

# Strukturierte Analyse

## Grundlegendes über Datenflussdiagramme

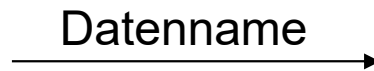
Stand 25.10.2017

## Datenflussdiagramm (DFD)

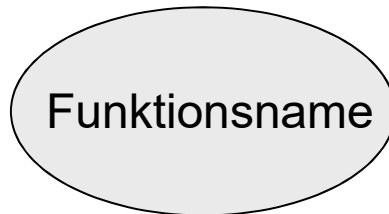
Ein **Datenflussdiagramm** (DFD, *data flow diagram*) beschreibt

- die Wege von **Daten** bzw. **Informationen** zwischen
  - **Funktionen**,
  - **Speichern** und
  - **Schnittstellen**
- die **Transformation der Daten** bzw. **Informationen** durch
  - **Funktionen**

## DFD-Symbole



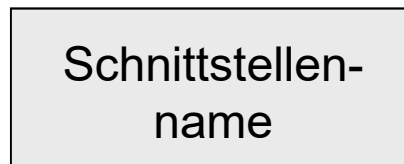
**Datenfluss**,  
(Kanäle, in denen Informationseinheiten fließen)



**Funktion** bzw. **Prozess** oder **Transformation**



**Datenspeicher**



**Schnittstelle** zur Umwelt (Terminator)

## DFD

### Grundidee beim Erstellen eines Datenflussdiagramms:

#### Vorstellung:

- das zu entwickelnde System laufe bereits.
- Initialisierung und Terminierung vorerst uninteressant
- Konzentration darauf, welche Daten von wo nach wo durch das System fließen

# Kontext-Diagramm

## Kontext-Diagramm

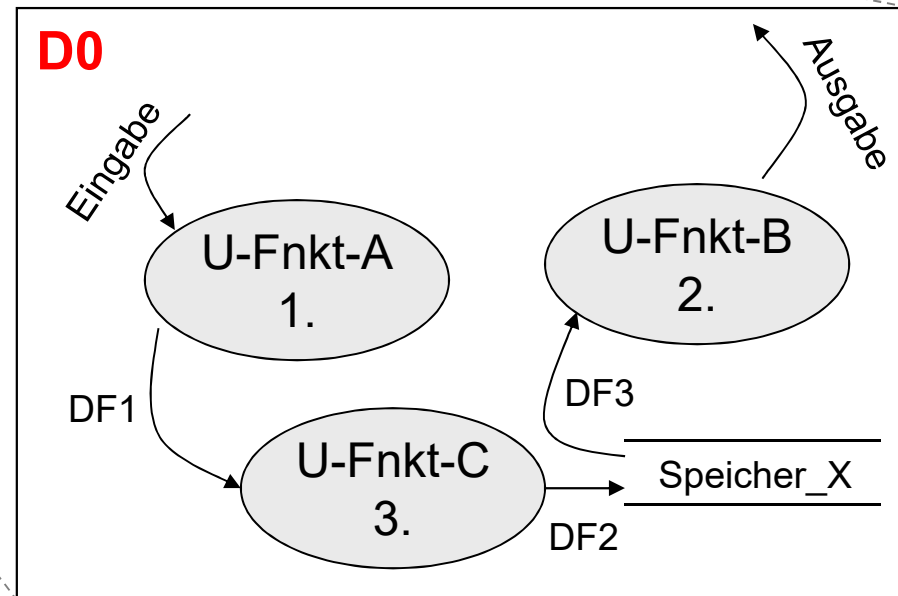
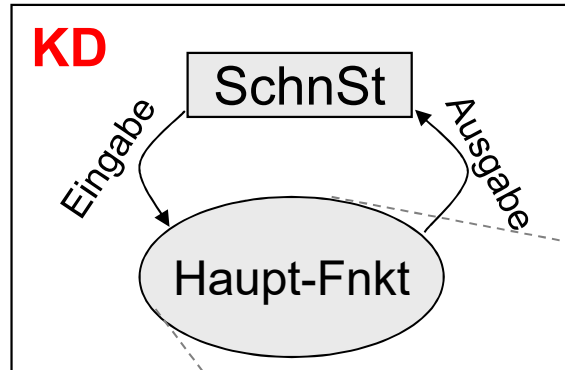
- bildet die **oberste Ebene** der Datenflussdiagramm-Hierarchie
- beschreibt die **Schnittstellen** des Gesamtsystems zur Außenwelt
- In einem Kontext-Diagramm wird das betrachtete System durch einen einzigen **Prozess** dargestellt. Dieser Prozess erhält noch keine Prozessnummer
- In der nächsten Stufe wird nun das zu betrachtende System weiter verfeinert ([Diagramm D0](#))
- Ein Kontextdiagramm enthält mindestens eine Schnittstelle

## Diagramm D0

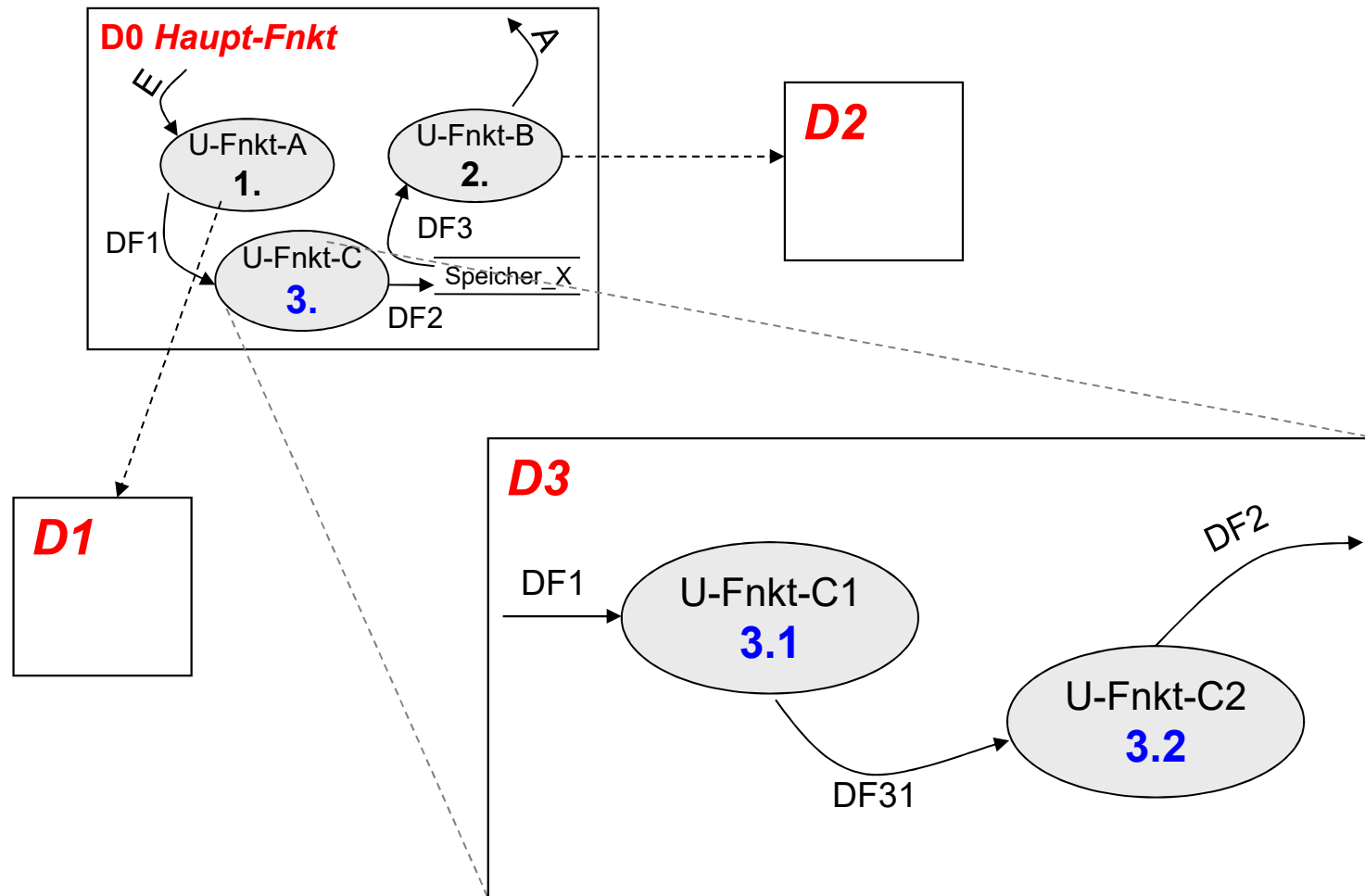
### Diagramm D0

- bildet die **zweite** Ebene der Datenflussdiagramm-Hierarchie
- erhält den Namen der Funktion des Kontextdiagramms
- hier werden nun die einzelnen Teilfunktionen definiert und durchnummeriert
- Jede Funktion einer weiteren Verfeinerung erhält die Basisnummerierung des übergeordneten Diagramms sowie eine weitere Unternummerierung

## Diagramm-Nummerierung



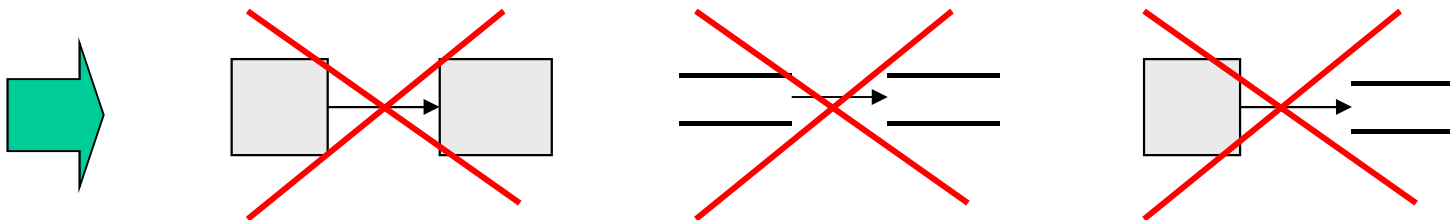
## Diagramm-Nummerierung (2)





## Regeln beim Zeichnen der Diagramme

- Jede **Schnittstelle** ist nur einmal vorhanden.  
**Ausnahme:**  
Wird dadurch das Diagramm unübersichtlich, dann kann eine Schnittstelle auch mehrfach gezeichnet werden
- Zwischen zwei **Schnittstellen** oder zwei **Speichern** und zwischen **Speichern** und **Schnittstellen** gibt es **keine direkten Datenflüsse!**



- Jeder **Datenfluss** hat einen Namen  
**Ausnahme:**  
Datenflüsse, die zu **Speichern** führen oder dort beginnen und keinen Namen haben, transportieren die **gesamten** gespeicherten Daten

## Semantische Regeln

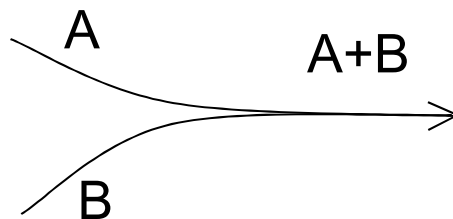
- Das DFD beschreibt den **Datenfluss**, **nicht den Kontrollfluss**. Daher enthält es weder Entscheidungen noch Schleifen
- **Datenerhaltungsprinzip**: jede Transformation kann nur die Ausgaben erzeugen, für die sie relevante Eingaben erhält
- Ein **Datenflussname** besteht aus einem Substantiv oder einem Adjektiv und einem Substantiv. Datenflussnamen enthalten niemals Verben
- Die Datenflussnamen sind so zu wählen, dass sie nicht nur die Daten beschreiben, die fließen, sondern etwas darüber aussagen, was über die Daten bekannt ist
- **Funktionsnamen** repräsentieren **Aktionen**

## Semantische Regeln (2)

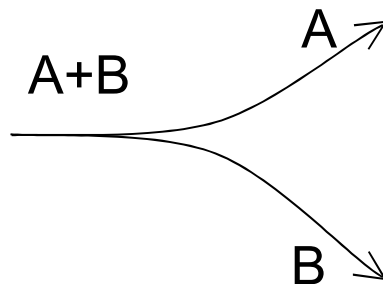
- **Verfeinerung:**
  - jede Transformation ergibt ein eigenes Diagramm
  - Die Ein-/Ausgaben des übergeordneten Diagramms müssen mit den Ein-/Ausgaben des untergeordneten Diagramms übereinstimmen
  - Datenspeicher, die von mehreren Transformationen benötigt werden, werden i.d.R. auf der obersten Ebene beschrieben

## Zusammenführen und Aufspalten von Datenflüssen

Verwendung: zur Reduktion von Pfeilen innerhalb eines Diagramms und über Diagrammebenen hinaus (Übersichtlichkeit).



Zusammenführung zweier Datenflüsse



Aufspaltung eines Datenflusses in zwei einzelne Datenflüsse. Dabei müssen alle (Teil-) Datenflüsse benannt werden!

# Beispiele

1. **Klausuranmeldung**
2. **Adressverwaltung**
3. **Bibliotheksverwaltung**

# 1. Beispiel Klausuranmeldung

(siehe Aufgabensammlung)

Eine Anwendung zur Unterstützung der Klausuranmeldungen für Studenten soll erstellt werden.

Dazu sollen zunächst zur Mitte des Vorsemesters die für das nächste Semester geplanten Klausuren in das System übernommen werden. Diese Eingaben (incl. Raum der Klausur, Länge der Klausur, Klausurtag, Aufsichtsführende, Prüfer etc.) werden vom Vorsitzenden des Prüfungsausschusses durchgeführt.

Wenn alle Klausuren erfasst worden sind, erzeugt das System zwei Prüfungspläne:

1. Der Prüfungsplan für die Professoren und Mitarbeiter beinhaltet die Aufsichtsführenden der jeweiligen Klausur und den jeweiligen Erst- und Zweitprüfer. Er wird allen Professoren und Mitarbeitern zugestellt.
2. Den Prüfungsplan für die Studenten ohne diese Informationen. Dieser Prüfungsplan wird den Studenten zugestellt.

## Beispiel Klausuranmeldung

Studenten können sich sodann an speziellen Terminals für ihre Klausuren anmelden.

Das System überprüft, ob die Studenten für die Klausur zugelassen sind und ob es die Klausur überhaupt gibt. Zur Überprüfung der Vorleistung wird eine bereits existierende Vorleistungsdatei genutzt.

Ist der Student für die Klausur zugelassen, erhält er eine Bestätigung und wird in die Teilnehmerdatei für die Klausur übernommen. Ansonsten erhält er eine Ablehnung mit Begründung.

Nach Ablauf des Anmeldezeitraums erhalten die jeweiligen Erstprüfer eine Liste mit den für ihre Klausur zugelassenen Studenten.

Entwickeln Sie ein **Datenflussdiagramm** für diesen Vorgang.

## Beispiel Klausuranmeldung

### Vorgehensweise zur Erstellung des Kontextdiagramms:

Ermitteln der Schnittstellen und Datenflüsse

- Schnittstellen:**
- Professoren
  - Studenten
  - Vorsitzender des Prüfungsausschusses

**Datenflüsse:**

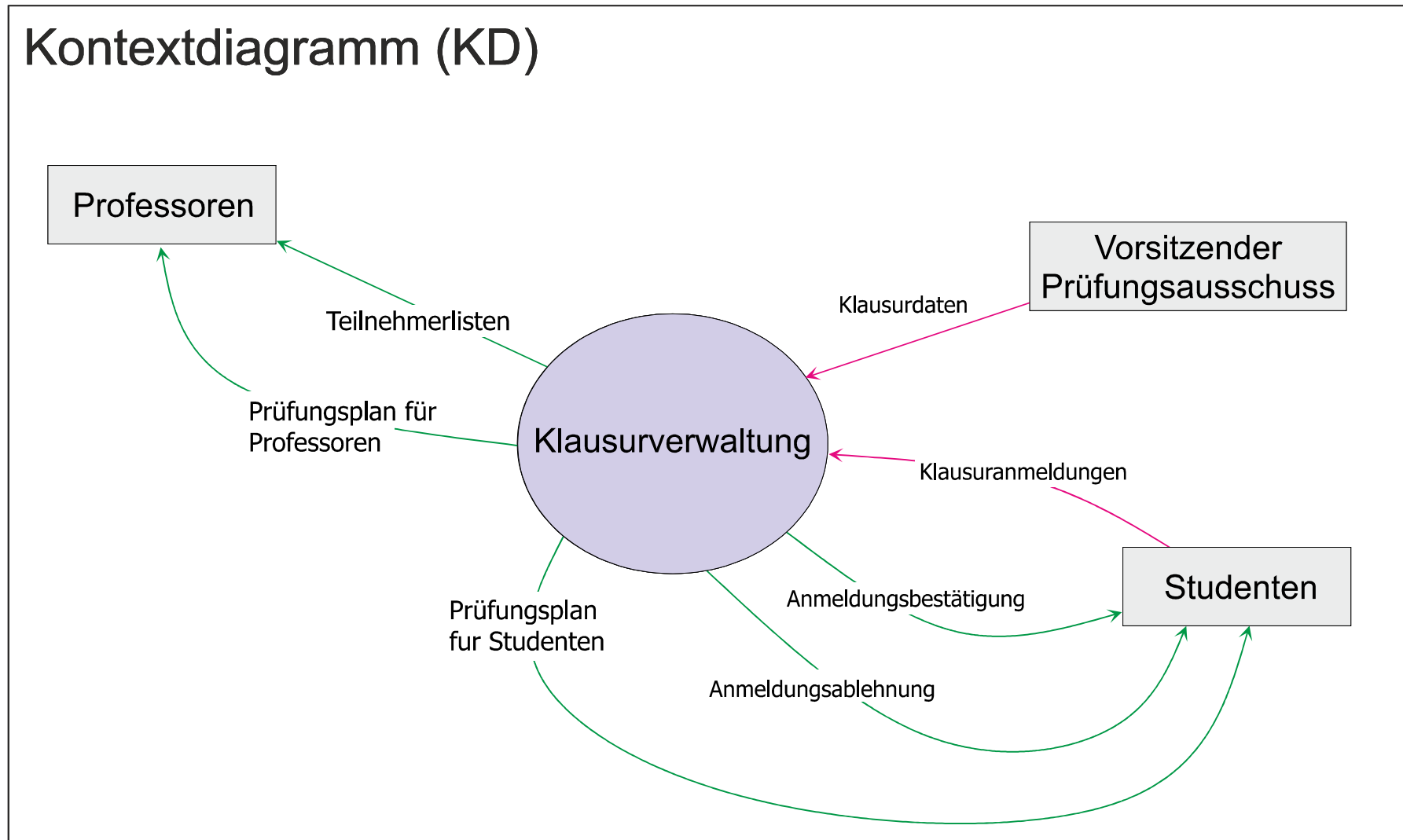
- Eingaben:**
- Klausurdaten
  - Klausuranmeldungen

- Ausgaben:**
- Prüfungsplan für Professoren
  - Prüfungsplan für Studenten
  - Anmeldungsbestätigung
  - Anmeldungsablehnung
  - Teilnehmerlisten



## Beispiel Klausuranmeldung

### Kontextdiagramm (KD)



# Beispiel Klausuranmeldung

## Vorgehensweise zur Erstellung des Diagramms D0:

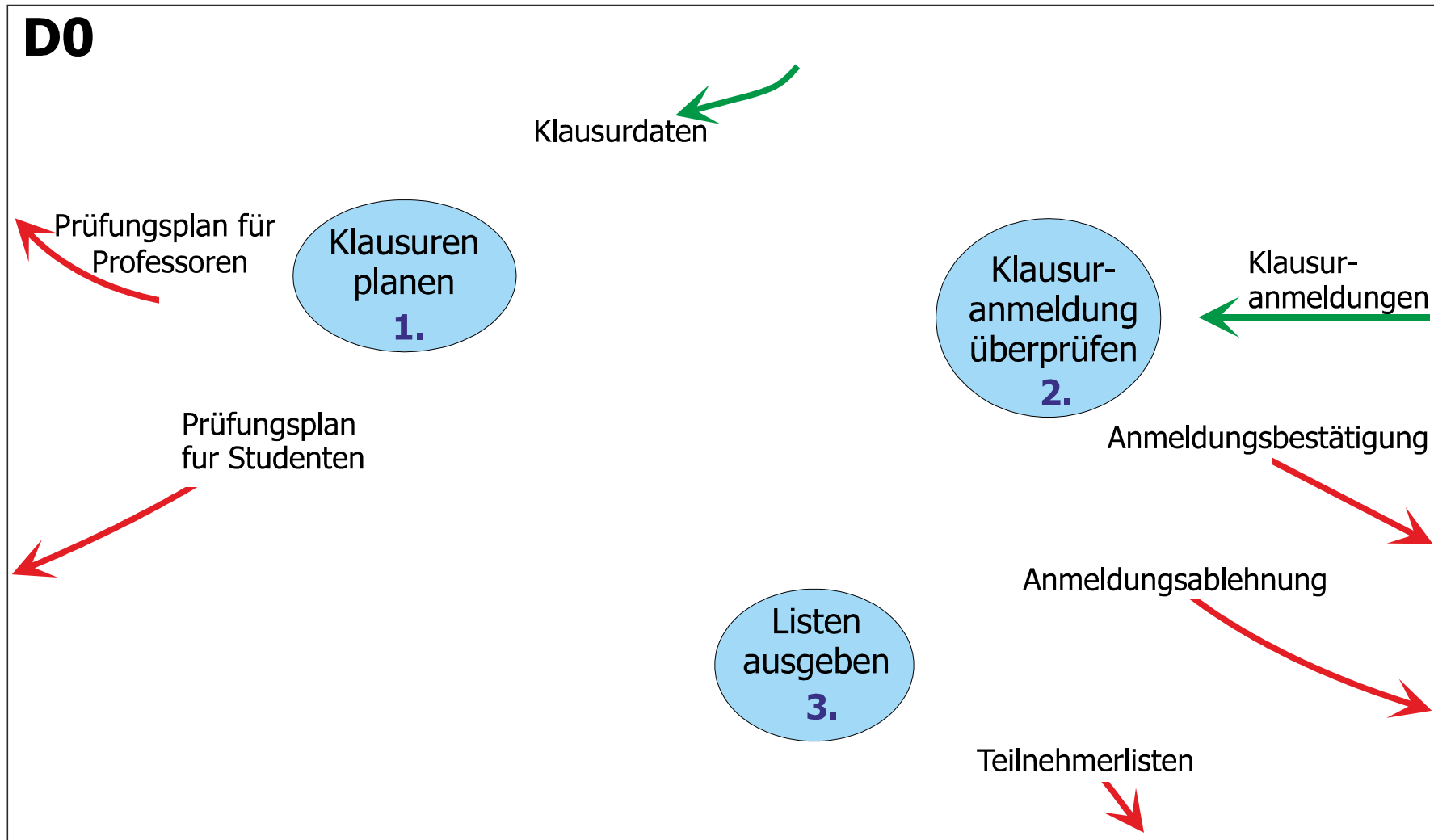
### 1. Ermitteln der einzelnen Hauptfunktionen

- Klausuren planen
- Klausuranmeldung überprüfen
- Listen ausgeben

### 2. Einzeichnen der Ein- und Ausgabe-Datenflüsse des Kontextdiagramms

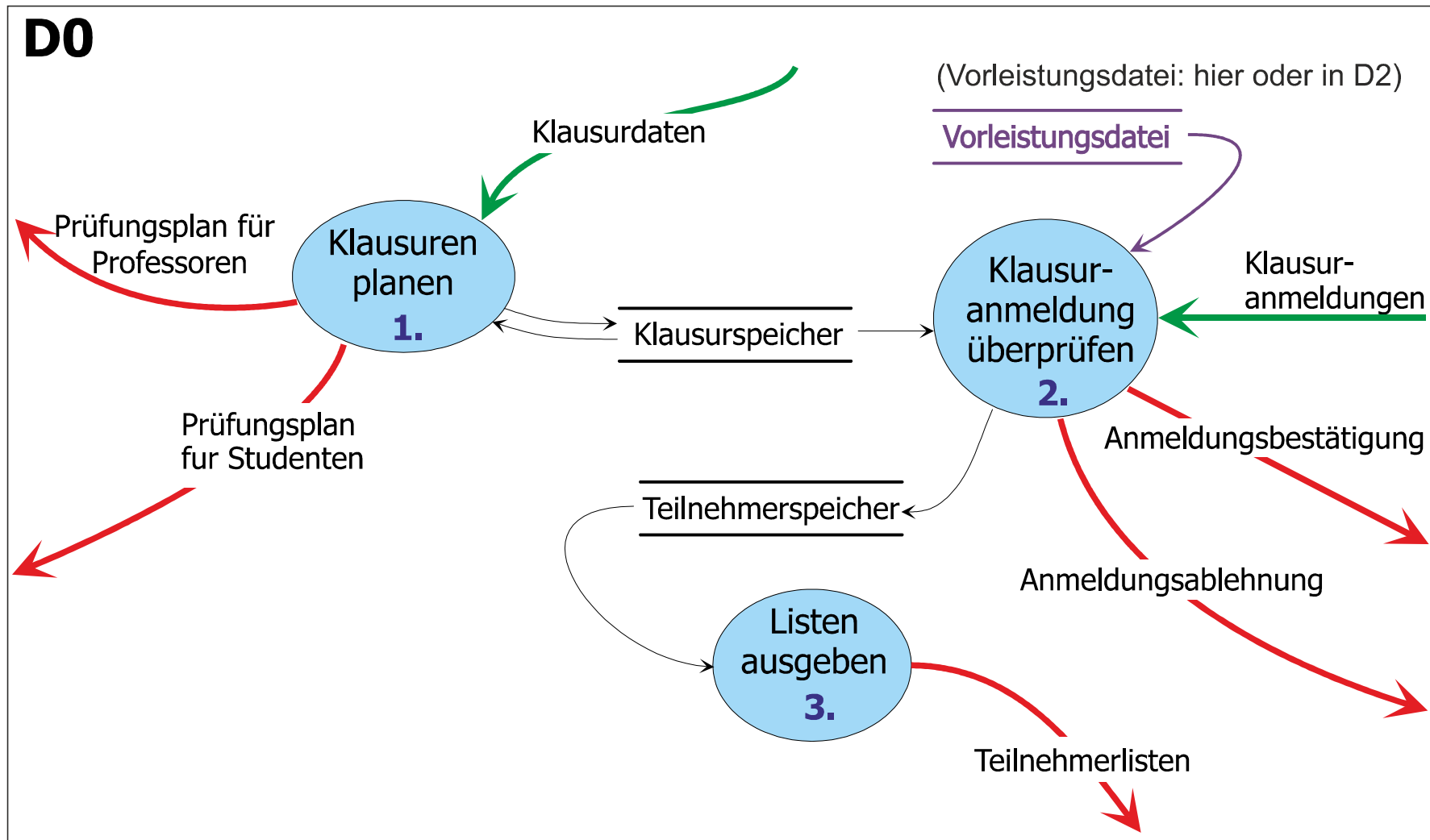
## Beispiel Klausuranmeldung (Erstellung von D0)

### 2. Einzeichnen der Ein- und Ausgabe-Datenflüsse



## Beispiel Klausuranmeldung (Erstellung von D0)

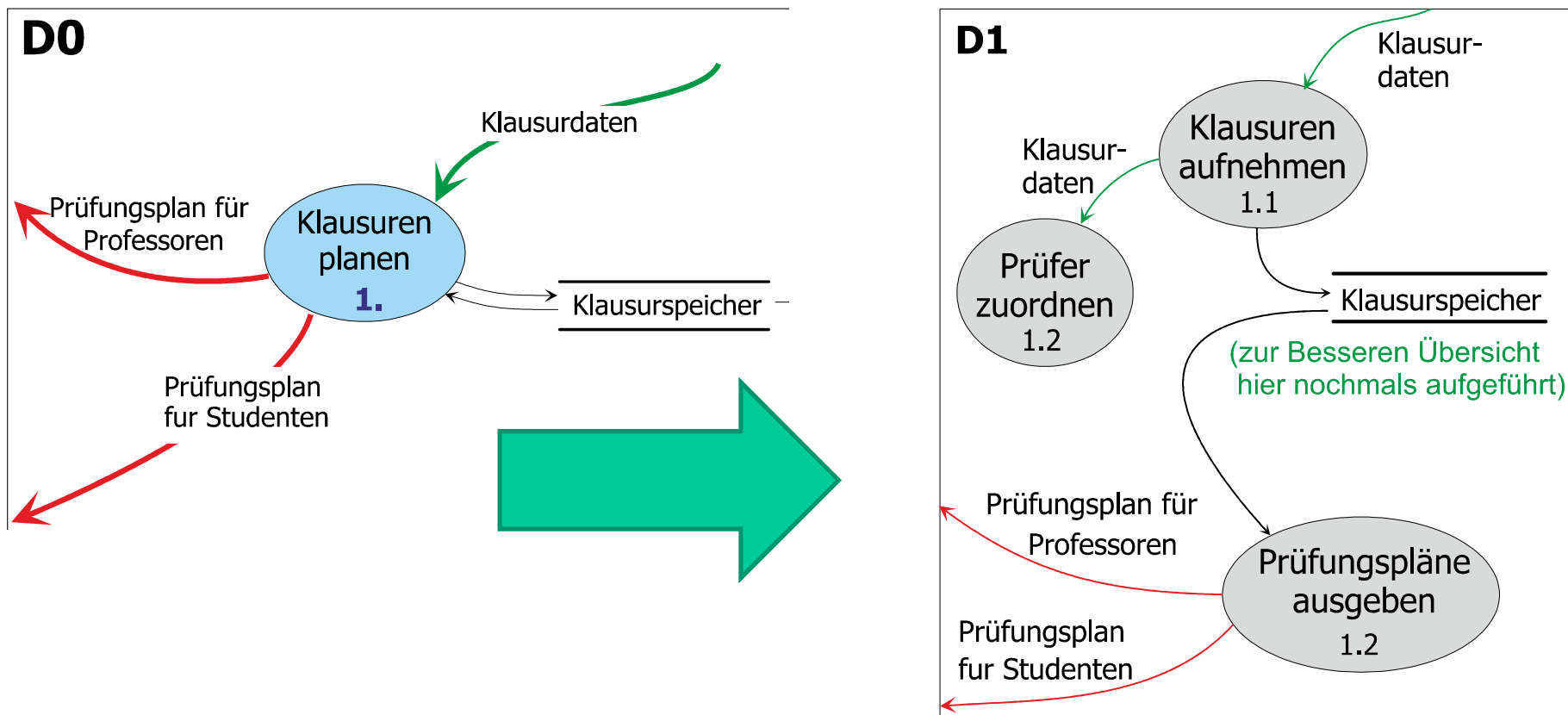
### 3. Vervollständigen mit internen Datenflüssen und Speichern



## Beispiel Klausuranmeldung

### Vorgehensweise zur Erstellung der Unterdiagramme D1, D2, ... , Dx.y:

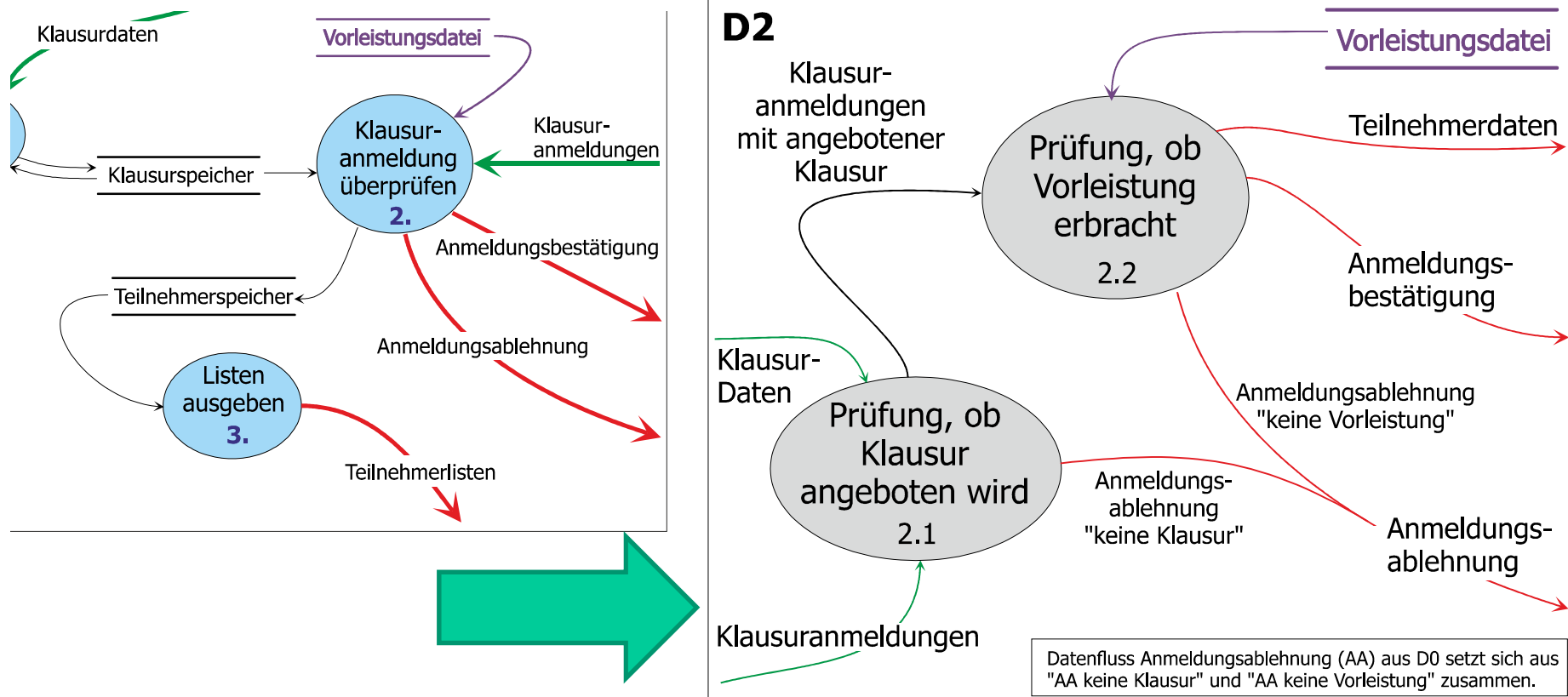
Analog zur Erstellung von D0 bezogen auf das übergeordnete Diagramm



## Beispiel Klausuranmeldung

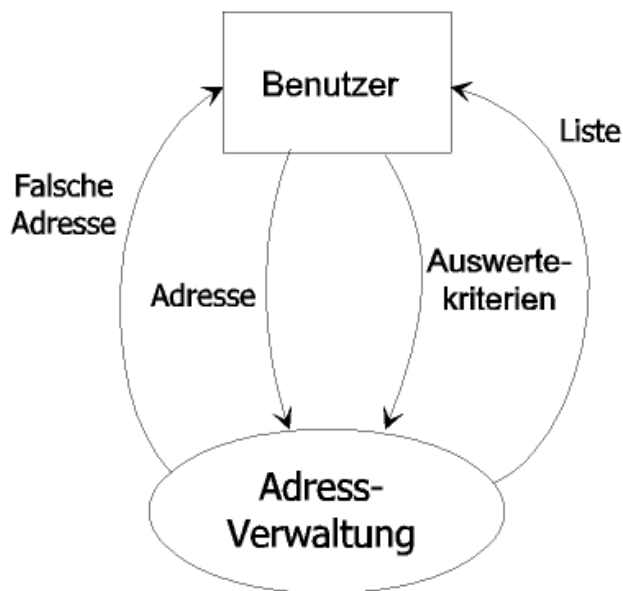
### Vorgehensweise zur Erstellung der Unterdiagramme D1, D2, ... , Dx.y:

Analog zur Erstellung von D0 bezogen auf das übergeordnete Diagramm

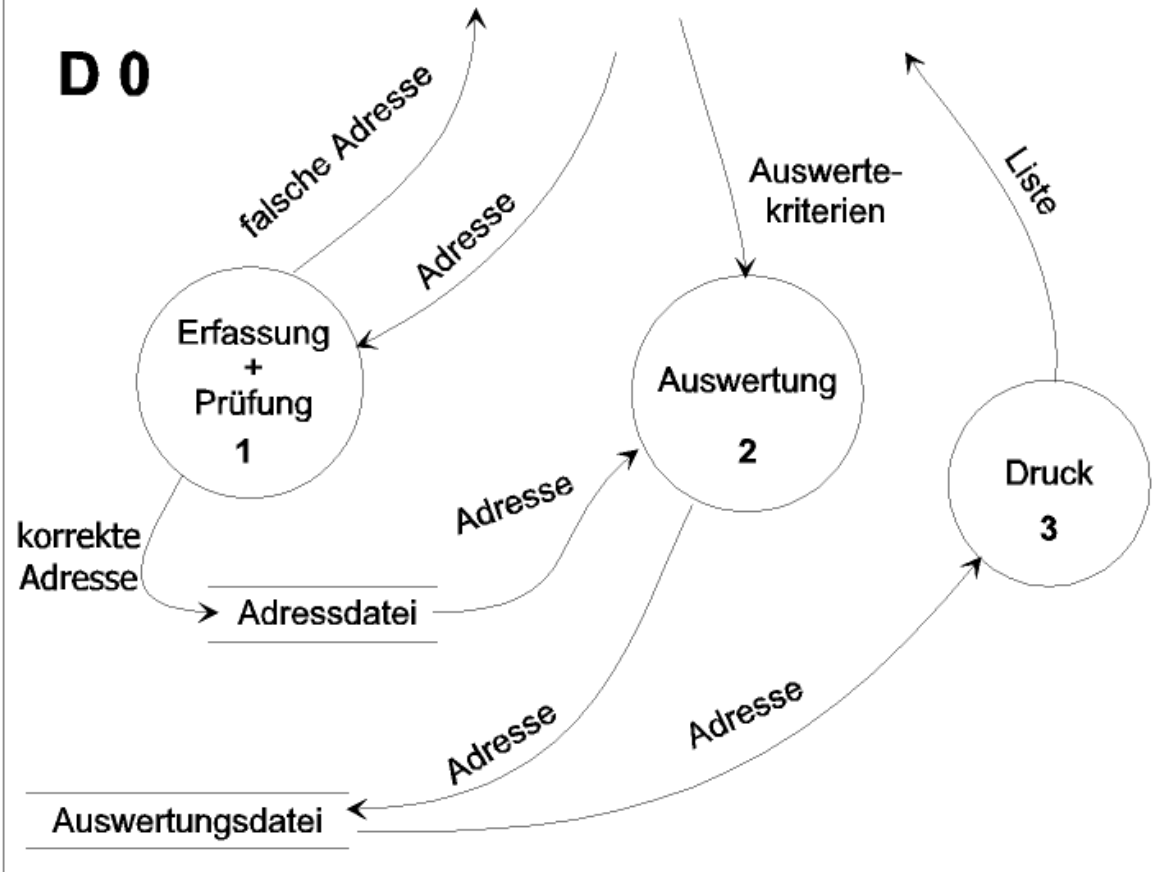


## 2. Beispiel Adressverwaltung

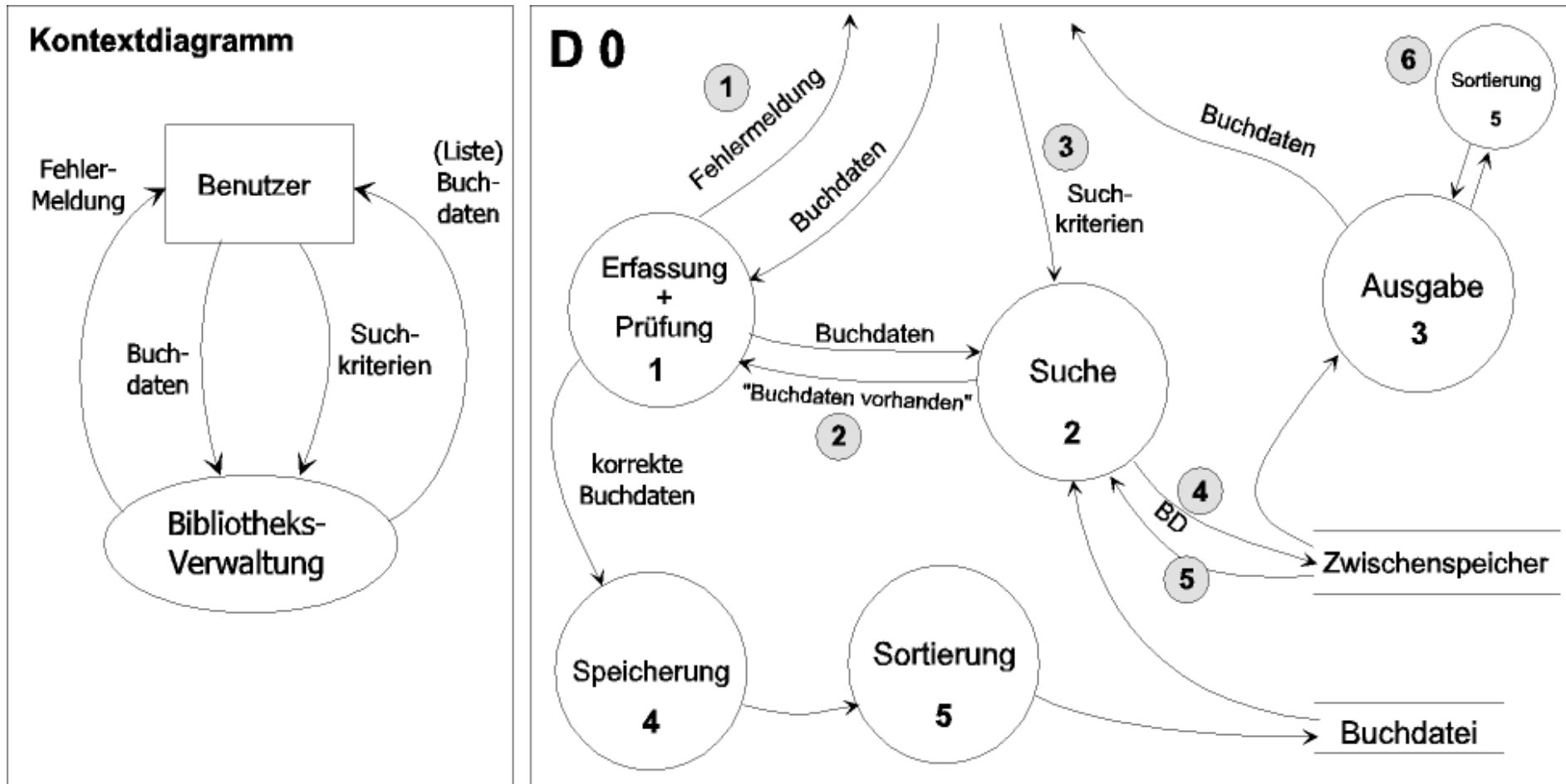
**Kontextdiagramm**



**D 0**



### 3. Beispiel Bibliotheksverwaltung



(Anmerkungen zu den Punkten ① - ⑥ siehe nächst Seite)



## Beispiele (Bibliotheksverwaltung (2))

1. Eine Fehlermeldung kann auch an die Ausgabefunktion weitergeleitet werden. Dies ist hier aus Übersichtlichkeitsgründen nicht erforderlich, wäre allerdings konsequenter.
2. Dasselbe gilt auch für diesen Punkt.
3. Die Eingabe der Suchkriterien kann ebenso über die Funktion *Erfassung+Prüfung* an die Funktion *Suche* weitergeleitet werden. Dies bedeutet, dass generell alle Eingaben über eine Funktion *Eingabe* abgehandelt werden kann, was für die Entwicklung der Benutzeroberfläche hilfreicher sein könnte.
4. Hier ist der Inhalt des Datenflusses das Suchergebnis (eine gefundene Menge von Buchdaten/Büchern).
5. Dieser Datenfluss entsteht, wenn eine Verfeinerung der Suche möglich sein soll.
6. Die Sortierung der Buchdaten kann hier auf dieser Ebene oder auch innerhalb oder auch vor der Funktion *Ausgabe* erfolgen.

# Strukturierte Analyse

nach Tom DeMarco

Bei der Strukturierten Analyse werden **drei** Darstellungsmittel eingesetzt:

1. **Datenflussdiagramm**      graphische Darstellung der Datenflüsse und Funktionen eines Systems
2. **Data Dictionary**      Erläuterung aller Datendefinitionen in einem Datenlexikon
3. **Prozess-Spezifikation**      (Mini-) Spezifikation, eine umgangssprachliche Beschreibung der Aufgabe der Funktionen

# Strukturierte Analyse

## Prozessspezifikation (Minispezifikation)

1. Umfang maximal 1 Seite. Wenn der Platz nicht ausreicht, muss die Transformation weiter zerlegt werden
2. Beschreibung der Transformationen in der Sprache des Anwenders
3. programmiertechnische Besonderheiten sind **nicht** zulässig
4. Beschreibungsmittel:

Je nach Transformation z.B.

- Nassi-Shneiderman-Diagramme,
- PAP,
- Pseudocode sowie
- Entscheidungstabellen

# Strukturierte Analyse

## Durchführung:

- Die Abstimmung der Arbeitsergebnisse ( DFD, Datenlexikon, Minispezifikation ) erfolgt nach jedem Schritt mit dem Benutzer (Anwender, Auftraggeber)
- Die sich daraus ergebenden Vorteile sind:
  - das spätere System entspricht den Anforderungen der Benutzer
  - Ergebnis ist eine umfassende, aber übersichtliche Spezifikation
  - Änderungen können leicht durchgeführt werden
  - Der Umfang der Dokumentationen ist wesentlich kleiner als herkömmliche Beschreibungen