

Konzeptionelles Modell

Brauchen wir Normalformen?

Lernziele

- Sie wissen, was ein konzeptionelles Modell ist und wie dieses schrittweise erstellt, ergänzt oder angepasst werden kann.
- Sie kennen den praktischen Nutzen der Normalformen und in etwa deren Struktur
- Sie wissen, dass Normalformen nicht immer die beste Lösung sind

Was ist eine Entität?

- Definition
 - **Entität = individuelles, identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungs-Welt**
- Beispiele
 - ◆ ein Ereignis das Flugzeug landet in NY
 - ◆ eine Person ein Mitglied der Klasse
 - ◆ eine Schule eine Ingenieurschule
 - ◆ ein Fach Informations-Systeme
 - ◆ ein reales Objekt ein Auto
 - ◆ ein abstraktes Konzept Fachgebiet

KDBD-Sicht der Entität

- *Immer* : eindeutig identifizierbar (->Schlüssel)
- *Immer* : darstellbare Einheit, unabhängig von der Existenz anderer Entitäten
- *Meistens* : Einheit, für die Informationen gesammelt werden, die auf einem Speichermedium festgehalten werden sollen

Was eine Entität ist, hängt also ***ganz entscheidend*** von unserem Gesichtspunkt ab: was für den einen eine Entität ist, ist für den Andern ein Attribut!

und was ist mit der Objektsicht?

- klassische Methoden zur Ermittlung von Klassen (und Entitäten!)
 - Textanalyse
 - Substantive sind Kandidaten für Klassen
 - Adjektive sind Kandidaten für Variabeln
 - Verben sind Kandidaten für Methoden
 - Realitätsbeobachtungen
 - Benutzersichtanalysen
 - Generalisierung und Spezialisierung
 - Ereignisanalyse
- Objekte = Entitätsmengen + Methoden

Eigenschaften: Definition

- **Eigenschaft** wird einer Entität zugeordnet und ermöglicht deren
 - Charakterisierung
 - Klassifizierung
 - unter Umständen : eindeutige Identifizierung
- Eine Eigenschaft hat einen Namen und einen (oder mehrere) Eigenschaftswerte

Beispiel:

Eigenschaft : Name; Eigenschaftswert : Peter; Entität : Person

Faktum : Definition

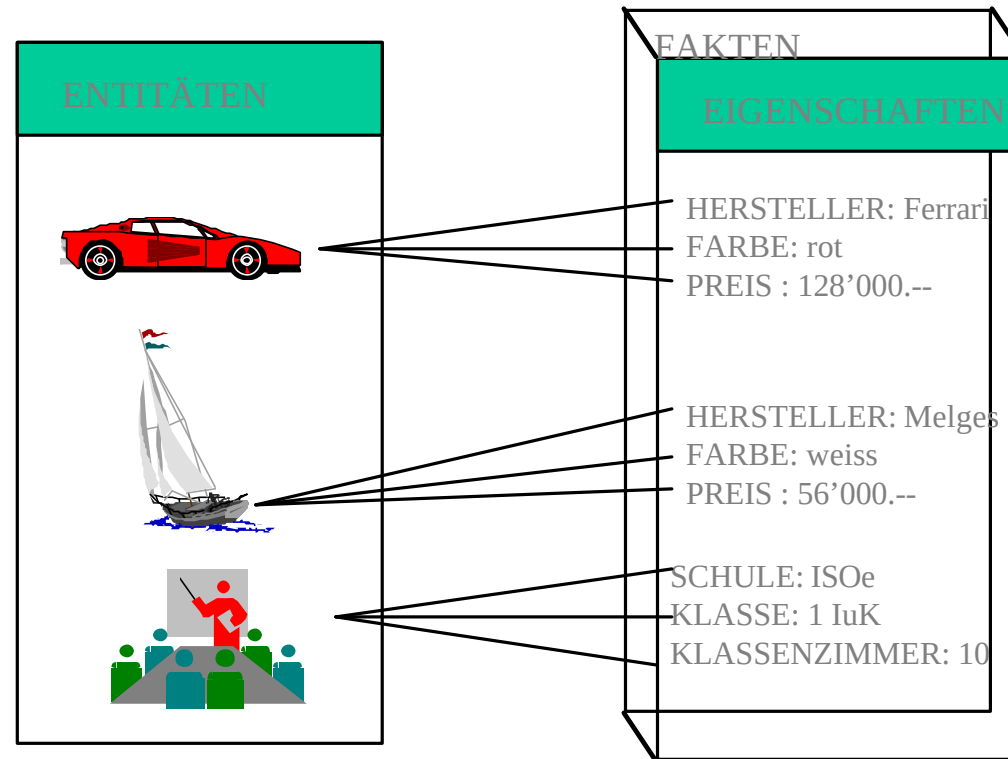
- **Faktum** ist eine Behauptung, derzufolge eine Entität für eine Eigenschaft einen bestimmten Eigenschaftswert aufweist

Beispiel

- Entität: Kind
 - NAME: Heinz
 - ALTER: 12
- Entität: Schüler
 - NAME: Heinz
 - ALTER: 12
- Faktum = Zuordnung <Eigenschaft, Eigenschaftswert> zu einer Entität
„es ist ein Faktum, dass Heinz 12 Jahre alt ist“

Entität - Eigenschaften

- Visualisierung



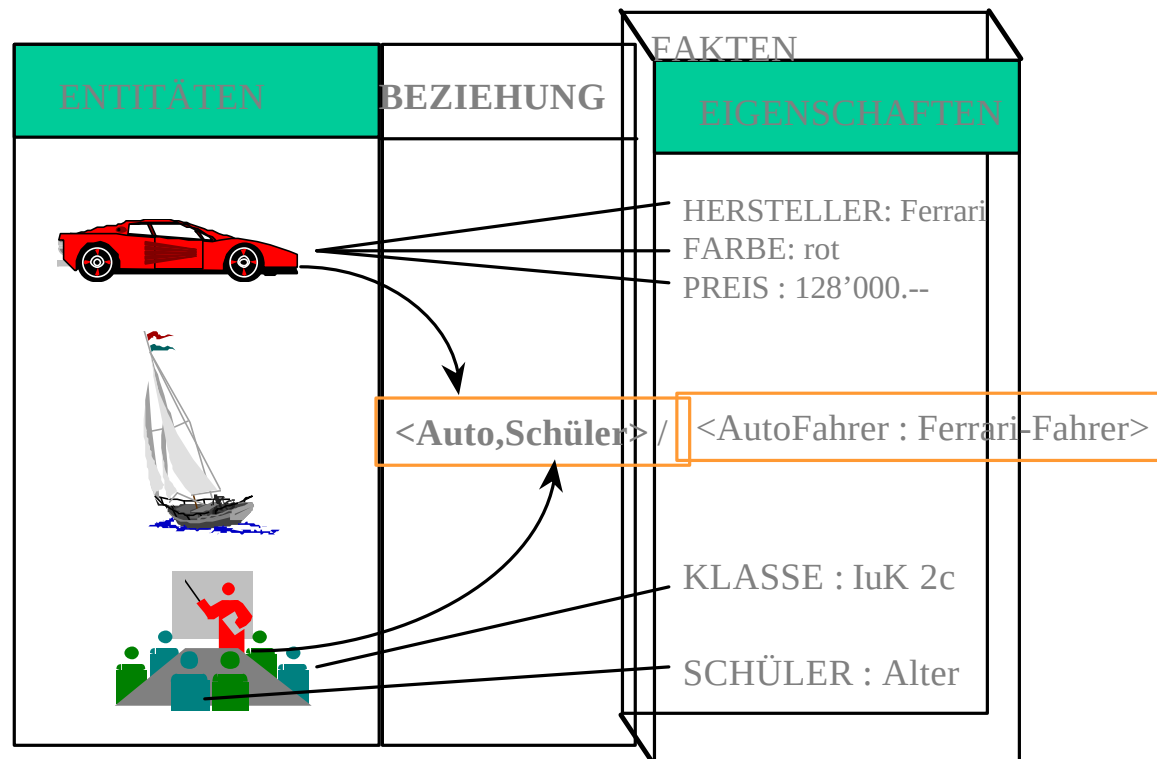
Was ist eine Beziehung?

- **Beziehung** assoziiert wechselseitig zwei (eventuell mehr als zwei) Entitäten

Beispiele

- Einfache Beziehungen:
 - Schüler einer Ingenieurschule
 - Flug einer Fluggesellschaft
- Mehrfach-Beziehungen:
 - Produkt(e) in einem Supermarkt
- Bedingte Beziehungen:
 - Produkt(e)einer Bäckerei führt eventuell Milch

Visualisierung der Beziehung



- *Tupel : Liste von Werten*

Beispiel: < 12S2, Hans, Zürich, KZO, low profile, MiniTower >

- *Relation : Menge von Tupeln, in der Regel tabellenförmig angeordnet*

Beispiel:

<132, Hans Meier , Klusstrasse 12 , Balsthal >

<324, Peter Heinzer, Bergweg 3 , Gigswil >

<543, Karl Born , Nathalienweg 5 , Ribe >

Schlüssel einer Relation

- **Schlüssel (einer Relation R) :**
 - Attribut, mit dessen Werten die Tupel der Relation R identifiziert werden können.
 - Unter Umständen lässt sich diese eindeutige Identifizierung **nur mit einem zusammengesetzten Schlüssel**, bestehend aus einer **minimalen** Attributskombination erzielen.
 - Ein zusammengesetzter Schlüssel ist immer dann minimal, wenn die Schlüsselkomponenten wechselseitig miteinander in Beziehung stehen.

Beispiel: PERSON PE# AHV# NAME ...
 (2 Schlüssel) 222 420 Britta ...
 (1:1) : jede Person hat eine Personalnummer und eine AHV Nummer

Normalisierung

- bei der Definition von Datenstrukturen sind bestimmte Gesetzmässigkeiten zu berücksichtigen
 - Ergebnis : Normalformen
- Normalformen
 - Vorteile:
 - keine Anomalien
 - Redundanzen eliminiert
 - Nachteil
 - Performance leidet sicher unter Normalisierung, aber die Daten sind dafür konsistent
 - konzeptioneller Aufwand

Ziele der Normalisierung

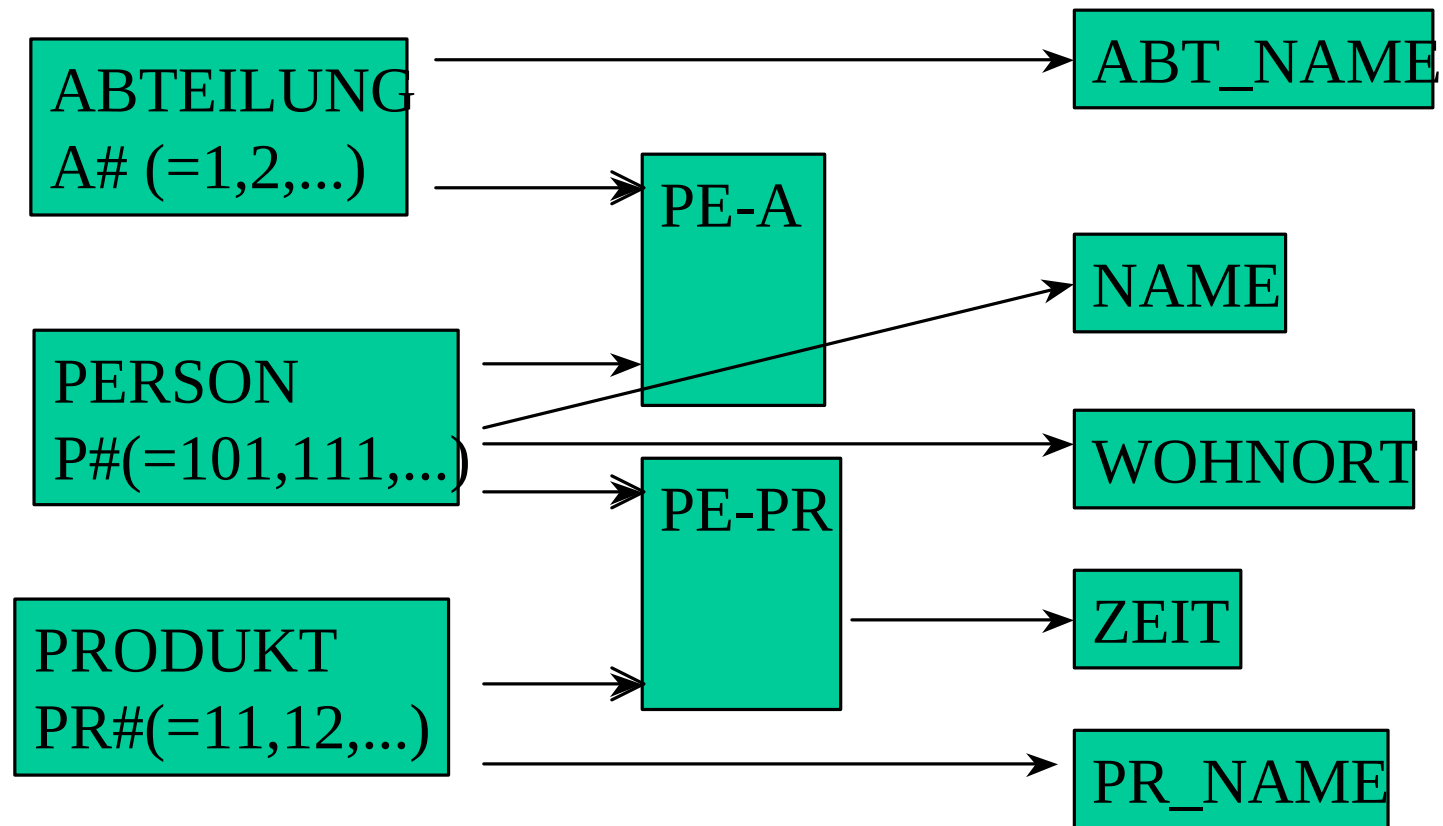
- Elimination von **Redundanz** (=mehrfaches Festhalten eines Faktums)
- Elimination von Schwierigkeiten (**Anomalien**), die im Zusammenhang mit Speicheroperationen wie Einschieben (Inserts), Löschen (Delete), Modifikationen (Updates) auftreten können
- das eindeutige Festhalten realitätskonformer Sachverhalte:
 - das Ermitteln von Datenstrukturen und das Festhalten fundamentaler Bedingungen, die die Datenintegrität garantiert

- GESUCHT : REALITÄTSMODELL, DAS DIE FOLGENDEN ENTITÄTSMENGEN ZU BERÜCKSICHTIGEN HAT :
 - PERSON (MITARBEITER)
 - ABTEILUNG
 - PRODUKT

Realitätsbeobachtung

- REALITÄTSBEOBACHTUNG:
 - EIN MITARBEITER HAT JEDERZEIT EINEN NAMEN
 - SOWIE EINEN WOHNORT
 - ER IST IN EINER ABTEILUNG TÄTIG UND
 - PRODUZIERT IN DER REGEL MEHRERE PRODUKTE
 - DABEI IST PRO PRODUKT UND PERSON EINE BESTIMMTE ZEIT ERFORDERLICH
 - JEDE ABTEILUNG HAT EINEN NAMEN
 - JEDES PRODUKT HAT EINEN NAMEN
 - IN EINER ABTEILUNG SIND MEHRERE PERSONEN TÄTIG
 - AN EINEM PRODUKT ARBEITEN IN DER REGEL MEHRERE PERSONEN

Visualisierung Abt-Pers-Prod



Probleme

- komplexe Handhabung : Anzahl Elemente im Tupel ist variabel
- hohe Datenredundanz : Fakten $\langle \text{Attribut}, \text{AttrWert} \rangle$ kommen mehrfach vor
 - Beispiel : $\langle \text{Abt\#}, \text{AbtName} \rangle$ kommt oft vor
- Anomalien in Speicheroperationen :
 - Beispiel : Änderungen bei $\langle \text{Abt\#}, \text{AbtName} \rangle$ in einem Personaltupel führt zu Dateninkonsistenzen

Anomalien : Beispiel

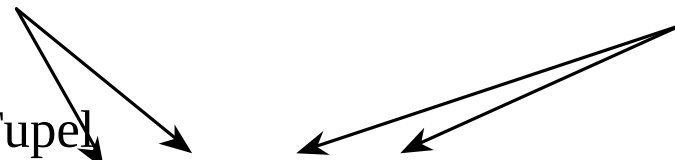
- Die Person arbeitet an mehreren Produkten, aber unterschiedlich lange
- Datensatz unnormalisiert:
 - $\langle P\#, \text{Meier}, \{\text{Flügel}, \text{Seitensteuer}\}, \{50\%, 25\%\}, \dots \rangle$
Die Zuordnung „Flügel - 50%“ und „Seitensteuer - 25%“ kann zwar gemeint sein, ergibt sich aber nicht aus den Daten oder deren Beziehungen

Relation Person unnormalisiert

	Name	Wohnort	Abteilung	AbtName	Prod	ProdName	Zeit
1	Hans Küng	Nänikon	2	Kleberei	1	Chassis	40'
					4	Verkleidung	20'
2	Beat Kern	Schwerzenbach	1	Administration	1012	Kreditoren	30'

- Anomalie:**

- es ist möglich ein Tupel **<102, Max Born, Wetzikon, 2, Optik, 1, Linsen, 60'>** einzufügen
- Sie verletzen dabei aber die Datenintegrität:
die Abteilung 2 (Kleberei) wird in diesem Tupel als Optik bezeichnet, das Produkt 1(Chassis) als Linsen; das System akzeptiert dies, da keinerlei Prüfungen definiert sind



Die erste Normalform : 1NF

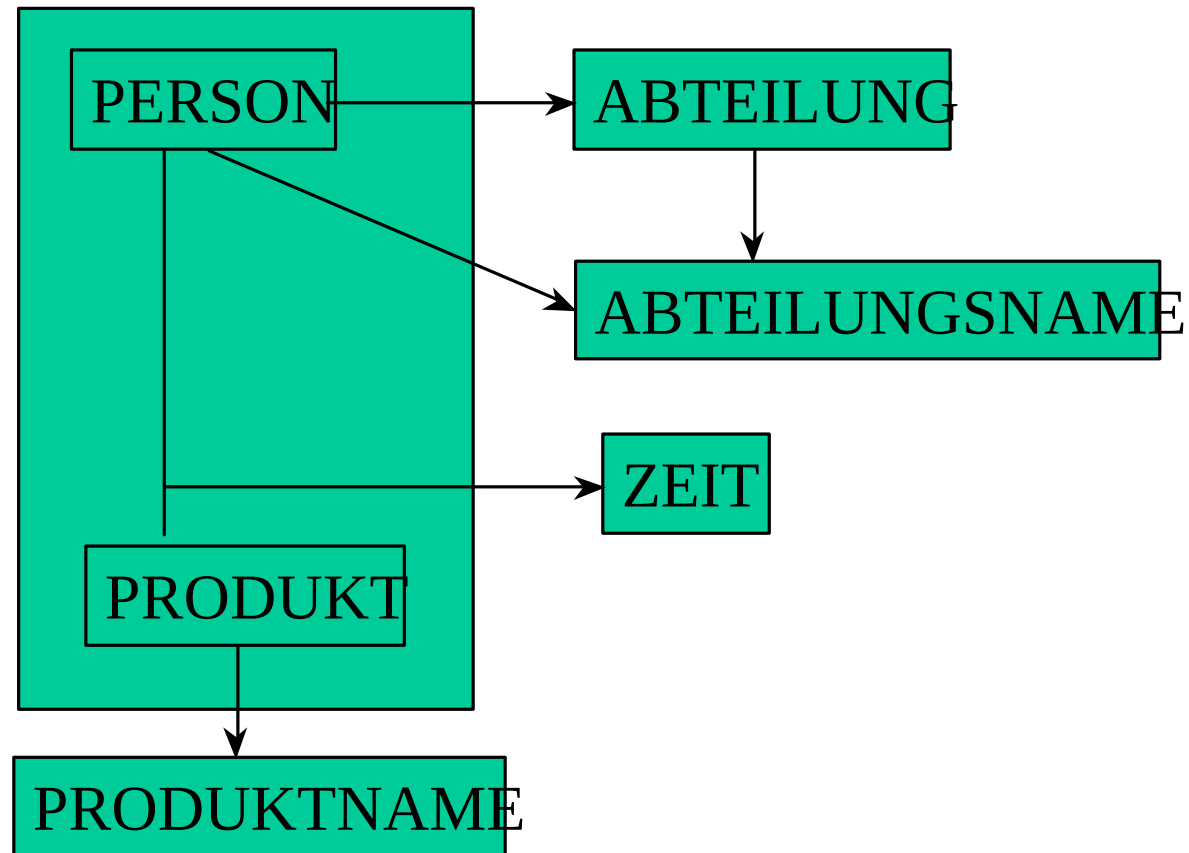
- gemäss Definition ist eine Relation in 1NF, wenn alle nicht dem Schlüssel angehörenden Attribute vom Schlüssel einfach (funktional) [oder konditionell] abhängig sind

Anomalien der 1 NF

Name	Wohnort	Abteilung	AbtName	Prod	ProdName	Zeit
1 Hans Küng	Nänikon		2 Kleberei	1	Chassis	40'
1 Hans Küng	Nänikon		2 Kleberei	5	Verkleidungen	20'
2 Beat Kern	Schwerzenbach		1 Administration	1012	Kreditoren	30'
2 Beat Kern	Schwerzenbach		1 Administration	1014	Debitoren	30'
3 Paul Odermatt	Uster		5 Entwicklung	9090	SystemMgr	60'

- **Anomalie:**
 - ich kann auch weiterhin ein Tupel mit inkonsistenten Daten einschieben!

Visualisierung



Die zweite Normalform : 2NF

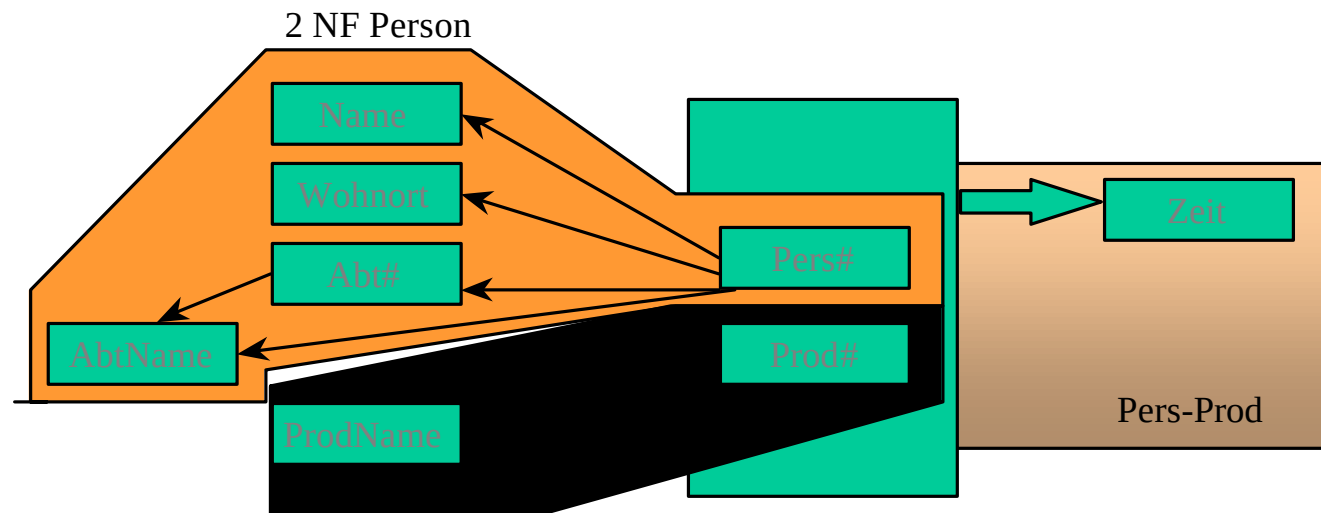
- gemäss Definition ist eine Relation in der 2NF , falls jedes dem Schlüssel nicht angehörende Attribut funktional abhängig ist vom **Gesamtschlüssel**, nicht aber von Teilschlüssel
- Was heisst das?
 - wir benötigen einen **zusammengesetzten** Schlüssel und mindestens ein Attribut

Die zweite Normalform : 2NF

- Beispiel:
 - in unser Relation
Person(Pers#, Name, Wohnort, Abt#, AbtName, Prod#, ProdName, Zeit)
sind folgende Abhängigkeiten erkennbar:
 - Pers# : Name, Wohnort
Prod# : ProdName
 - einzig “Zeit” ist von beiden Schlüsselattributen abhängig,
erfüllt also das 2NF Kriterium

Visualisierung der 2NF

- es hat sich bewährt, die Relation grafisch darzustellen:
da der Abteilungsname von der Person abhängt, kann die Abt nicht abgetrennt werden! Die Anomalie ($\langle \text{Abt\#}, \text{AbtName} \rangle$ nicht eindeutig) bleibt!



Anomalien der 2 NF



Person	Name	Wohnort	Abteilung	AbtName
1	Hans Küng	Nänikon	2	Kleberei
1	Hans Küng	Nänikon	2	Kleberei
2	Beat Kern	Schwerzenbach	1	Administration
2	Beat Kern	Schwerzenbach	1	Administration
3	Paul Odermatt	Uster	5	Entwicklung
3	Klaus Ehrlich	Walisellen	2	Informatik
100	Peter Schäuble	Schwerzenbach	3	Informatik
110	Hans-Peter Frei	Zollikon	2	Informatik

- 1) ein Updaten der Abteilung in der Personen-Relation kann weiterhin zu Anomalien führen
Grund : <Abteilung; AbtName> [eindeutig] können mehrdeutig werden
- 2) ein Insert eines Tupels, wie z.B. 3 Klaus Ehrlich... ist dagegen NICHT möglich, weil der Schlüssel 3 bereits besetzt ist!
- 3) Inserts mit noch nicht existierenden Schlüssel (100, 110) sind jedoch möglich und können zu Inkonsistenzen führen. Warum?

Transitive Abhängigkeit

was ist der Grund für die weiterhin bestehenden Probleme?

Realität:

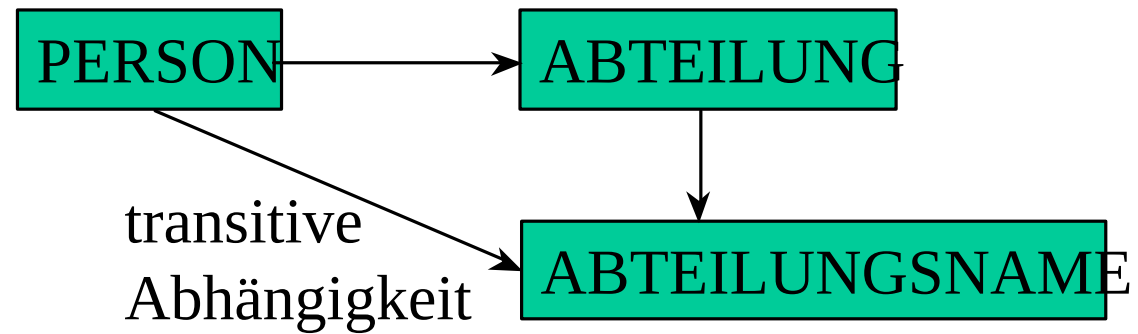
- 1) es gilt PersNr $\rightarrow$ AbtNr (jede Person gehört zu einer und nur einer Abt)
- 2) die Umkehrung stimmt aber NICHT!
- 3) der Abteilungsname hängt seinerseits eindeutig von der Abt.Nr ab

Transitive Abhängigkeit

- **Wir haben also folgende Situation:**
 - im Tupel <Pers#, PersName, Wohnort, **Abt#**, **AbtName**>
hängt ein Attribut (**AbtName**) eindeutig von einem Attribut ab (**Abt#**), welches **seinerseits** eindeutig vom Schlüssel abhängt; das Attribut hängt aber NICHT vom Schlüssel ab (und kann somit inkonsistent werden)

Dieser Sachverhalt wird als “transitive Abhängigkeit” bezeichnet!

Visualisierung



Transit: via Abteilung

Datenredundanz

Transitiv abhängige Attribute führen zu Datenredundanzen:
eine konkrete Instanzierung des Tupels $\langle \text{Abt\#}, \text{AbtName} \rangle$
(z.B. $\langle 2, \text{Kleberei} \rangle$) kommt in der Relation “Person”
mehrfach
vor!

Diese Redundanz kann zu den Inkonsistenzen führen!

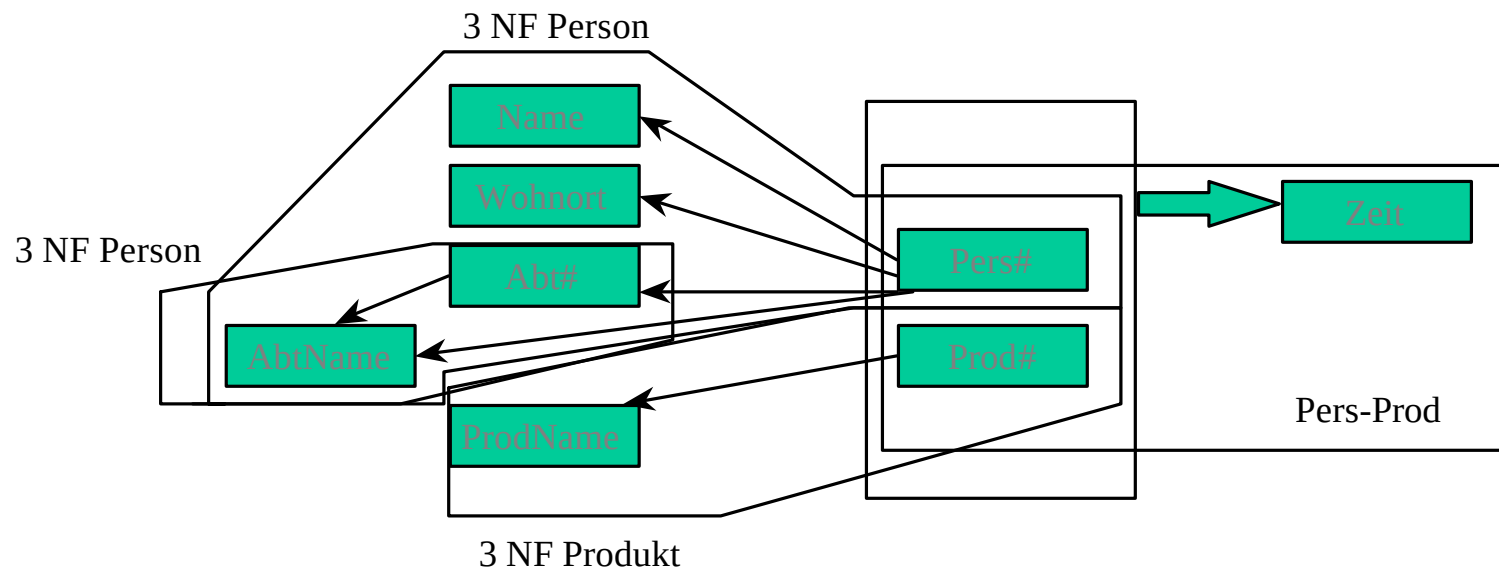
Was schlagen Sie als Ausweg vor?

Natürlich die Auftrennung des Tupels in unabhängigere
Tupel!

Das ist genau der Schritt zur dritten Normalform (3NF)!

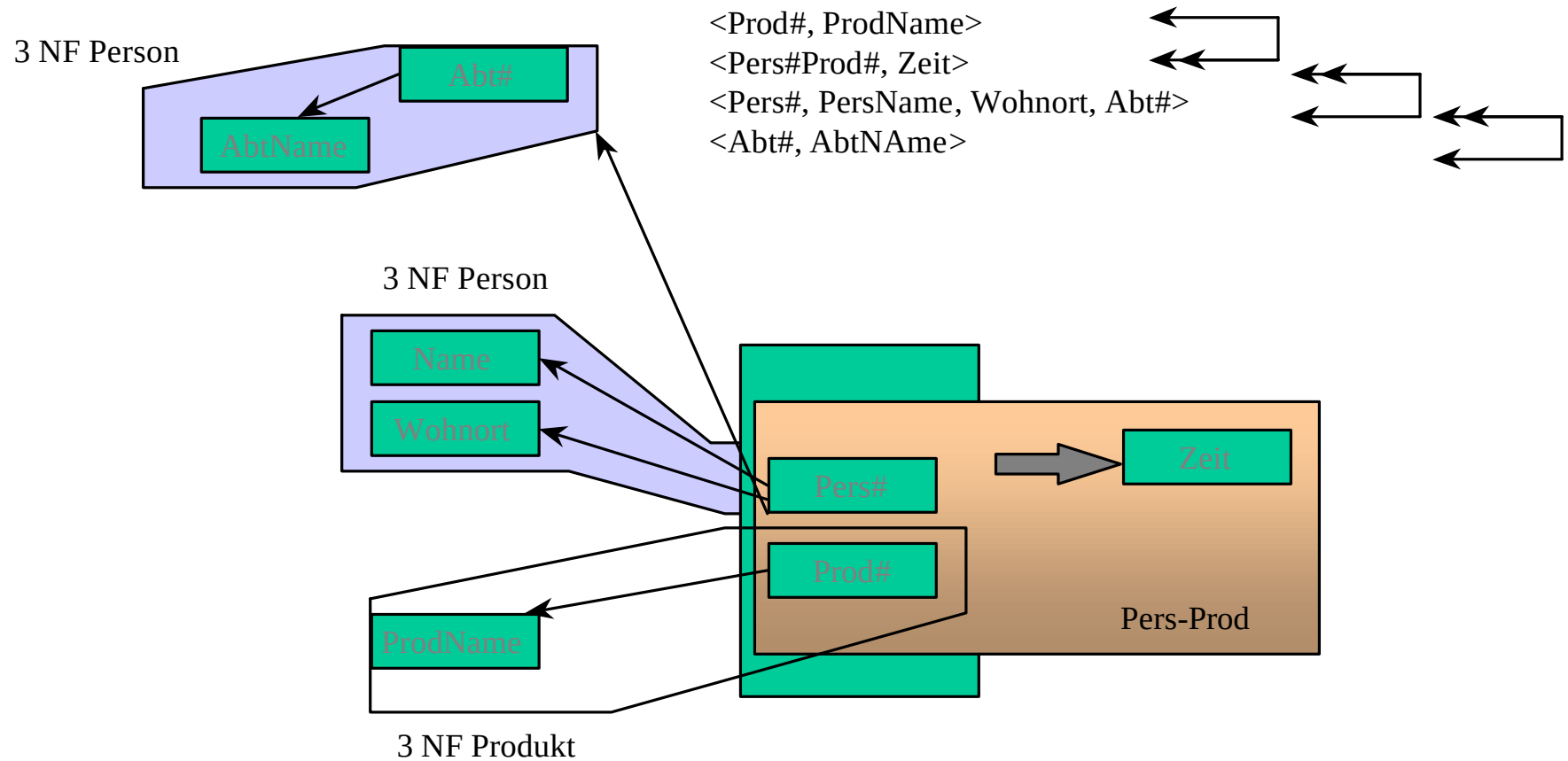
Anomalienbeseitigung : die 3NF

Die formale Definition der 3NF :
eine Relation ist in der 3NF, falls sie in der 2NF ist UND
keine transitiven Abhängigkeiten aufweist!





Anomalienbeseitigung : die 3NF



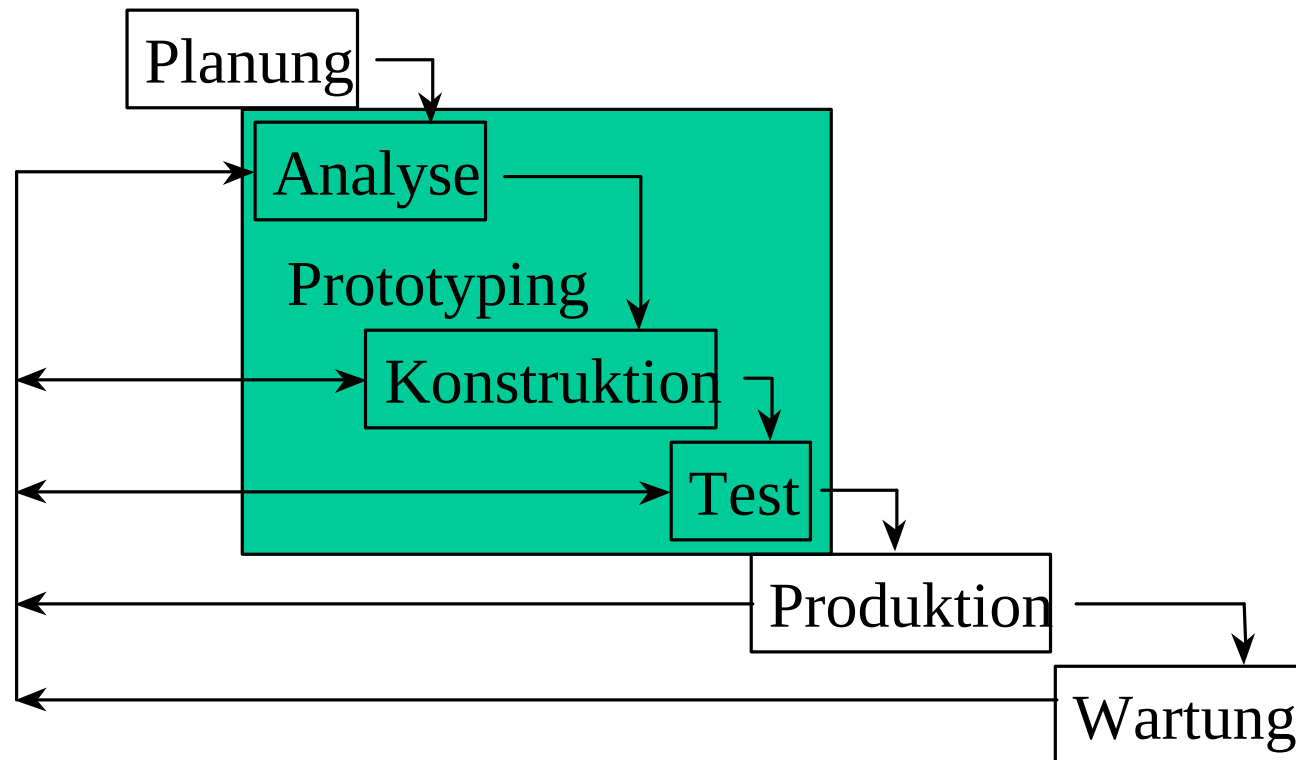
Alternative Wege

- Könnte man nicht alternativ vorgehen und andere Ergebnisse finden?
(primär müssen Sie sich an die Realitätsbeobachtung halten!)
- Versuchen Sie an Stelle der PERSON Relation eine andere Ausgangsrelation zu definieren und die Anomalien zu beseitigen!
Sie haben bereits Vorschläge für Relationen (weiter vorne)
- Ergebnis?

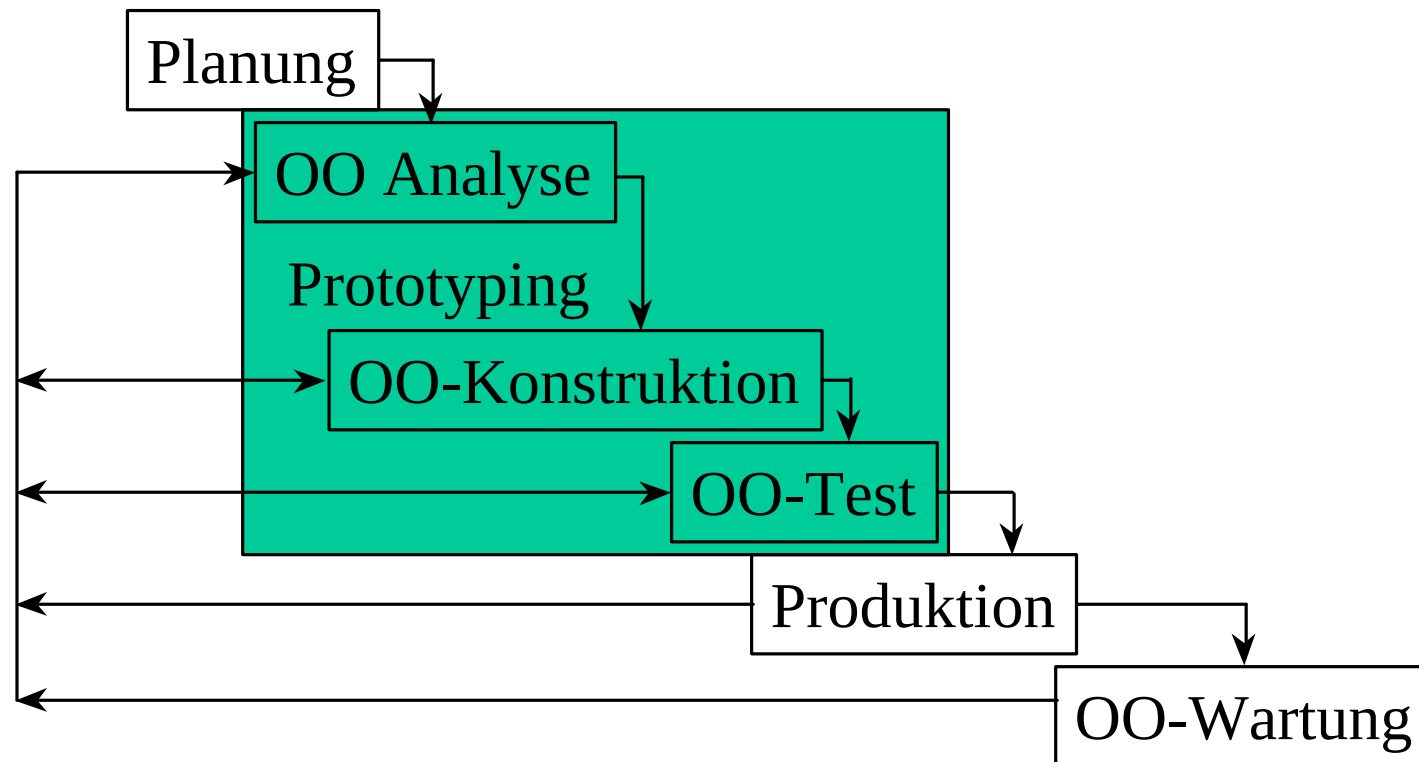
Vorgehensmodell(e)

Wie geht man in Datenbank-Projekten vor?
Wie gelangt man praktisch zur DB Struktur?

Konventioneller Anwendungsentwicklungsprozess



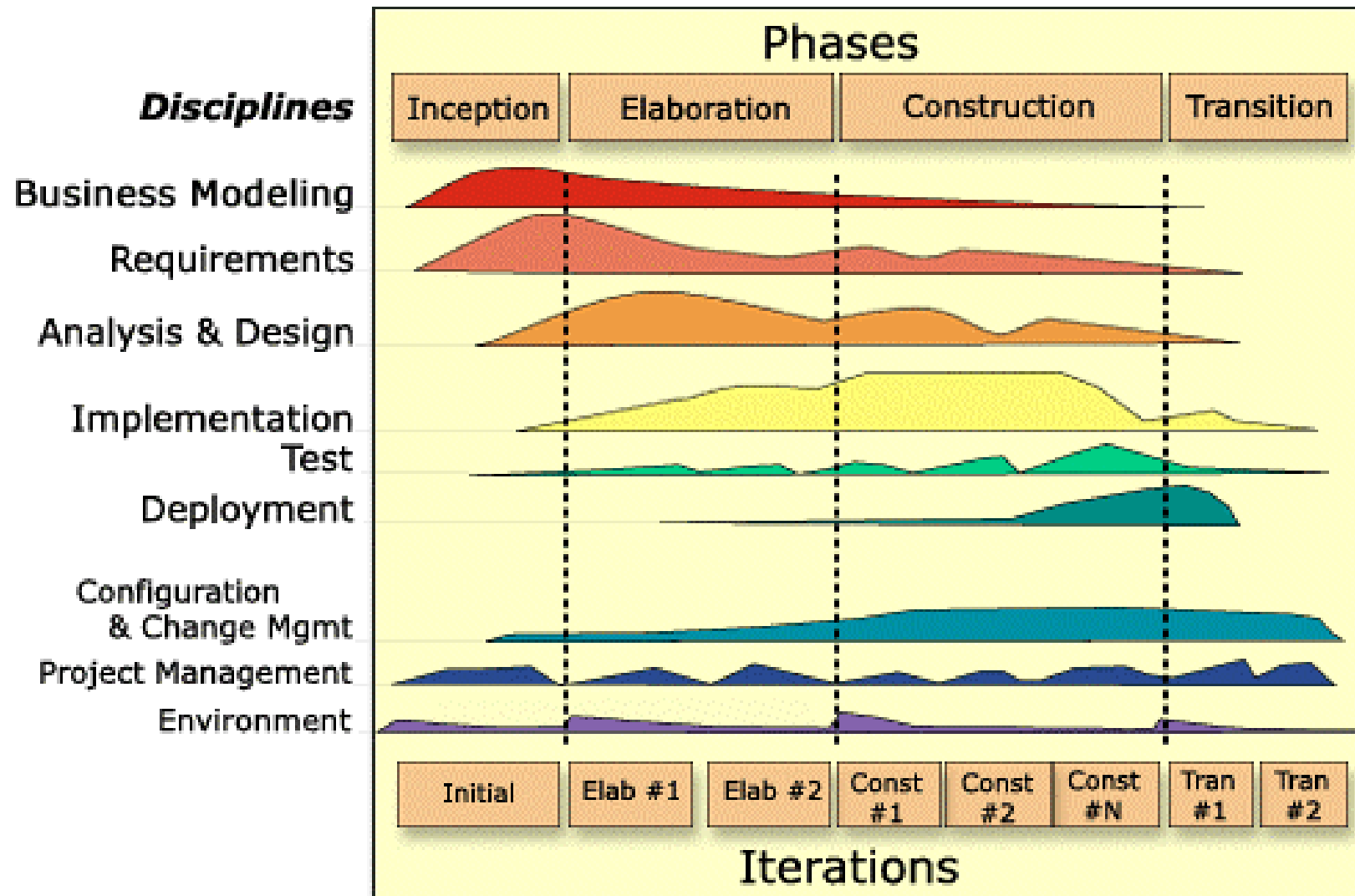
Objektorientierter Anwendungsentwicklungsprozess



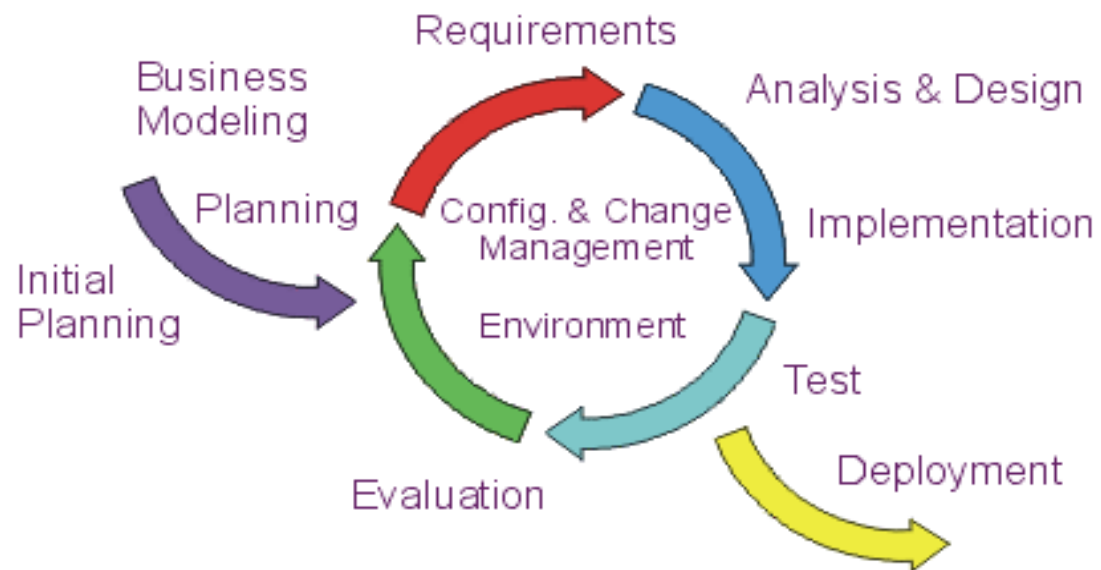
(Evolutionäres) Prototyping

- beim Prototyping wird möglichst früh eine vereinfachte Version erstellt, um in wichtigen, noch offenen Bereichen Erfahrungen zu sammeln
- beim evolutionären Prototyping (“Versioning”) stellt der Prototyp kein Wegwerfprodukt dar; es wird iterativ von einem Kern ausgehend entwickelt
- Heutige Standard- Entwicklungsmethode
 - Rational Unified Process
 - iteratives, mehrdimensionales Modell
 - gelingt auch mit mittelmässig begabten Teams
- Alternative
 - Extreme Programming
 - leichtgewichtiger Prozess
 - setzt in der Regel Experten voraus

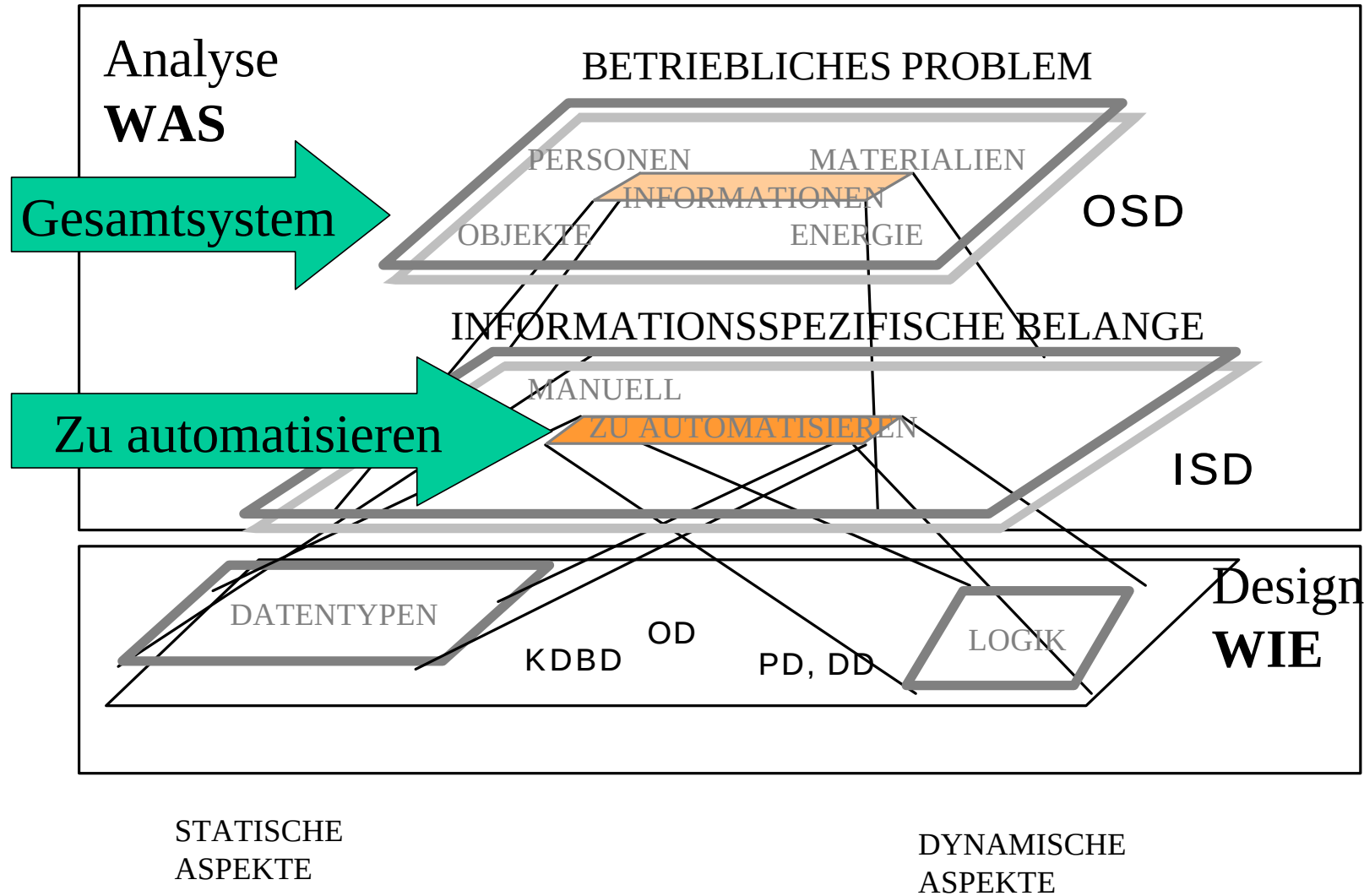
Rational Unified Process



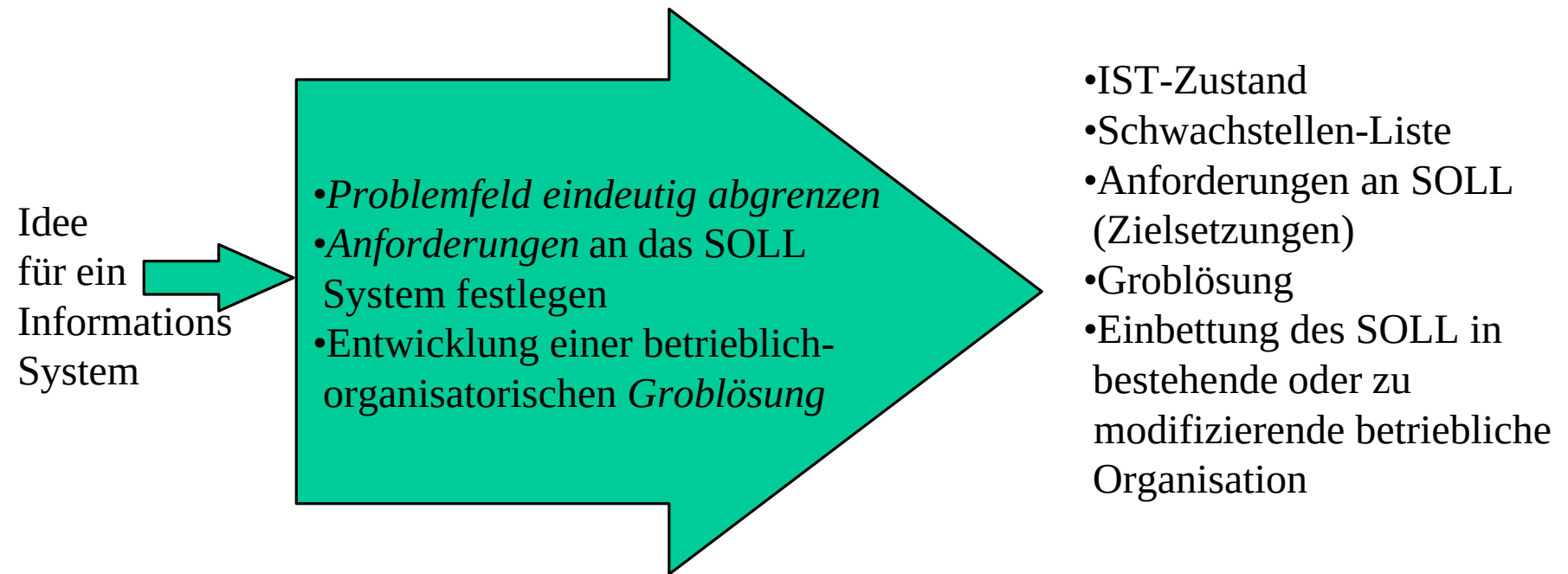
Objektorientierter Anwendungszyklus



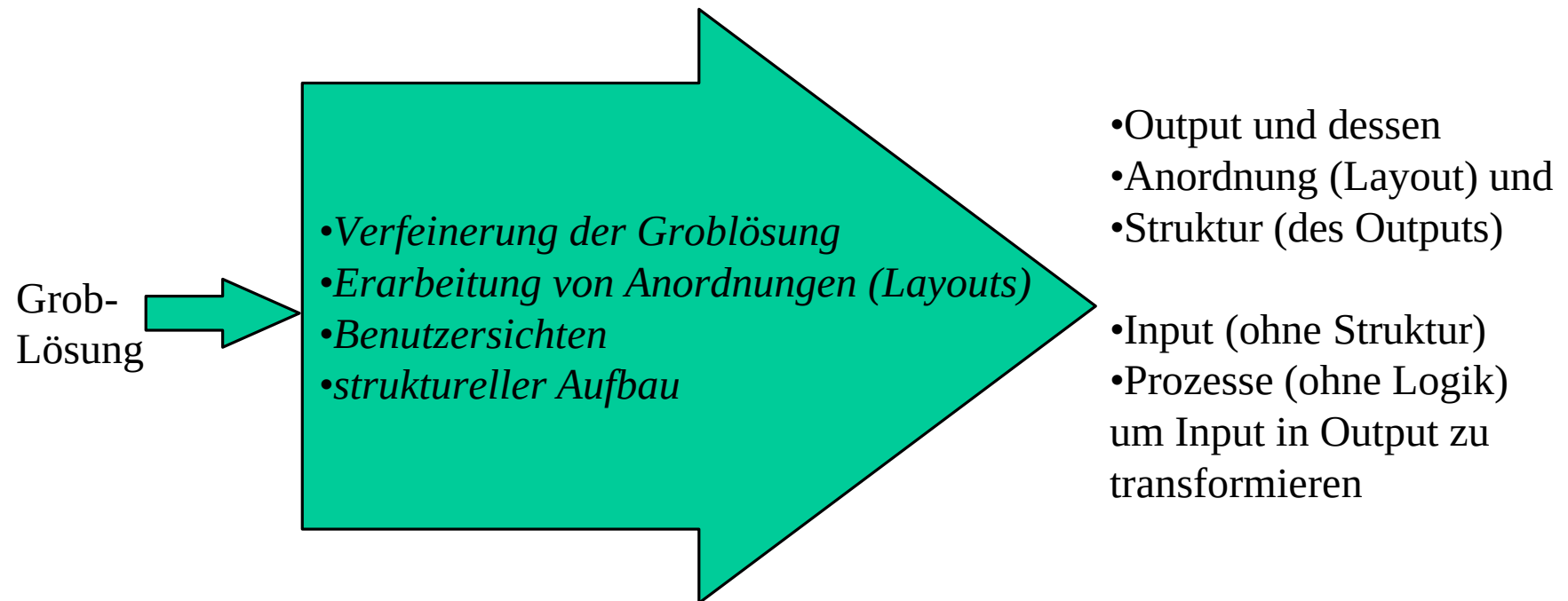
Betriebliche Informationssystem-Entwicklung



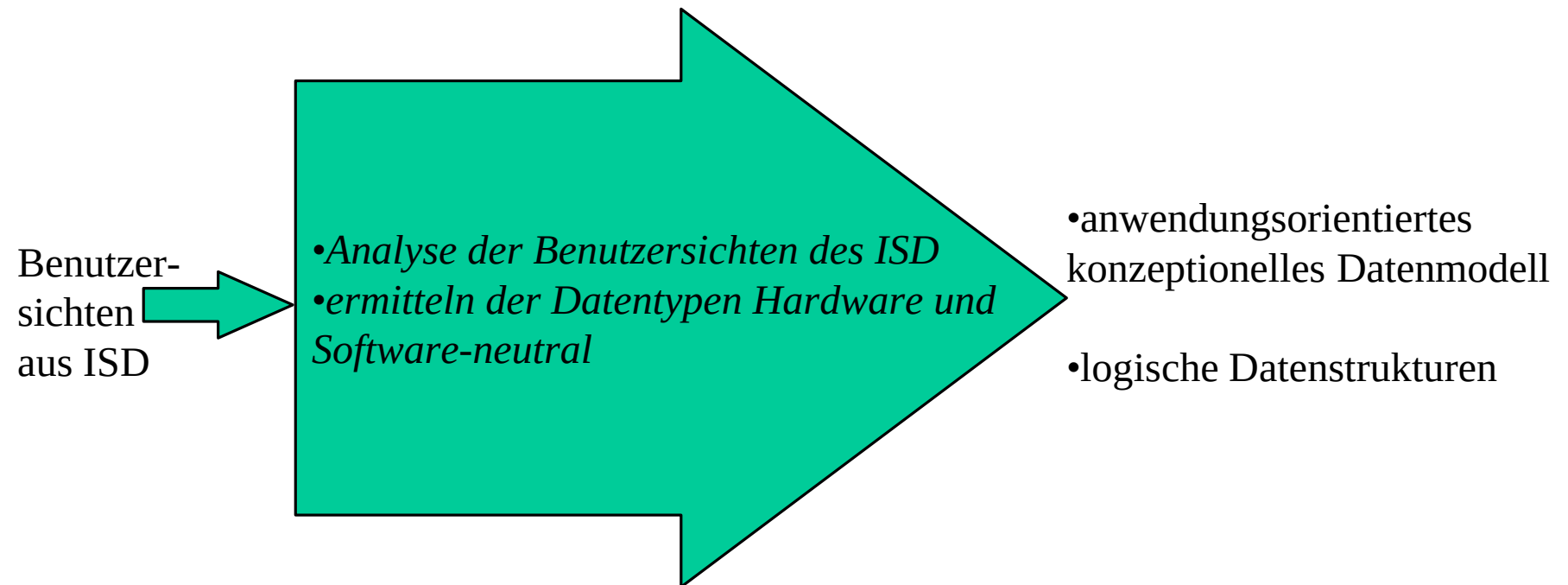
Objektsystem-Design (OSD)



Informationssystem-Design (ISD)



Konzeptionelles DB-Design (KDBD)



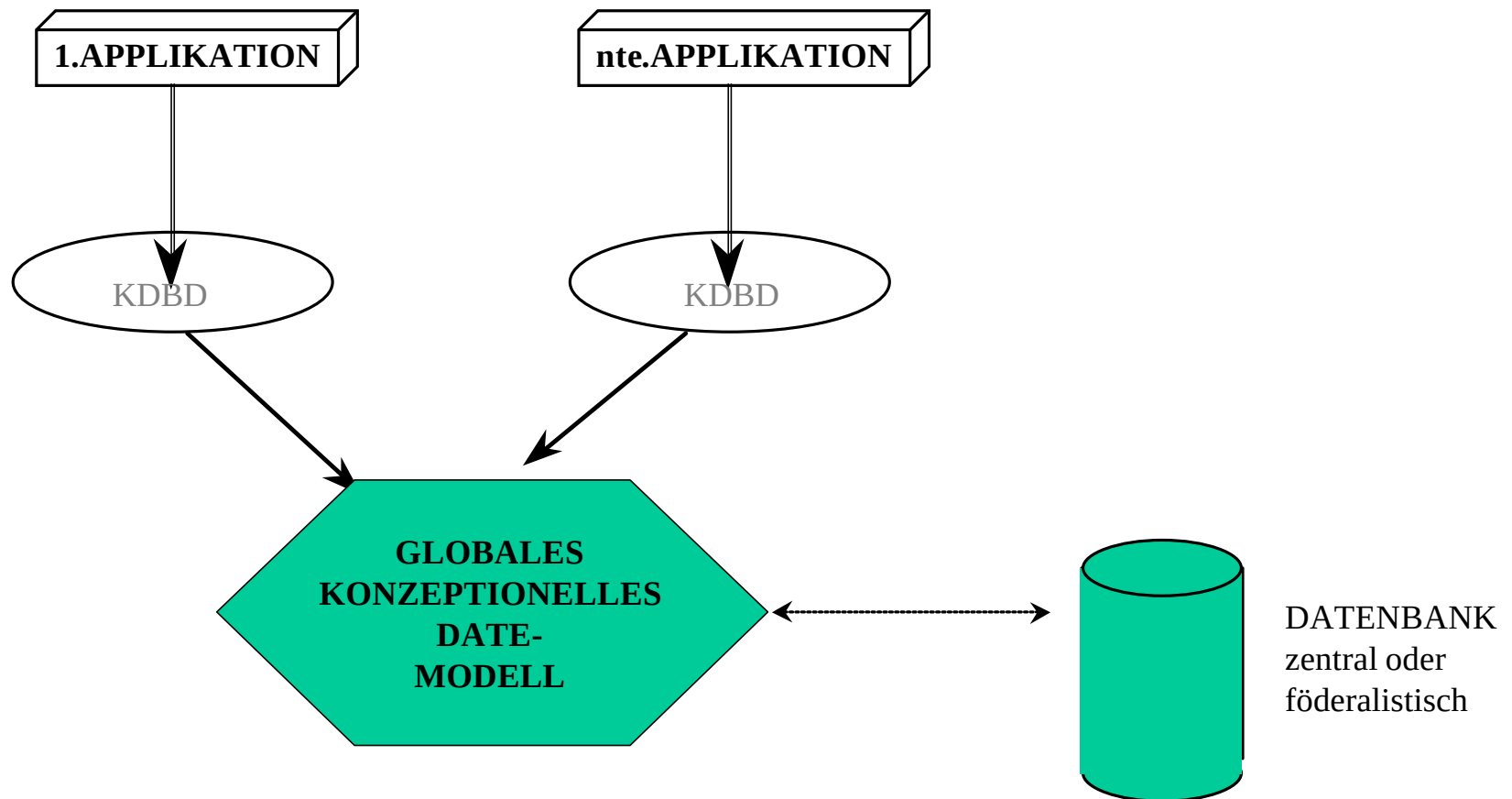
Prozessdesign (PD) und Dialogdesign (DD)



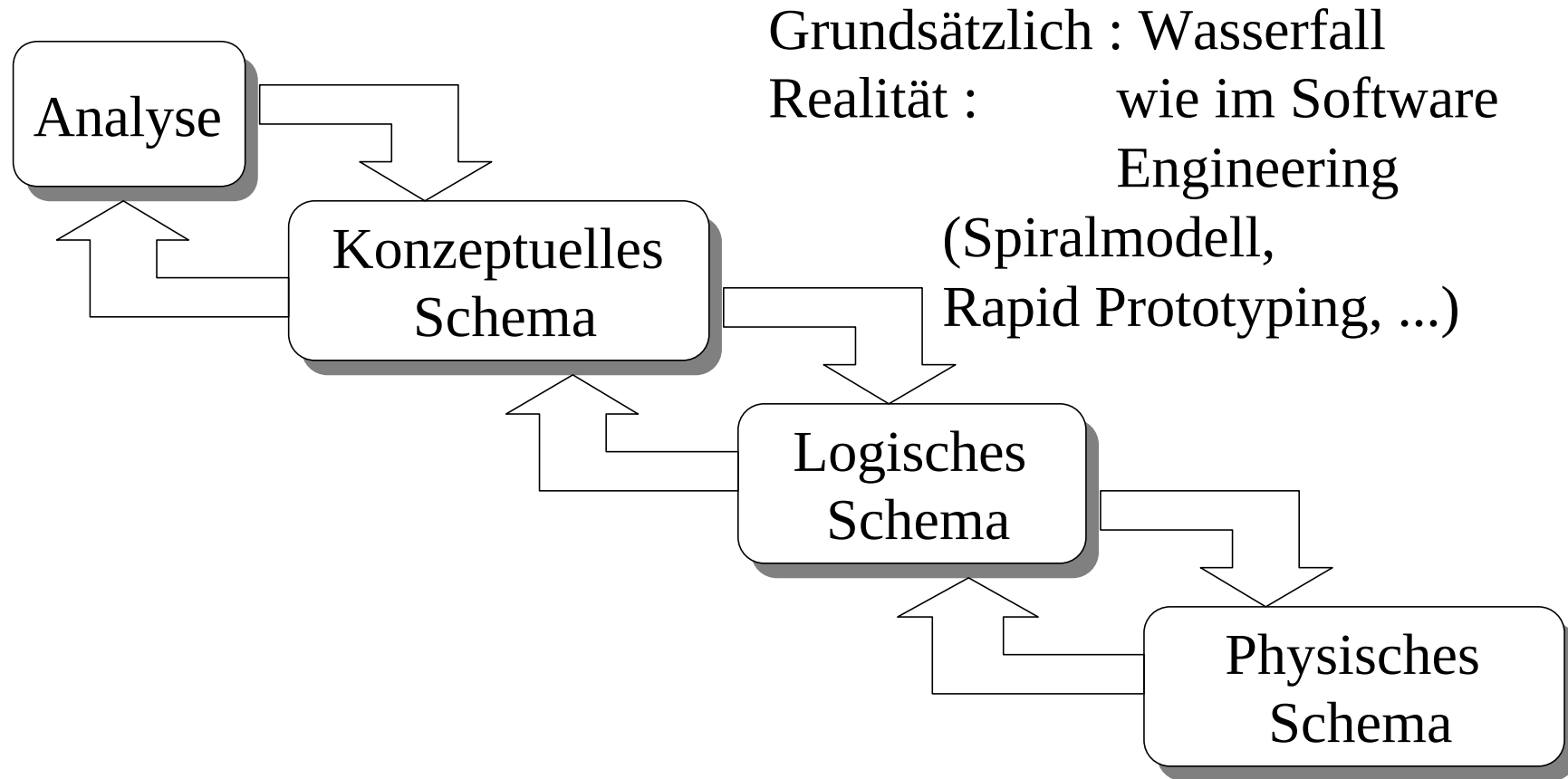
- PD : programmneutrale Beschreibung der Prozesslogik
- DD : effiziente, ergonomische Mensch-Maschine-Kommunikation



Praxis : Einbettung von Standardsoftware “Prespecified Computing”



Schemata erstellen



Praktische Vorgehensweise

- Realitätsbeobachtung
 - Benutzersichtenanalyse
 - Datenbestandanalyse
 - Interviews
 - Formularanalyse : Spezialfall der Benutzersichten
-
- Expertenbefragung
 - Analyse des „Best of the Class“
 - Analyse kommerzieller Systeme
 - Literaturstudium der Domain-Bibel

Formularanalyse

- Analysieren Sie ein Standard-Formular in Hinblick auf:
 - Entitäten, Attribute, Relationen
- Normalisieren Sie das resultierende Modell
- Beispiel : die Auftragsbestätigung
 - Sie sollten erkennen
 - Eine Offerte ist
 - fast ein Auftrag (aus Datensicht), Auftragsbestätigung
 - fast ein Rüstschein, Lieferschein
 - fast eine Rechnung
 - => Abstraktion : Geschäftsfall
 - ein Kunde und ein Lieferant sind aus Datensicht fast identisch
 - => Abstraktion : Geschäftspartner