

# Software Product Lines

## Concepts, Analysis and Implementation

### Programmier-Paradigmen für Software-Produktlinien (3/3)

**Dr. Malte Lochau**

Malte.Lochau@es.tu-darmstadt.de

# Inhalt

## I. Einführung

- Motivation und Grundlagen
- Feature-orientierte Produktlinien

## II. Produktlinien-Engineering

- Feature-Modelle und Produktkonfiguration
- Variabilitätsmodellierung im Lösungsraum
- Programmierparadigmen für Produktlinien

- Präprozessoren, Komponenten/Frameworks
- FOP, AOP, DOP
- **Build-Systeme**

## III. Produktlinien-Analyse

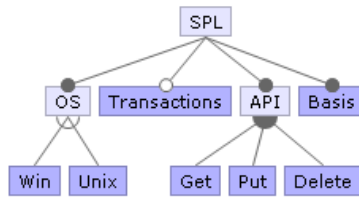
- Feature-Interaktion
- Testen von Produktlinien
- Verifikation von Produktlinien

## IV. Fallbeispiele und aktuelle Forschungsthemen

# Software-Product-Line Engineering

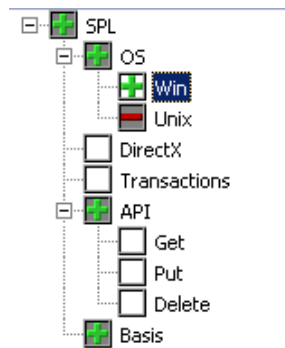
Domain Eng.

Feature-Modell

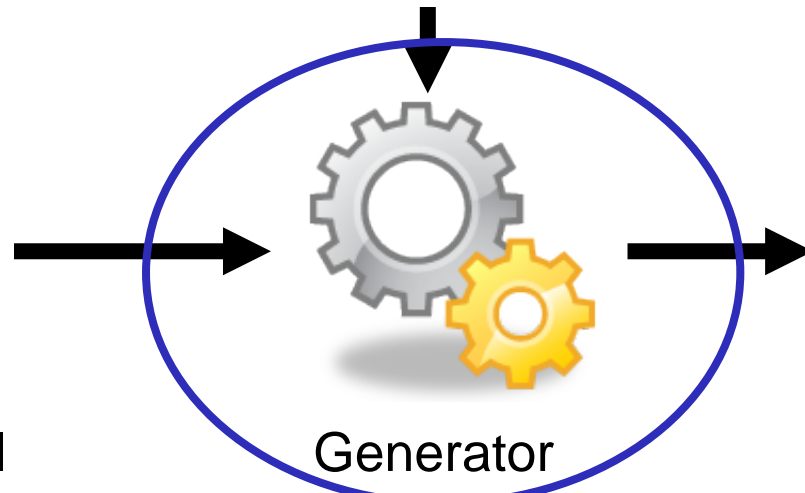


Application Eng.

Feature-Auswahl



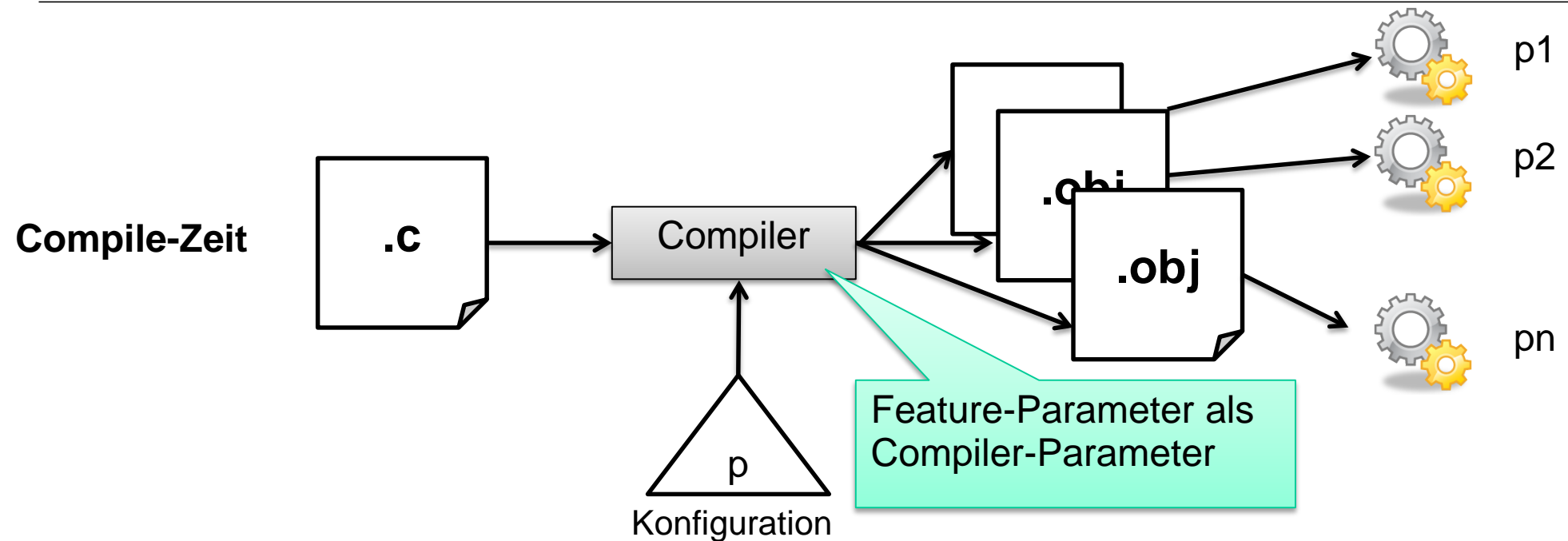
Wiederverwendbare Implementierungsartefakte



CUST_NO	CUSTOMER	CONTACT	CONTACT	PHONE
1	1,001 Signature ...	Dale J.	Little	(619) 531...
2	1,002 Dallas Tec...	Olen	Brown	(214) 990...
3	1,003 Bufile, Criff...	James	Bufile	(617) 481...
4	1,004 Central Trans	Escabeth	Brocket	61 211 9...
5	1,005 DT Systems	Tai	Wu	(852) 850...
6	1,006 DataServe	Tomas	Bright	(613) 221...
7	1,007 Mrs. Beauv.		Mrs. Beauv.	
8	1,008 Anini Vacat.	Leilani	Briggs	(808) 930...
9	1,009 Max	Max		22 01 23...

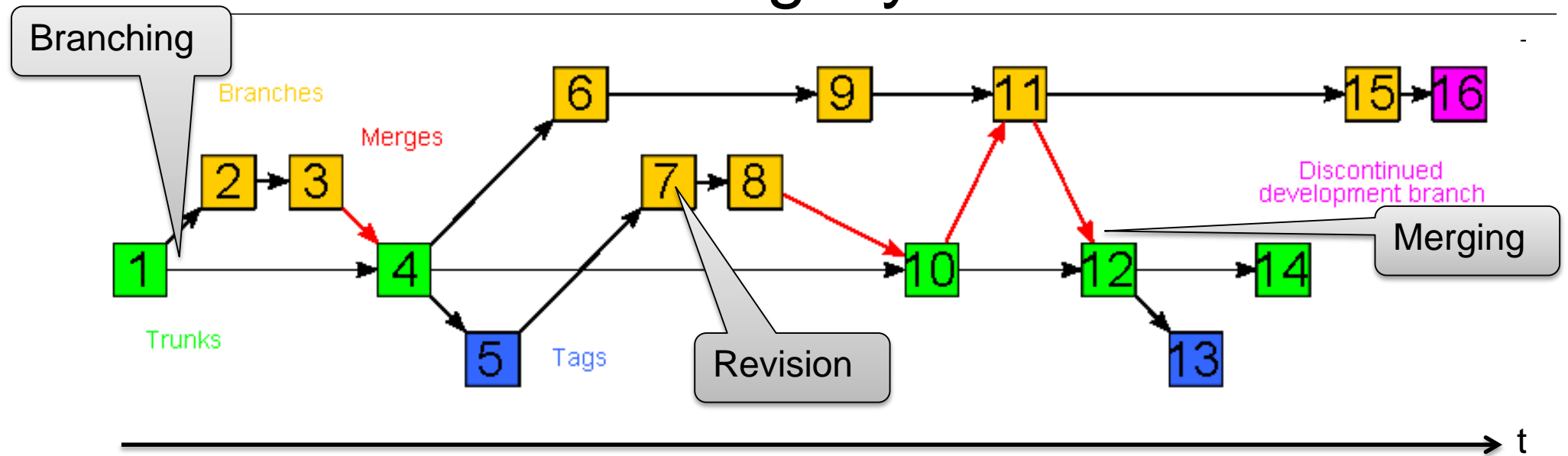
Fertiges Program

# Variabilität zur Compile-Zeit



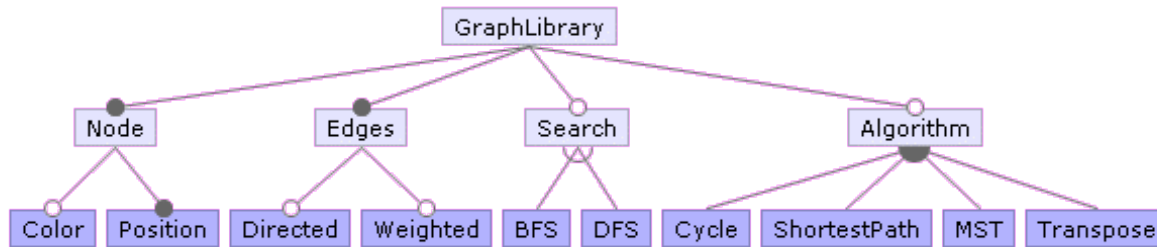
- Nur der für die Programmvariante benötigte Quelltext wird übersetzt
- Quelltext wird vor der Übersetzung passend ausgewählt (CPP), zusammengebaut (FOP), ...
- Ziel: Konfigurationsspezifische, optimierte Programmvarianten

# Versionsverwaltungssysteme



- Versionierung von Quelltextdateien als Teil des Konfigurationsmanagements
- Archivierung alter Quelltextversionen mit Historie
  - Zeitstempel und Benutzerkennung
  - Änderungen werden als “Delta” gespeichert
- Prozess: Checkout – Change – Commit – Update – Change – Commit - ...
- Beispielsysteme: CVS, SVN, Visual SourceSafe, Perforce, SCCS, Git, Mercurial. ...

# Beispiel: Versionierung der Graph-SPL



Schnellere Suche

Varianten

**Variante 3** ohne Gewichte  
und ohne Farben

**Variante 2** ohne Gewichte

**Basis-Variante**  
(mit Farben und Gewichten)

Versionen

NullPointerException  
Exception  
abfangen

# Verwaltung von SPL Artefakten

---

Eine SPL umfasst sämtliche Feature-Artefakte als Ergebnis des Domain-Engineering:

- Java Code
- Dokumentation
- Modelle
- Build-Skripte: Ant / Makefile, ...
- Lizenzen
- Grammatiken
- Kompilierte Dateien
- HTML, JavaScript, CSS
- Testfälle
- ...

=> Zu beachten: Merging und Konfliktbehandlung von Binärdateien schwierig

# Versionen einer Produktlinie

Artefakt-Revisionen →

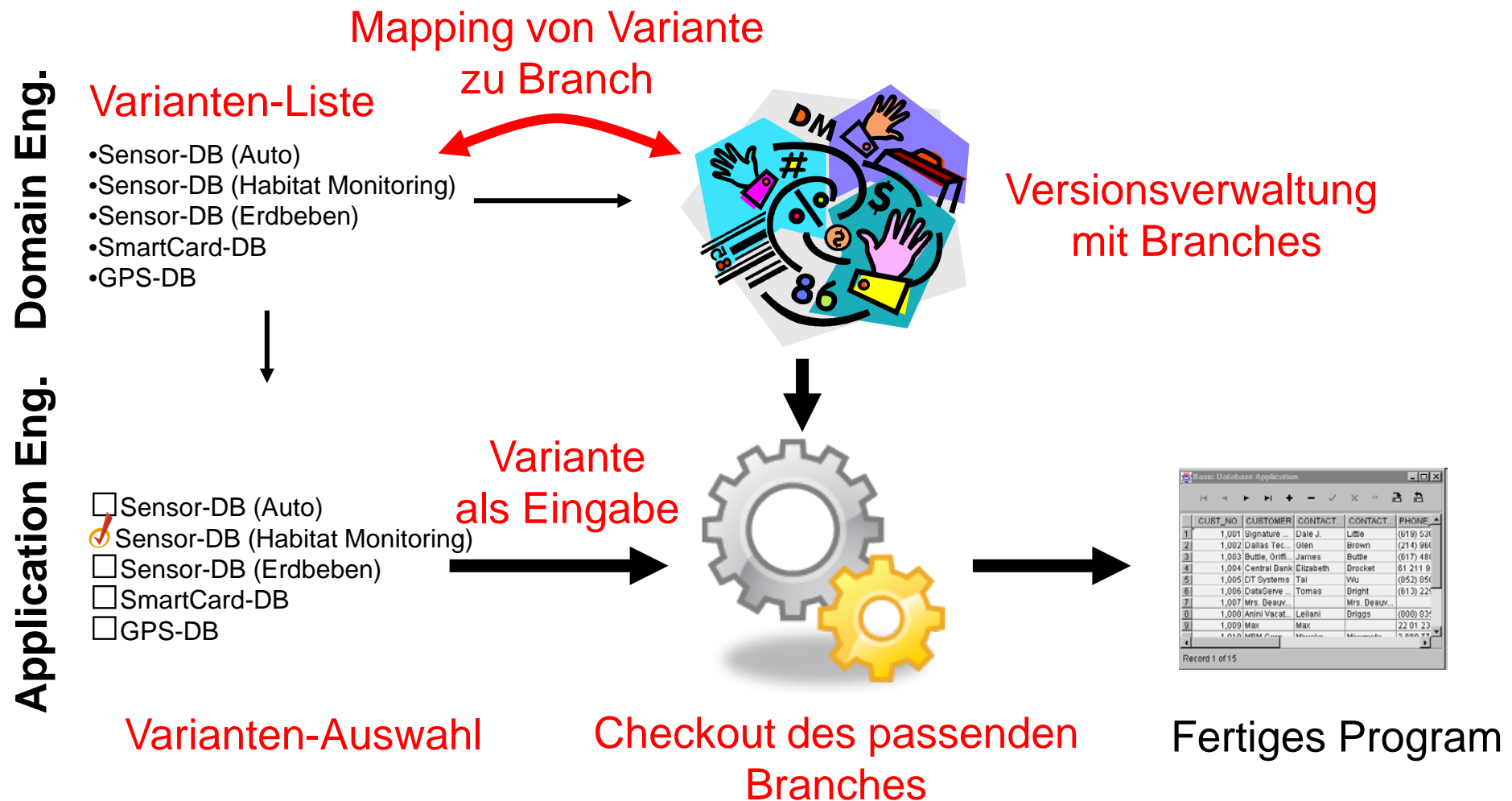
	V1.0	V1.1	V2.0	V3.0
Sensor-DB (Auto)	X	X	X	X
Sensor-DB (Habitat Monitoring)	X	X	X	
Sensor-DB (Erdbeben)			X	X
SmartCard-DB	X	X	X	X
GPS-DB				X

- **Version einer Produktlinie:** aktuelle Version aller Artefakte
- **Version einer Produktvariante:** Version der verwendeten Artefakte zum Zeitpunkt der Produktgenerierung



# Produktlinien mit Versionsverwaltung

- Entwicklung von Varianten in Branches
- Mergen von Änderungen zwischen Branches



# Diskussion: Produktlinien mit Versionsverwaltung

---

## Vorteile

- Etablierte, stabile Systeme
- Wohldefinierter Prozess
- Gute Werkzeugintegration

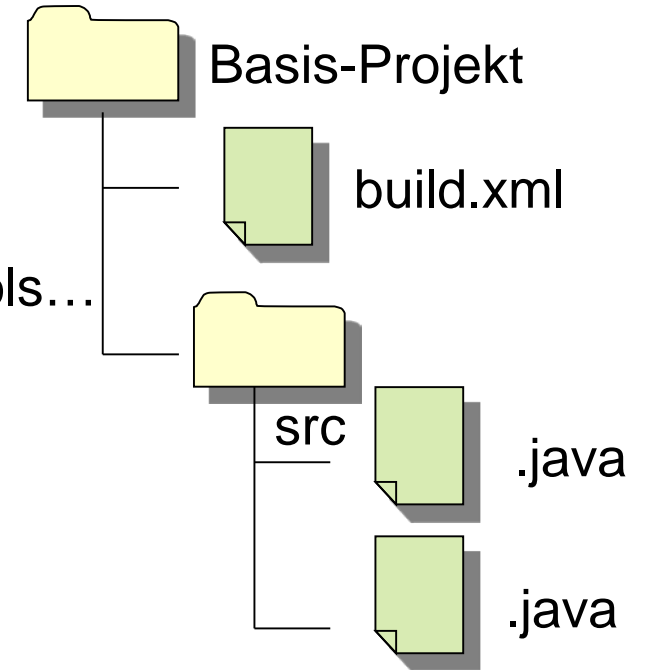
## Nachteile

- Vermischen von Revisionen und Varianten
  - Bei Major Release möglicherweise Re-Implementierung der Minor Releases
  - ...
- Entwicklung von Varianten, nicht Features: flexible Komposition von Feature-Artefakten nicht direkt unterstützt
- Keine strukturierte Wiederverwendung (meistens Copy & Edit)
- Hoher Wartungsaufwand („manuelles“ Merging)

# Produktlinien mit Build-Systemen

## Build-Systeme

- Automatisierung des Build-Prozesses
- Kopiert Dateien, ruft Compiler auf, startet weitere Tools...
- Mehrere Schritte mit Abhängigkeiten / Bedingungen
- Werkzeuge: make, ant, maven, ...

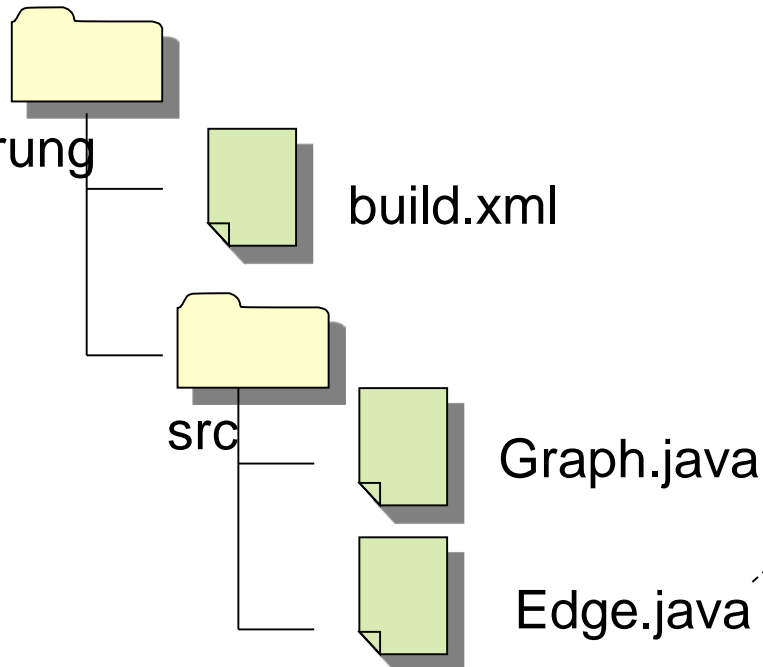


## Produktlinien mit Build-Systemen

- Pro Variante eine Konfigurationsdatei / ein Build-Skript
- Variable Artefakte (Dateien) beim Übersetzen einbeziehen / ausschließen
- Basis-Dateien mit produktspezifischen Varianten überschreiben

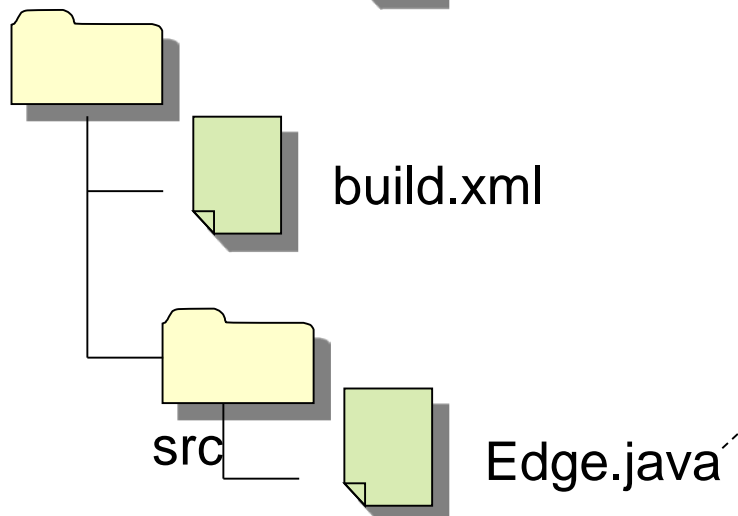
# Beispiel: Graph-Bibliothek

Basis-  
Implementierung



```
class Edge {  
    Node a, b;  
    Edge(Node _a, Node _b) { a = _a; b = _b; }  
    void print() {  
        a.print(); b.print();  
    }  
}
```

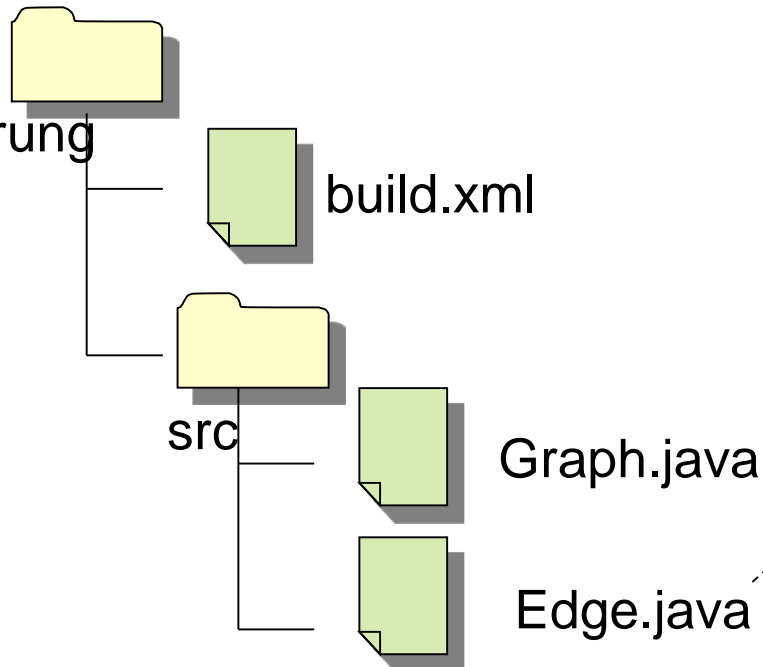
Variante mit  
Gewichten



```
class Edge {  
    Node a, b;  
    Weight weight;  
    Edge(Node _a, Node _b) { a = _a; b = _b; }  
    void print() {  
        a.print(); b.print();  
        weight.print();  
    }  
}
```

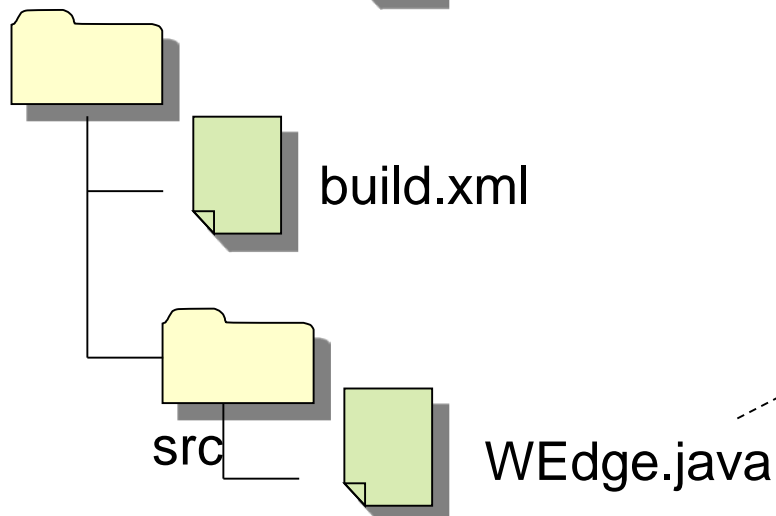
# Beispiel: Graph-Bibliothek

Basis-  
Implementierung



```
class Edge {  
    Node a, b;  
    Edge(Node _a, Node _b) { a = _a; b = _b; }  
    void print() {  
        a.print(); b.print();  
    }  
}
```

Variante mit  
Gewichten



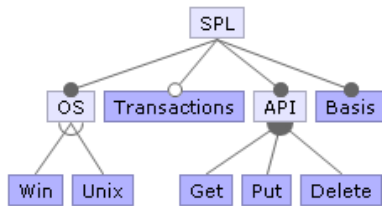
```
class WEdge extends Edge {  
    Weight weight;  
    void print() {  
        super.print();  
        weight.print();  
    }  
}
```

...und alle new-Aufrufe anpassen;  
ggf. Factory Pattern verwenden

# Produktlinien mit Build-Systemen






Application Eng. Domain Eng.

## Feature-Modell



## Basis-Implementierung



- Sensor-DB (Auto) 
- Sensor-DB (Habitat Monitoring) 
- Sensor-DB (Erdbeben) 
- SmartCard-DB 
- GPS-DB 



Build Skript pro Variante  
+ spezifische Dateien

Standard Build  
(make, ant, ...)

CUST_NO	CUSTOMER	CONTACT	CONTACT	PHONE
1	1,001 Signature	Date J.	Little	(619) 531
2	1,002 Dallas Tec.	Glen	Brown	(214) 961
3	1,003 Butte, Grift	James	Butte	(617) 481
4	1,004 Central Bank	Elizabeth	Drocket	61 211 9
5	1,005 DT Systems	Tai	Wu	(852) 051
6	1,006 DataGene	Tomas	Bright	(613) 221
7	1,007 Mrs. Beauv...		Mrs. Beauv...	
8	1,008 Anini Vacat...	Lellani	Griggs	(808) 831
9	1,009 Max	Max		22 01 23
10	1,010 IBM Comp	Michael	M...	5 00 77

Record 1 of 15

Fertiges Program

# Diskussion: Produktlinien mit Build-Systemen

---

## Vorteile:

- Relativ einfacher Mechanismus
- Hohe Flexibilität – beliebige Änderungen für Varianten
- Wenig Vorplanung notwendig

## Nachteile

- Entwicklung für jede Variante getrennt, ggf. hoher Aufwand
- Änderungen nur auf Dateiebene (überschreiben ganzer Dateien)
- Änderungen an der Basisimplementierung schwierig

# Referenzen

---

- *M Staples, D Hill: Experiences adopting software product line development without a product line architecture.* Proceedings APSEC, pp.176- 183, 2004 (Artikel zu Erfahrungen in der Industrie beim Einsatz von Build-Systemen und Versionsverwaltung)