



Speed Up Development

40 BUILDING ON
YEARS OF
COMMITMENT

Mapping Generierung mit ODI
Martin Wichert
November 2015

© CGI Group Inc.

CGI

Experience the commitment®

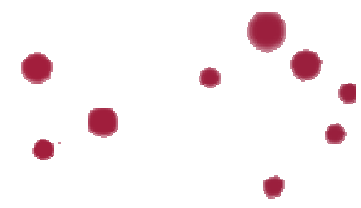
Agenda

- Vor- und Nachteile von automatisierter Mapping Generierung
- Warum sich gerade Data Vault für die Generierung eignet
- Namenskonventionen
- Vorgehensweise



Speed up Development

Vor- und Nachteile



40
YEARS

CGI

Experience the commitment®

Vor- und Nachteile

Nachteile:

- Aufwand den Generator zu erstellen
- Aufwand den Generator zu testen
- Aufwand um dem Generator die nötigen Informationen zu liefern



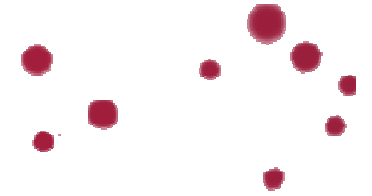
Vor- und Nachteile

Vorteile:

- Schnelle Generierung von großen Mengen an Mappings
- Geringerer Testaufwand für die einzelnen Mappings
- Konstantere Qualität der Mappings
- Leichtes hinzufügen von zusätzlichen Mappings
- Schnelles anpassen bei Änderungen an allen Mappings



Speed up Development Data Vault und Generierung



40
YEARS

CGI

Experience the commitment®

Data Vault und Generierung

- Nur 4 Typen von Mappings
 - Hubs
 - Links
 - Hub-Satelliten
 - Link-Satelliten
- Keine Business Logik
- Einheitliche Strukturen bei Metadaten



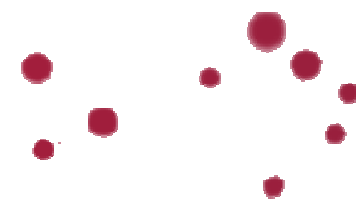
Beispiel eines Satelliten

- Metadaten
 - MD_*
- Zugehöriger Hub ist EVENT_REP_H
 - Künstlicher Schlüssel EVENT_REP_H_ID
- Primärschlüssel des Satelliten
 - EVENT_REP_H_ID
 - BEG_DATE
 - MD_LOAD_ID

EVENT_REP_501	
PK	r EVENT_REP_H_ID
	r EVENT_REP_SHK
	d INSERT_DATE
PK	d BEG_DATE
	d END_DATE
	n EVENT_FLAGS_ID
	n REF_STATUS_ID
	n STATUS_ID
	n ORGA_STRUCTURE
	...
	n ADD_CAUS_UNIT_S'
	d ADD_CAUS_UNIT_S'
	n MD_DWH_ID
	n MD_LOAD_ID
PK	d MD_LOAD_DATE
	v MD_LOAD_USER
	v MD_UPDATE_USER
	d MD_UPDATE_DATE
	v MD_SRC_SYSTEM



Speed up Development Namenskonventionen



40
YEARS

CGI

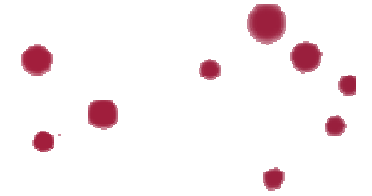
Experience the commitment®

Namenskonventionen

- Tabellen
 - Suffix: `_H`, `_Sxx`, `_L`, `_LSxx` für Hub Hub-Satellit, Link und Link-Satellit
 - Satelliten werden benannt wie die Parent-Tables
 - Data Vault Tabellen werden nach den Quelltabellen benannt
- Felder heißen im Stage und Core gleich
- Schlüssel
 - Business Keys im Hub heißen ID
 - Die künstlichen Schlüssel heißen wie die Hubs mit Suffix `_ID`
- Metadaten
 - Die Metadatenfelder heißen in allen Tabellen gleich



Speed up Development Vorgehensweise



40
YEARS

CGI

Experience the commitment®

Vorgehensweise

Übergabe an Generator:

- Stage Model
- Core Model
- Optional Whitelist
- Special Table Mapping
- Special Field Mapping



Vorgehensweise

Generator liest aus den ODI-Metadaten aus:

- Namen der Stage Tabellen
- Namen der Core Tabellen

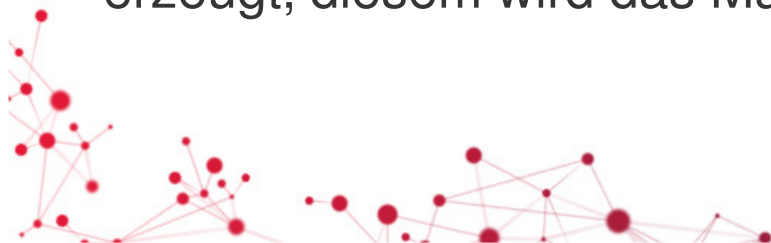
Generator Ordnet zu:

- Für jede Stage Tabelle wird geprüft, ob es passend benannte Core Tabellen gibt
- Für jede Stage Tabelle wird geprüft, ob es eine Zuordnung im Special Table Mapping gibt

Für jede gefundene Zuordnung wird ein leeres Mapping erstellt

```
log_map = new Mapping(map_name, mapping_folder)
```

Dann wird eine Prozedur aufgerufen, die den Inhalt des Mappings erzeugt, diesem wird das Mapping die Quelle und das Ziel übergeben



Vorgehensweise am Beispiel eines Satelliten

Erstellen der Data Stores

- Quelltable mit den Daten die Geladen werden sollen.
 - `def src = new DatastoreComponent(log_map, qds)`
- Zieltabelle in die die Daten geladen werden sollen
 - `def tgt = new DatastoreComponent(log_map, tds)`
- Abgleichquelle Die Zieltabelle als Datenquelle zum Abgleich damit Delta ermittelt werden kann
 - `def dtg = new DatastoreComponent(log_map, tds)`



Vorgehensweise am Beispiel eines Satelliten

EVENT_REP_S011	
PK	EVENT_REP_H_ID
R	EVENT_REP_SHK
V	REPORTING_PERSON
V	EXTERNALREF
D	INSERT_DATE
PK	BEG_DATE
D	END_DATE
N	EVENT_FLAGS_ID
N	REF_STATUS_ID
N	STATUS_ID
...	...

IRS_EVENT_REP	
N	ID
V	REPORTING_PERSON
V	EXTERNALREF
D	INSERT_DATE
D	BEG_DATE
D	END_DATE
N	EVENT_FLAGS_ID
N	REF_STATUS_ID
N	STATUS_ID
N	FINANCIAL_ID
...	...

EVENT_REP_S01	
PK	EVENT_REP_H_ID
R	EVENT_REP_SHK
V	REPORTING_PERSON
V	EXTERNALREF
D	INSERT_DATE
PK	BEG_DATE
D	END_DATE
N	EVENT_FLAGS_ID
N	REF_STATUS_ID
N	STATUS_ID
...	...



Vorgehensweise am Beispiel eines Satelliten

- Anlegen eines Aggregators

```
agg = new AggregateComponent(log_map, Name)
```

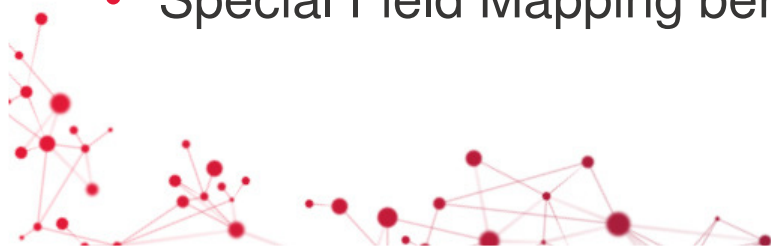
- Auslesen der Spalten im Ziel

```
def tgt_atts = tgt.getAttributes()
```

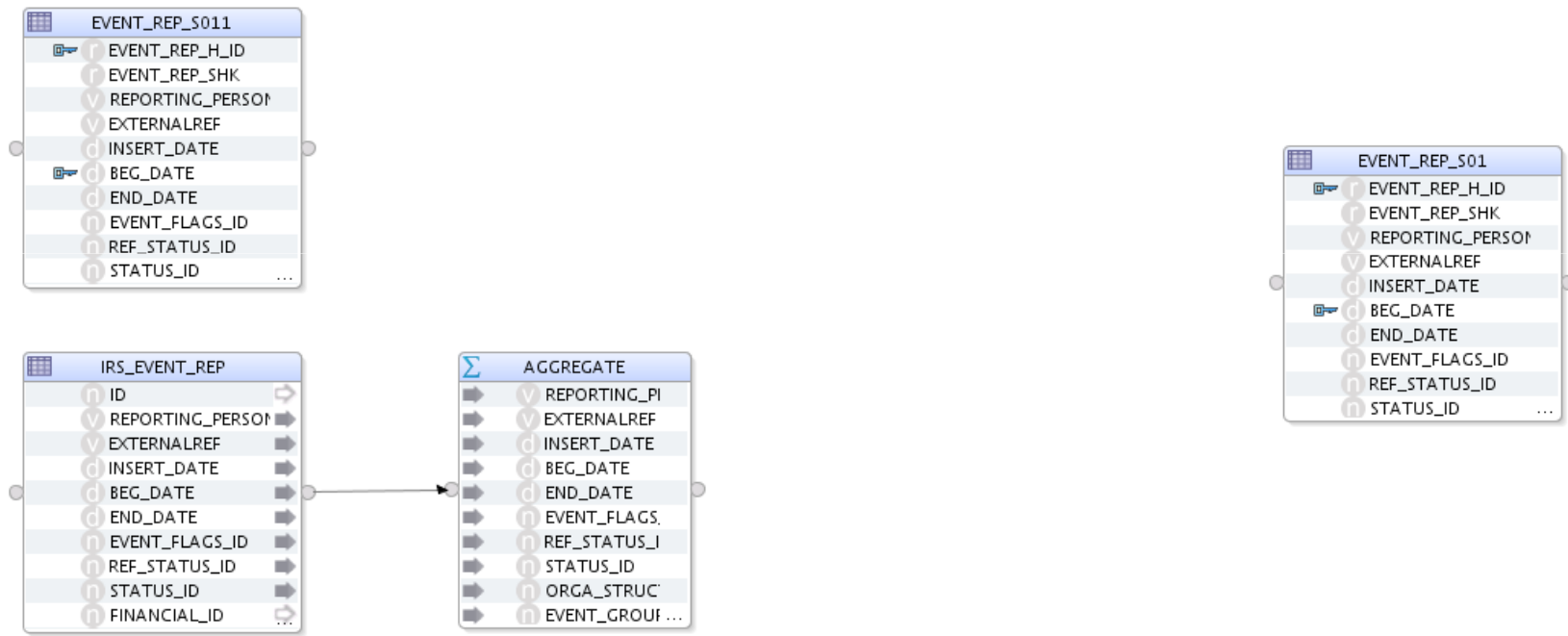
- Anlegen der Felder im Aggregator

```
tgt_atts.each{ attr ->  
agg.addAttribute(attr.name, Expr, null, null, null)  
}
```

- Special Field Mapping berücksichtigen



Vorgehensweise am Beispiel eines Satelliten



Vorgehensweise am Beispiel eines Satelliten

- Einfügen einer Expression Component

```
exp = new ExpressionComponent(log_map, Name)
```

- Nicht Metadatenfelder aus der Abgleichquelle in die Expression ziehen

```
atts = tgt_atts.findAll{!it.name.startsWith("MD_")}
```

```
atts.each{attr ->
```

```
exp.addExpression(attr.name, qual_attr, null, null, null)
```

- Den PK der Tabelle bestimmen

```
def primary_key = tds.getPrimaryKey()
```

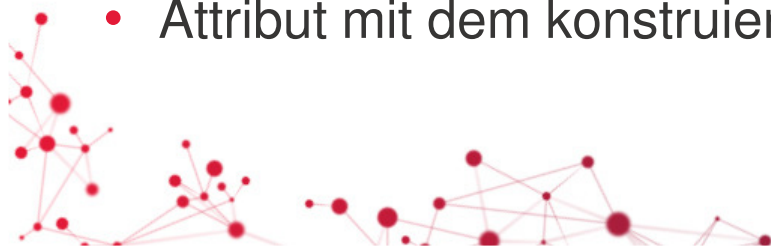
```
def primary_key_cols = primary_key.getColumns().name
```

- Rank konstruieren

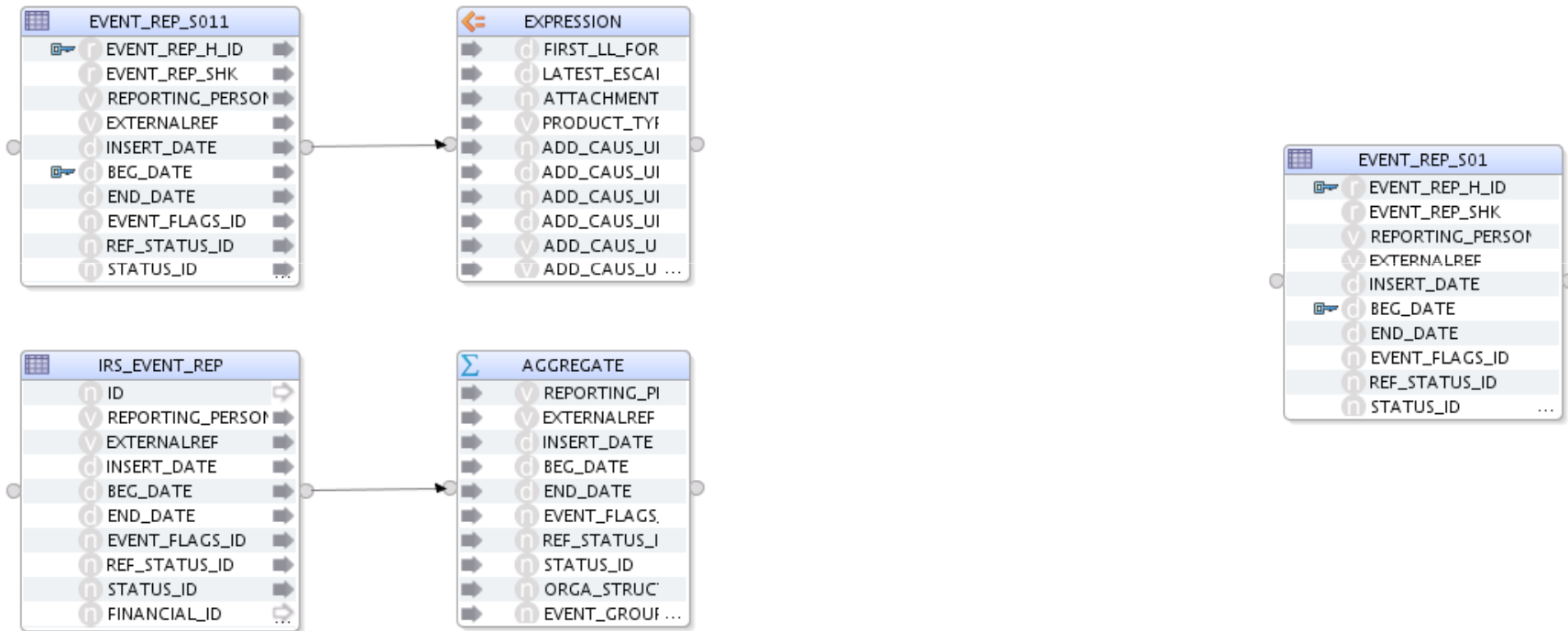
- Gruppiert nach PK ohne MD_LOAD_DATE

- Absteigend sortiert nach MD_LOAD_DATE

- Attribut mit dem konstruierten Rank in die Expression einfügen



Vorgehensweise am Beispiel eines Satelliten



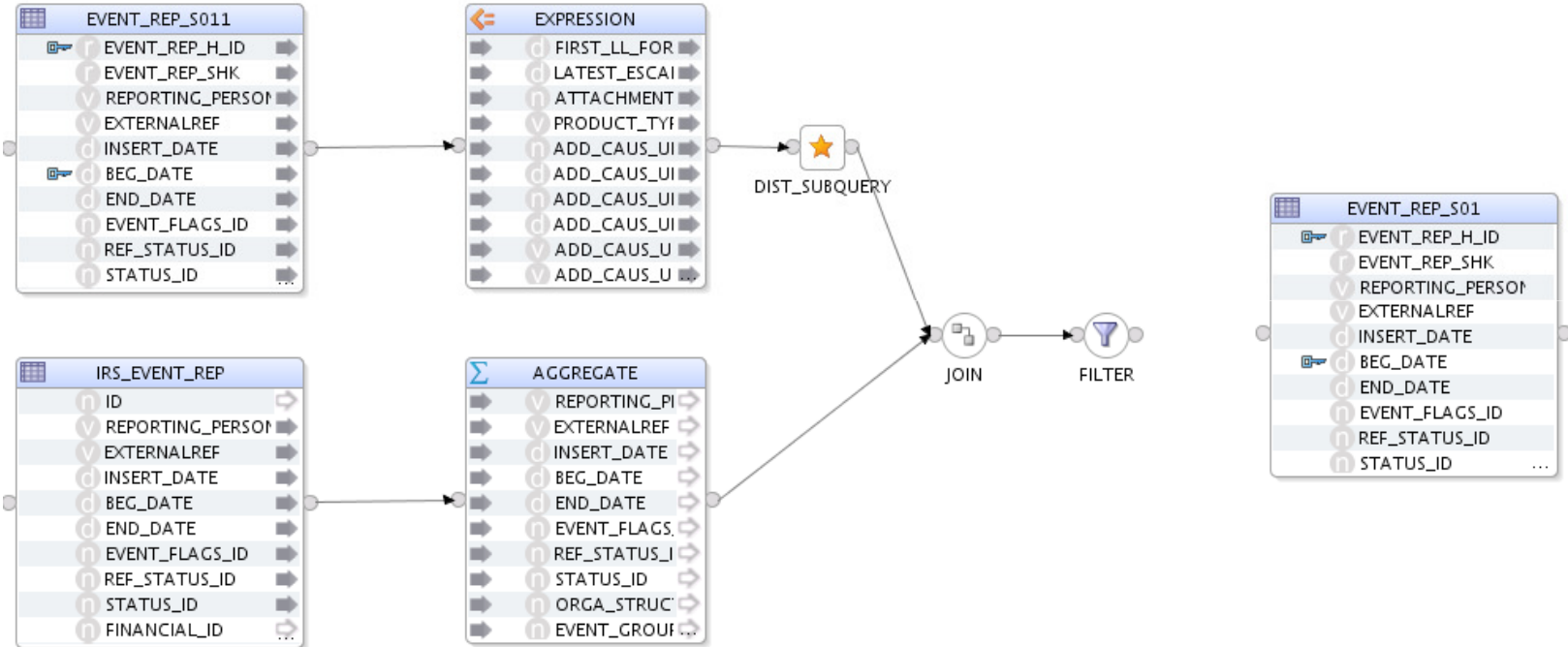
Vorgehensweise am Beispiel eines Satelliten

Vergleichen der Datenströme

- Distinct Component Einfügen
 - Alles Übernehmen um eine Subquery zu erzwingen
- Join Component einfügen
 - Joinbedingung PK ohne MD_LOAD_DATE
- Filter Component einfügen
- Filterbedingung bestimmen
 - Einzelattributsabgleich unter Berücksichtigung des Special Field Mappings
 - Rank = 1

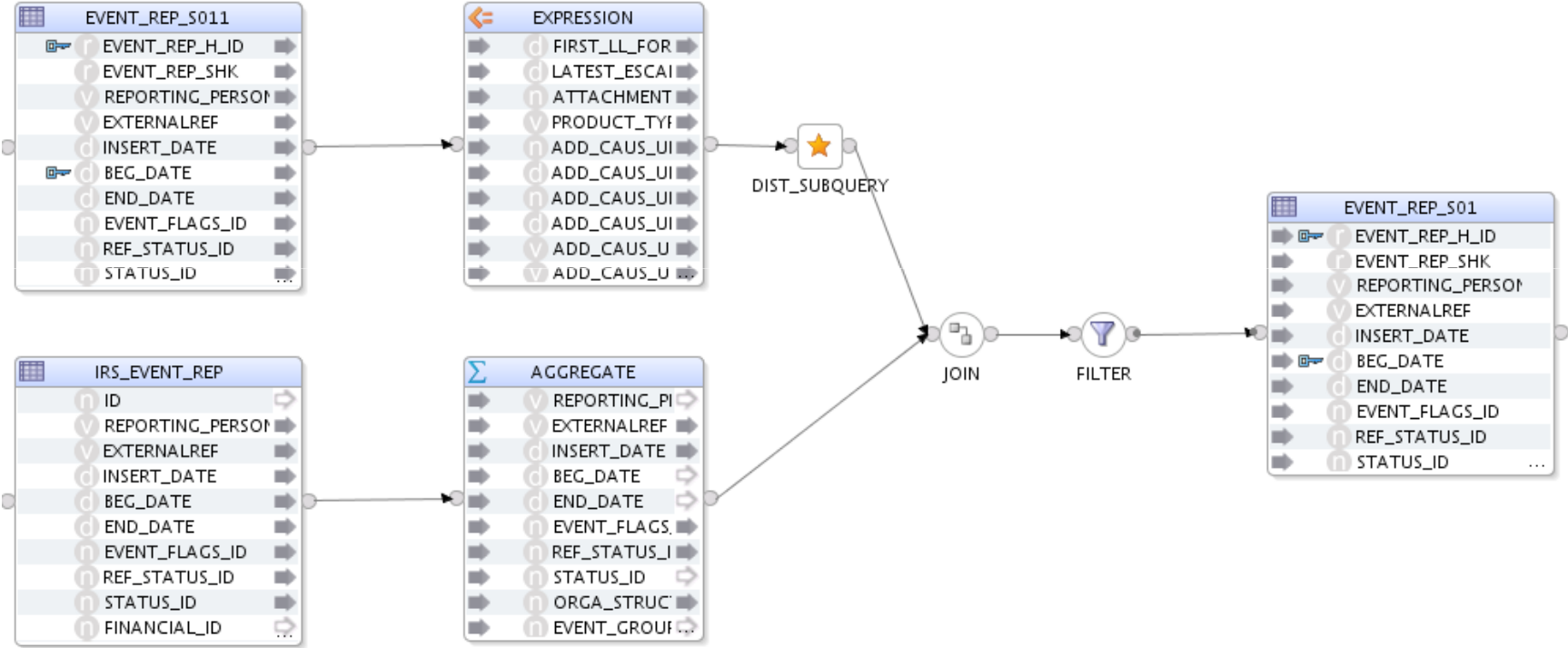


Vorgehensweise am Beispiel eines Satelliten



Vorgehensweise am Beispiel eines Satelliten

Verbinden des Filters mit dem Target und Einfügen der Expressions



Möglichkeiten und Grenzen

- Theoretisch ist alles möglich was manuell entwickelt werden kann
 - Datentyp Konvertierung
 - Data Quality Prüfungen
 - Abfangen von Sonderfällen
- Praktisch gibt es Begrenzungen
 - Komplexität
 - Wartbarkeit



Fragen

