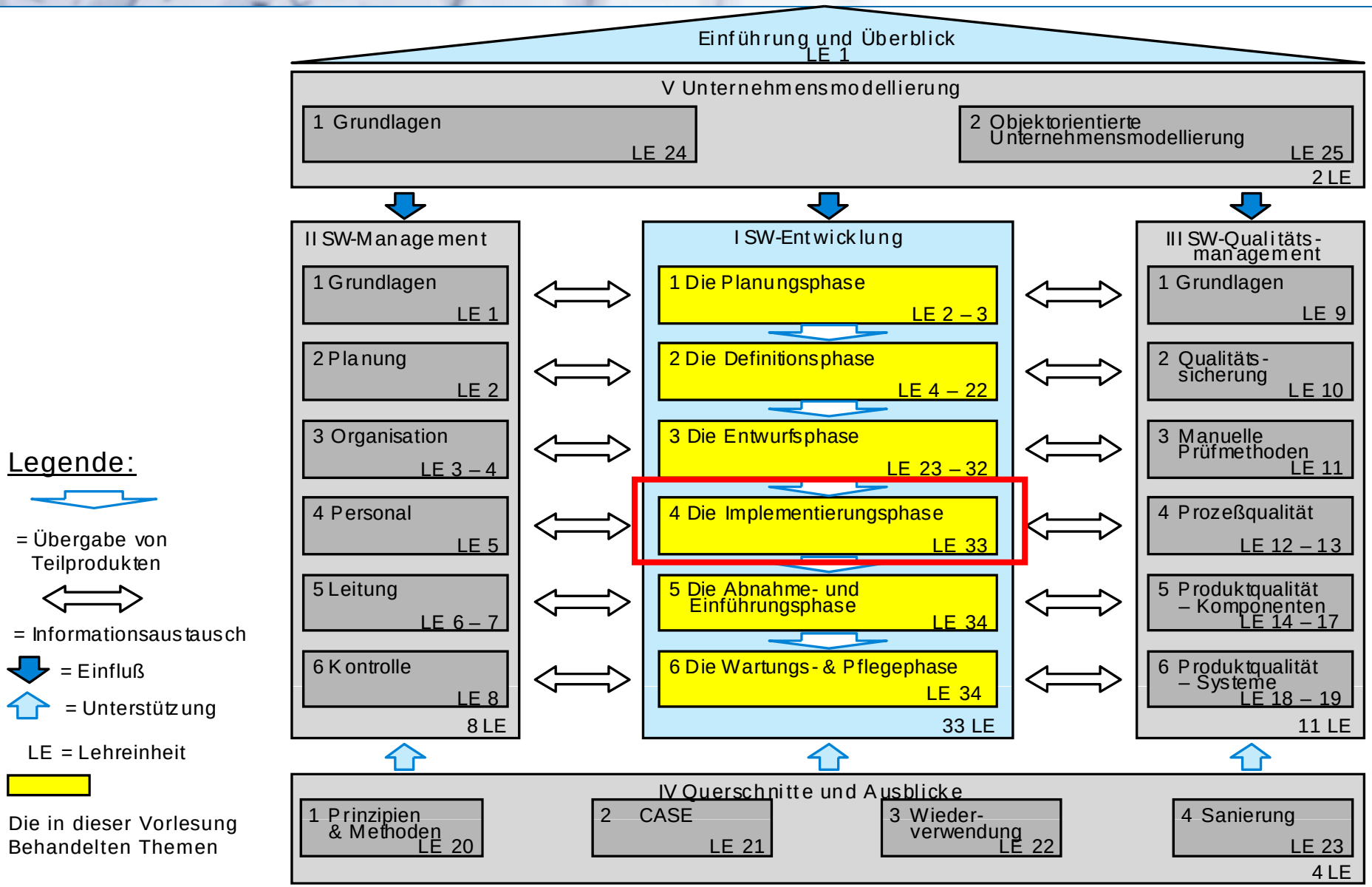
Objektorientiertes Design / Softwarekomponenten

- Grundlage für den objektorientierten Entwurf ist in der Regel das **OOA-Modell**.
- Das entstehende **OOD-Modell** wiederum bildet die Grundlage für die Implementierung.
- Die Implementierung erfolgt in einer oder mehreren konkreten **Programmiersprachen**.

- **Komponentenbasierte Softwareentwicklung**
  - Erlaubt die einfache, schnelle und preiswerte Herstellung individueller, integrierter Anwendungen durch Zusammenbau von vorgefertigten Halbfabrikaten bzw. Komponenten.
- Dazu benötigt man **Halbfabrikate**, die
  - im Allgemeinem kleiner als Anwendungen sind, d. h. einen stärkeren Komponenten- bzw. Bausteincharakter besitzen;
  - deutlich größer als Klassen sind, d. h. mehr Funktionalität kapseln;
  - Komplexität verbergen.
- Halbfabrikat (component-ware)
  - Ist ein **abgeschlossener, binärer Software-Baustein**, der eine anwendungsorientierte, semantisch zusammengehörende Funktionalität besitzt, die nach außen über Schnittstellen zur Verfügung gestellt wird.
  - Beim Entwurf des Halbfabrikats wurde auf hohe Wiederverwendbarkeit großer Wert gelegt.

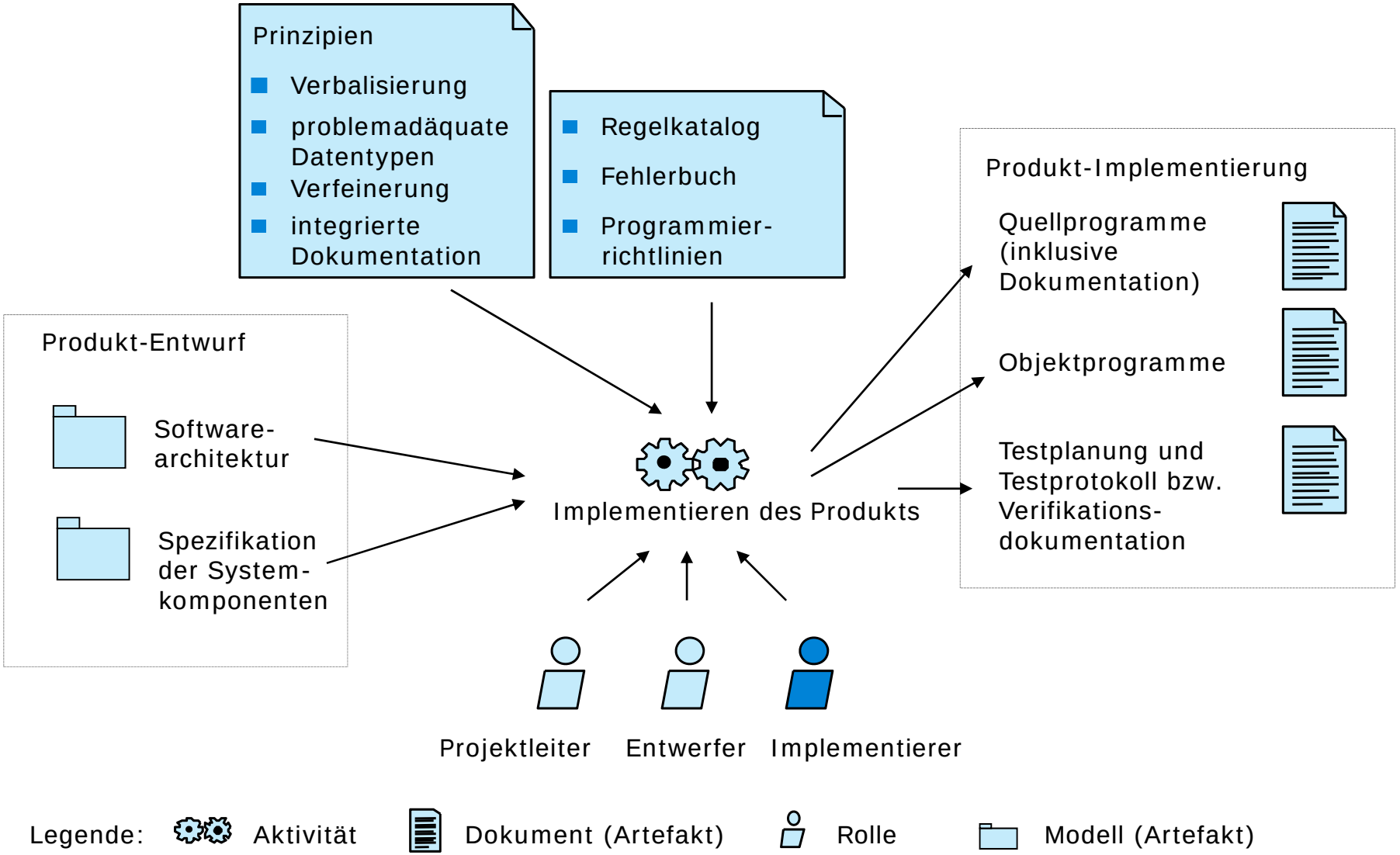
Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



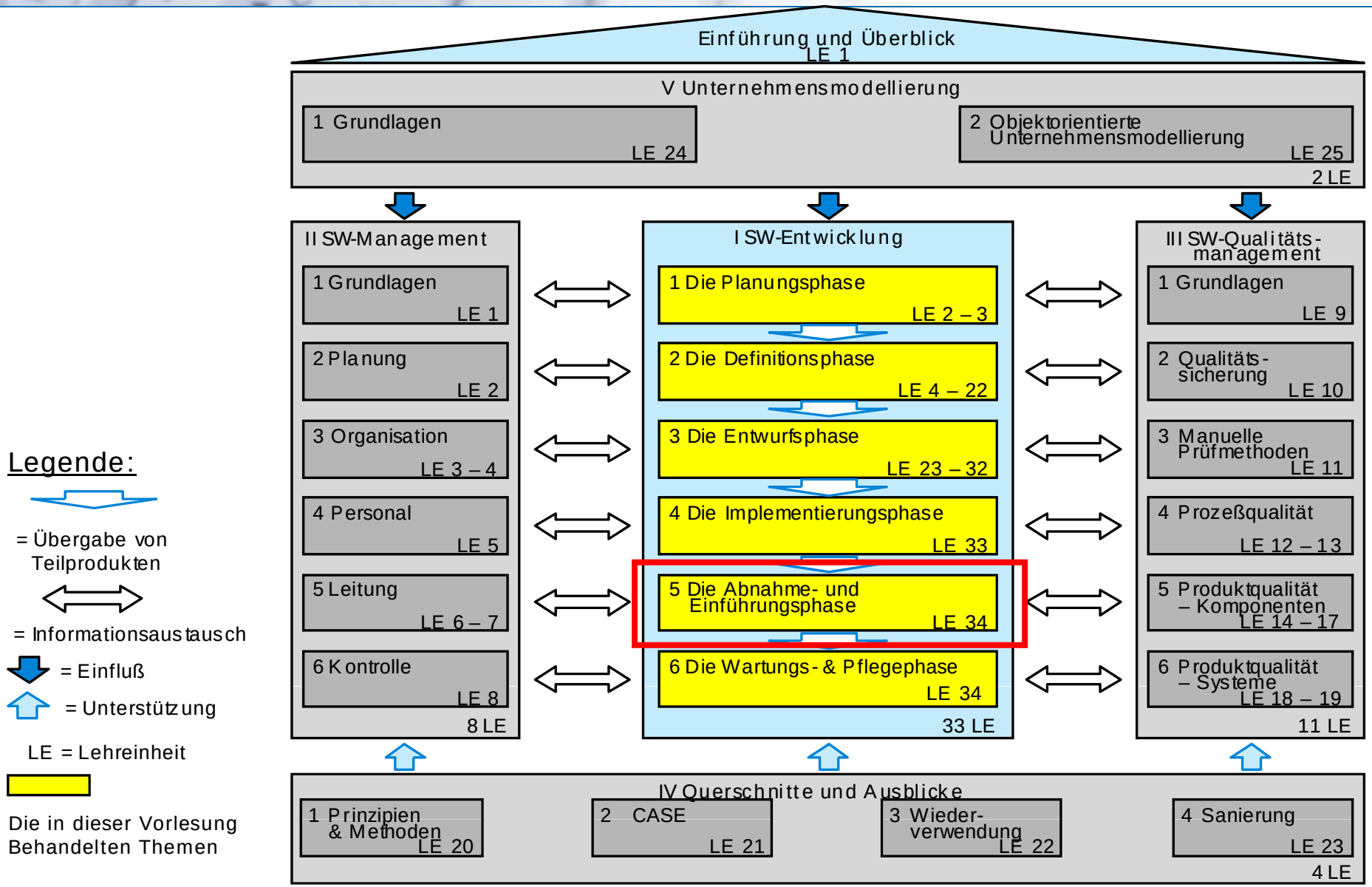
Übersicht Implementierungsphase

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



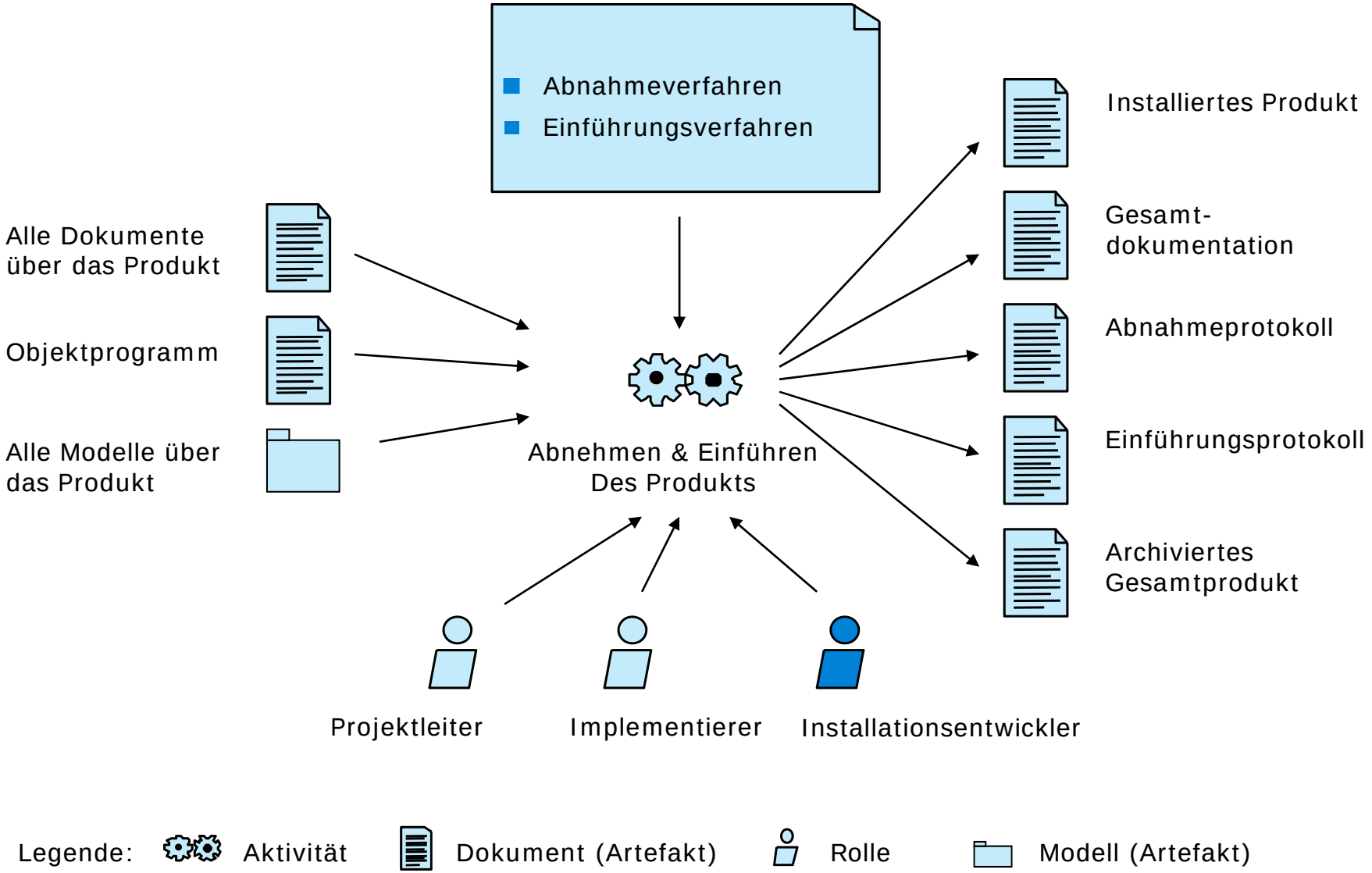
Abnahme- und Einführungsphase

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Abnahme- & Einführungsphase
  - Das fertiggestellte Gesamtprodukt wird abgenommen und beim Anwender eingeführt, d.h. in Betrieb genommen
  - Ab diesem Zeitpunkt unterliegt das Produkt dann der Wartung & Pflege
- Abnahmephase: Tätigkeiten
  - Übergabe des Gesamtprodukts einschließlich der gesamten Dokumentation an den Auftraggeber.
  - Mit der Übernahme verbunden ist i. Allg. ein Abnahmetest
  - Innerhalb einer Abnahme-Testserie ist es auch sinnvoll Belastungs- oder Stresstests durchzuführen
  - Das Ergebnis der Abnahmephase ist ein Abnahmeprotokoll.
- Einführungsphase: Tätigkeiten
  - Installation des Produkts: Einrichtung des Produkts in dessen Zielumgebung zum Zwecke des Betriebs.
  - Schulung der Benutzer und des Betriebspersonals: Nach der Installation des Produkts sind die Benutzer in die Handhabung des Produkts einzuweisen.
  - Inbetriebnahme des Produkts: Übergang zwischen Installation und Betrieb

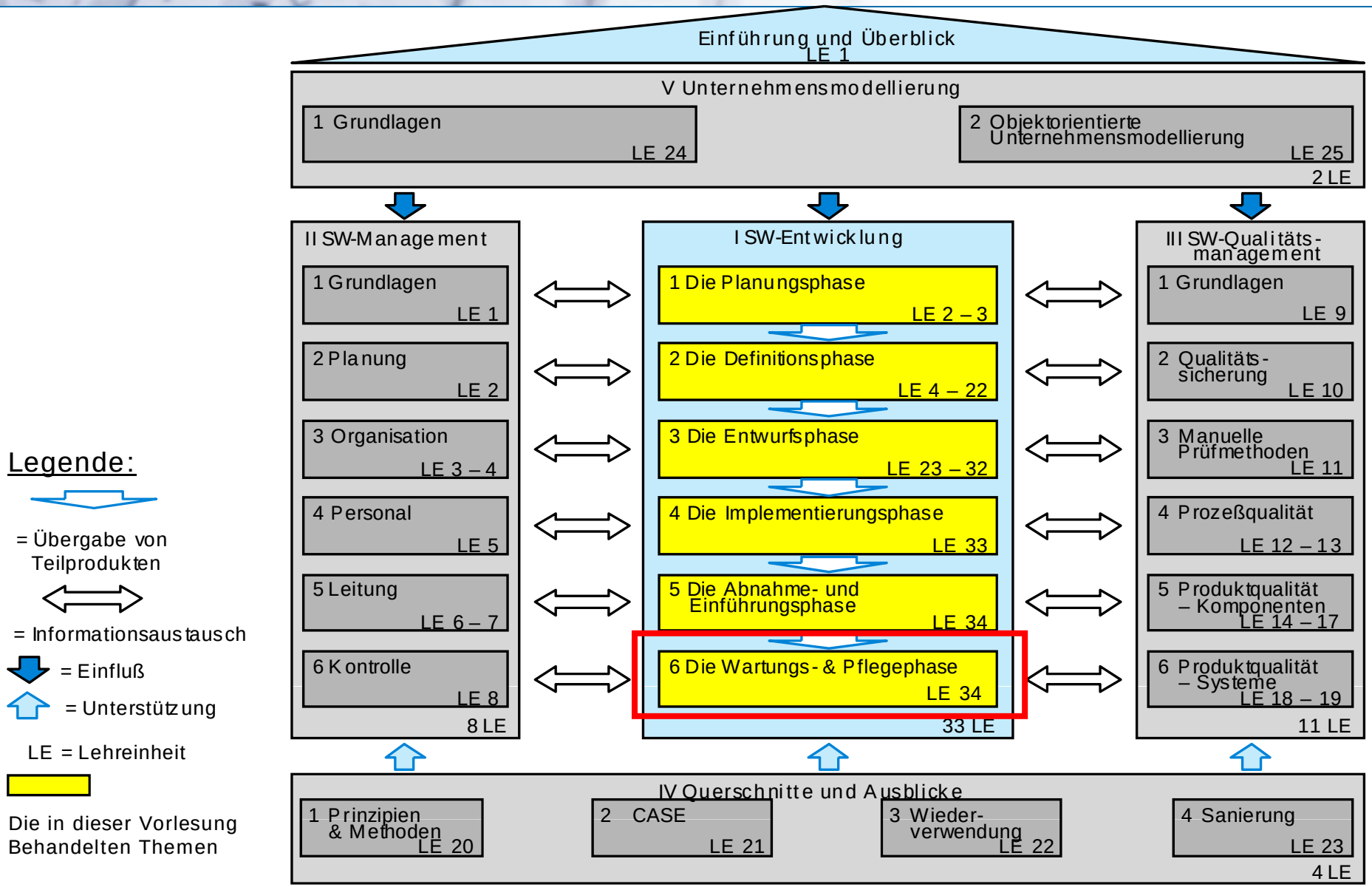
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Übersicht Abnahme- und Einführungsphase



Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Wartungs- und Pflegephase

- Wartung und Pflege
  - Beginnt mit der erfolgreichen Abnahme und Einführung einer Software
- Nach der Inbetriebnahme eine Produktes...
  - treten im täglichen Betrieb Fehler auf
  - ändern sich die Umweltbedingungen (neue Systemsoftware, neue Hardware, neue organisatorische Einbettung, ...)
  - entstehen neue Wünsche und Anforderungen (neue Funktionen, geänderte Benutzungsoberfläche, erhöhte Geschwindigkeit, ...)
- Alterung von Software
  - Software, bei der nicht ständig Fehler behoben und Anpassungen sowohl an die Umwelt als auch an neue Anforderungen vorgenommen werden, altert und ist irgendwann veraltet
  - Sie kann dann nicht mehr für den ursprünglich vorgesehenen Zweck eingesetzt werden
  - »Software veraltet in dem Maße, wie sie mit der Wirklichkeit nicht Schritt hält« [Sneed 83].
- Zwei Kategorien der Wartungs- & Pflege
  - korrektive Tätigkeiten (Wartung)
  - progressive Tätigkeiten (Pflege)

Übersicht Softwaretechnik

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme