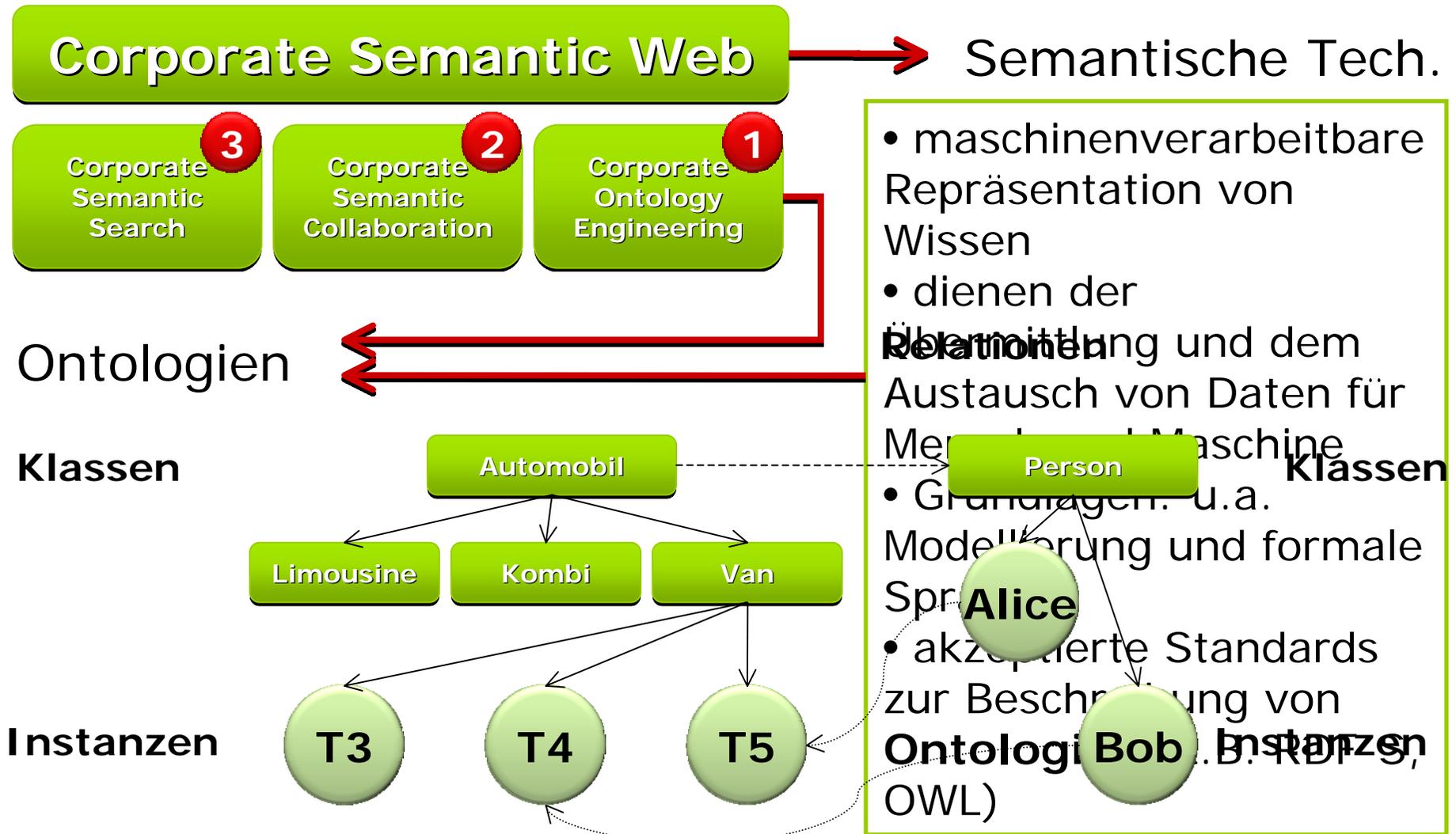


# Corporate Ontology Engineering

Ontologie Versionierung, Modularisierung und Integration

Markus Luczak-Rösch, Gökhan Coskun  
AG Netzbasierende Informationssysteme  
Freie Universität Berlin

# Was sind Ontologien (im Corporate Semantic Web)?



# Überblick

---

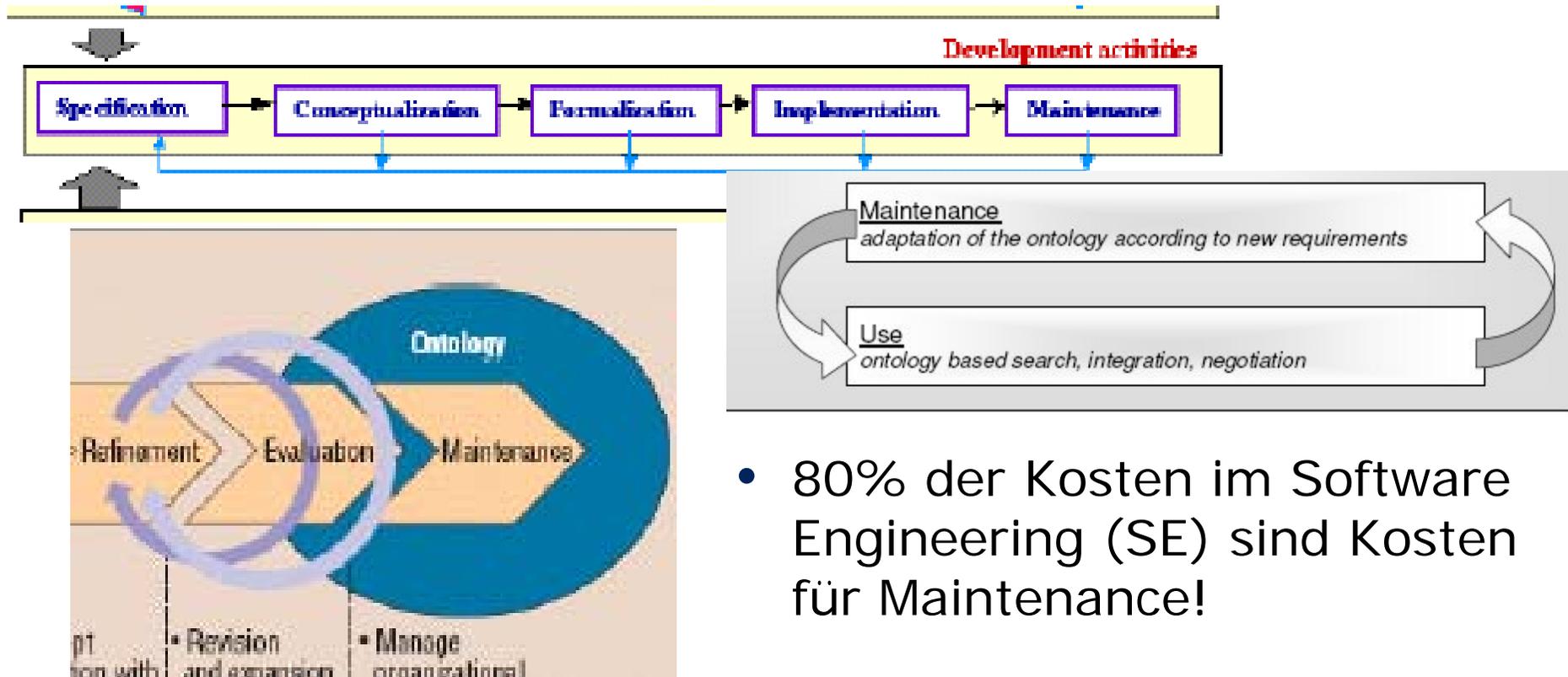
- Was sind Ontologien?
- Kurzeinführung Ontology Engineering
- Ontologien in der Unternehmensrealität
- Anforderungen an Corporate Ontology Engineering
- COLM - Ontologielebenszyklen aus Unternehmensperspektive
  - Agile Ontologieevolution
  - Modularisierung und Integration

# Ontology Engineering

---

- Entwicklung von
  - einer „individuellen Kunst“
  - zu einer eigenständigen Ingenieursdisziplin, charakterisiert durch Definition von
    - Prozessen
    - Lebenszyklen
    - Methodiken, Werkzeugen und Ontologiesprachen
- grundsätzliche Prinzipien vom Software Engineering adoptiert
  - Neueste Ansätze definieren Übersetzungsleitfäden (z.B. spezielles OE Glossar)

# Prozess- und kostenorientierte Probleme



- 80% der Kosten im Software Engineering (SE) sind Kosten für Maintenance!
- Was verbirgt sich hinter Ontology Maintenance?

# Ontologien im Unternehmensumfeld

- Szenario 1:  
ontologiegestütztes  
Wissensmanagement

- Heterogene Infra-  
struktur
- Asynchrone  
Evolution der  
Einzelsysteme



- Szenario 2:  
ontologiebasierte Web  
Services

- agiler Ontologie-  
lebenszyklus

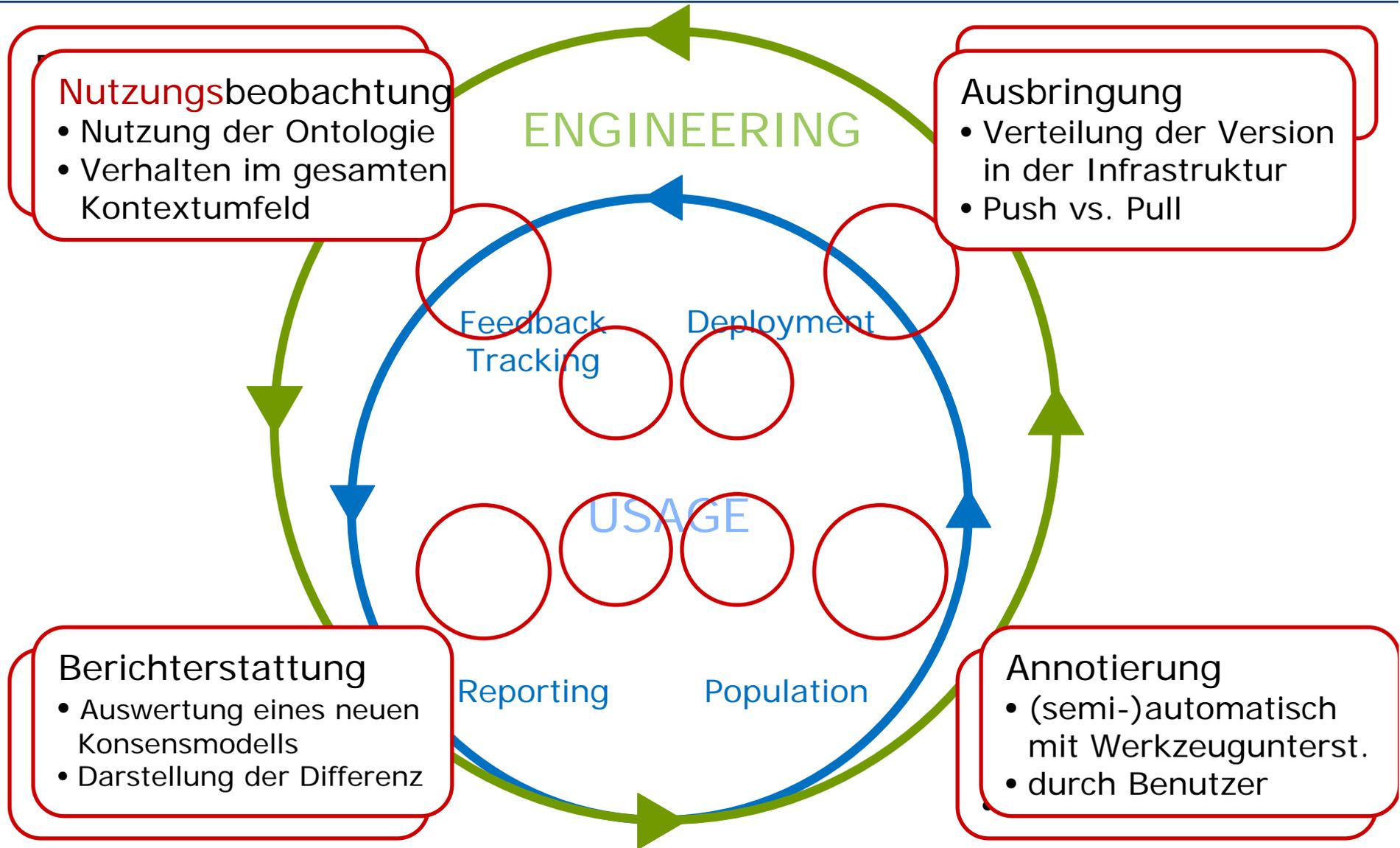


# Anforderungen an Corporate Ontology Engineering

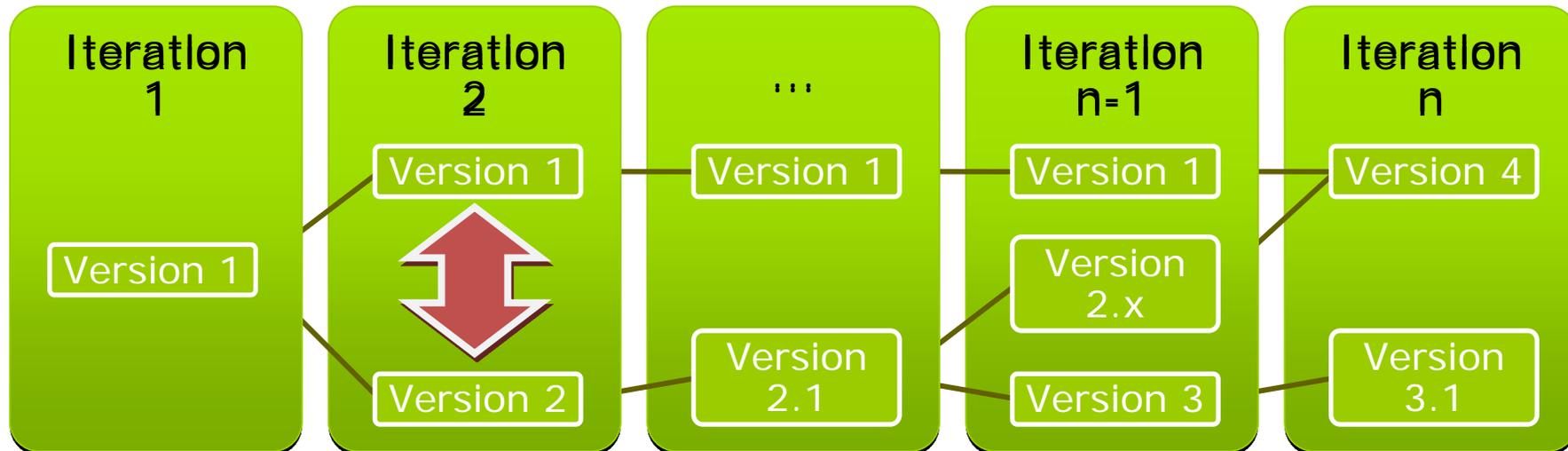
---

- Kernphase im Prozess ist die Evolution von Ontologieprototypen
- es wird ein agiler Prozess mit Werkzeugunterstützung benötigt
- Werkzeuge für Experten UND nicht-Experten
- agile Versionierungsstrategie
- nachvollziehbare Kosten-Nutzen-Modelle

# COLM – Ontologielebenszyklen aus Unternehmensperspektive

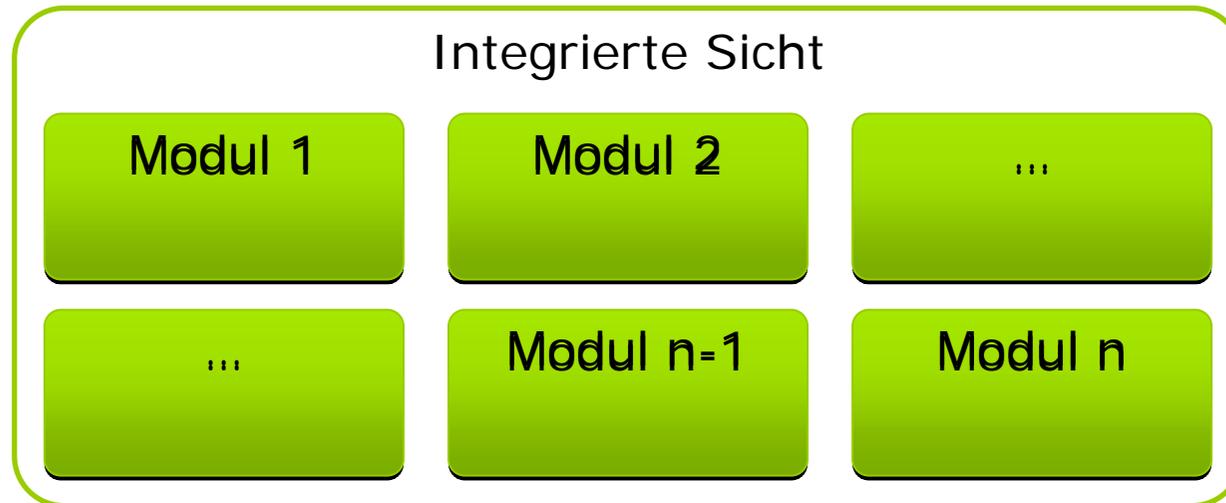


# Agile Ontologieversionierung



- Warum versionieren?
  - die aktuelle Version beschreibt eine Domäne unvollständig oder schlecht
  - asynchron entwickelte Ontologien über die gleiche Domäne
- Wie ermittelt man nachvollziehbar die Notwendigkeit einer neuen Ontologieversion?

# Modulare Aufteilung von Ontologien



- Warum modularisieren?
  - Übersichtlichkeit/Wartung
  - Local Reasoning
  - Wiederverwendung
- Wie erzeugt man aus großen Ontologien kleinere Subontologien?

**Koexistierende Versionen verwalten**

**Wartbarkeit durch Modularisierung**

**Kosteneffizienz und –transparenz**

