Reengineering-Projekte in der Ausbildung

Klaus Bothe
Institut für Informatik
Humboldt-Universität zu Berlin

1. **Motivation durch reale Reengineering-Projekte**

2. **Reengineering-Aktivitäten**

3. **Probleme mit Reengineering-Tools**

4. **Pläne und Probleme**

Motivation durch reale Reengineering-Projekte

R. Koschke:

Vorlesungen zum Thema Software-Reengineering,

2. Workshop Software Reenginnering,

Bad Honnef, Mai 2000

Negativ-Image:

→ **Gegenstand des RE:**

schlecht strukturierte Altsysteme,

in unmodernen Programmiersprachen

V o r l e s u n g e n: Motivationsprobleme?

P r o j e k t e: höhere Motivation?

Unser Projekt: der Beginn

- Sommer 1998
- Anfrage:

  Arbeitsgruppe der Physik der HU-Berlin

- Bitte: Unterstützung bei der Wartung

- Gegenstand:

  Labor-Steuerprogramm (RTK-System)

  Eigenentwicklung

  Entwickler nicht mehr anwesend

Unser Projekt: Anfangszustand

- Dokumente und Umgebung:

  Programmsourcen vorhanden (?)

  großes Programm (27 KLOC)

  Borland C++ , Windows 3.11

  keine weiteren Dokumente !!!

  (Pflichtenheft, Design,

  Nutzerhandbuch ...)

- Auftraggeber: großes Interesse, kooperativ
- Zeitdruck: gering
- Geld: (fast) keines

Projekt-Ziele:

- ***Reverse engineering***

  - Erarbeitung fehlender Dokumente
    (Anforderungs-Spezifikation, Entwurf, ...)
  - Architektur-Überarbeitung
  - Programm-Verstehen (Kommentierung der Quellen)

- ***Nutzerwünsche (Reengineering)***

  - Stabilität des Systems
  - Erweiterung: neue Geräte, ...
  - Portierung: Borland -> Visual, Win 3.11 -> Win 98

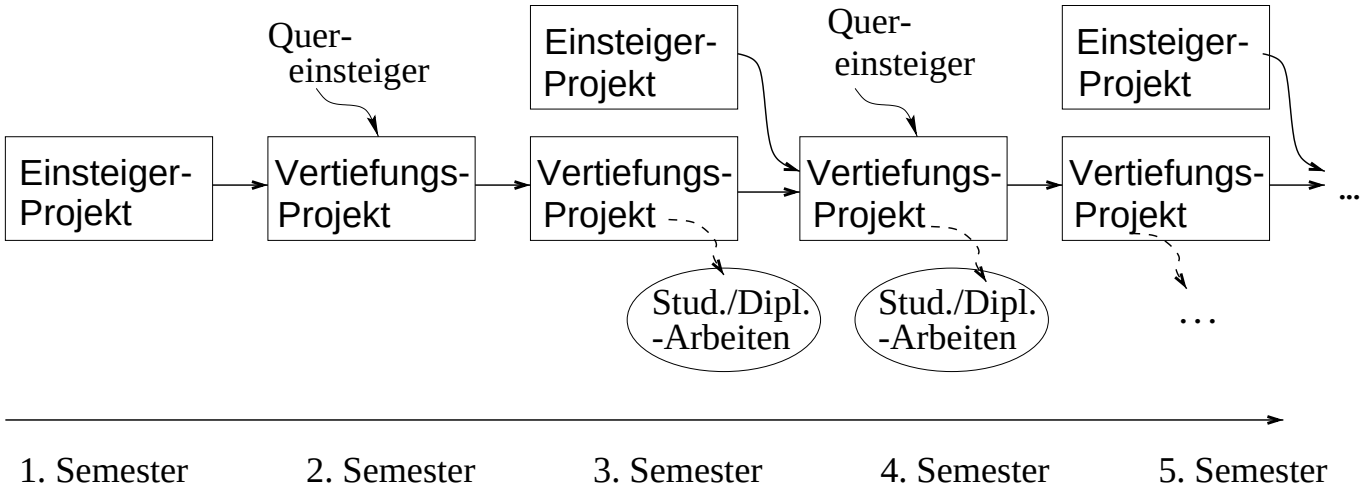
- ***Allgemeine Projektziele***

  Team work, Projekt-Management,

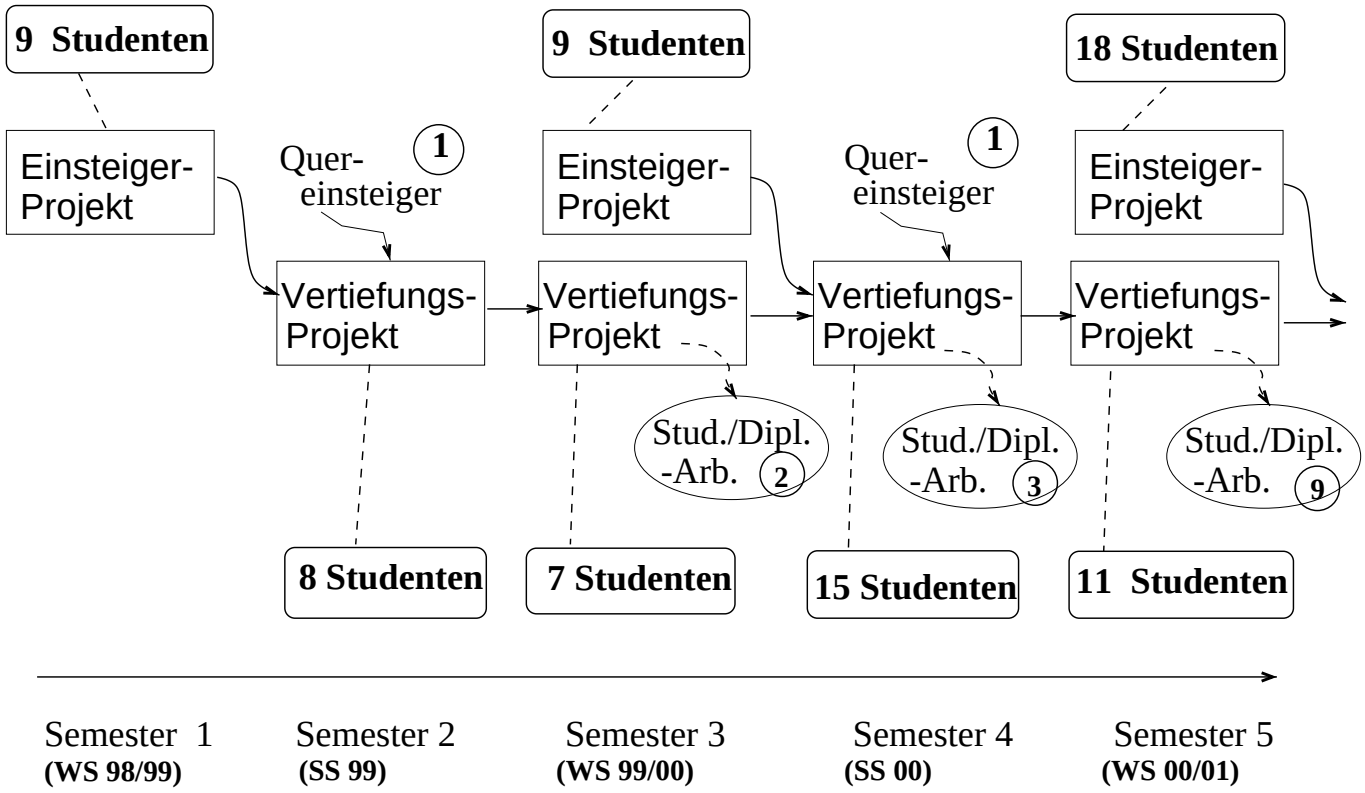
  Kontakt mit realen und fachfremden Kunden,

  Studien- und Diplomarbeiten, ...

Multisemester-Projekt: Organisation



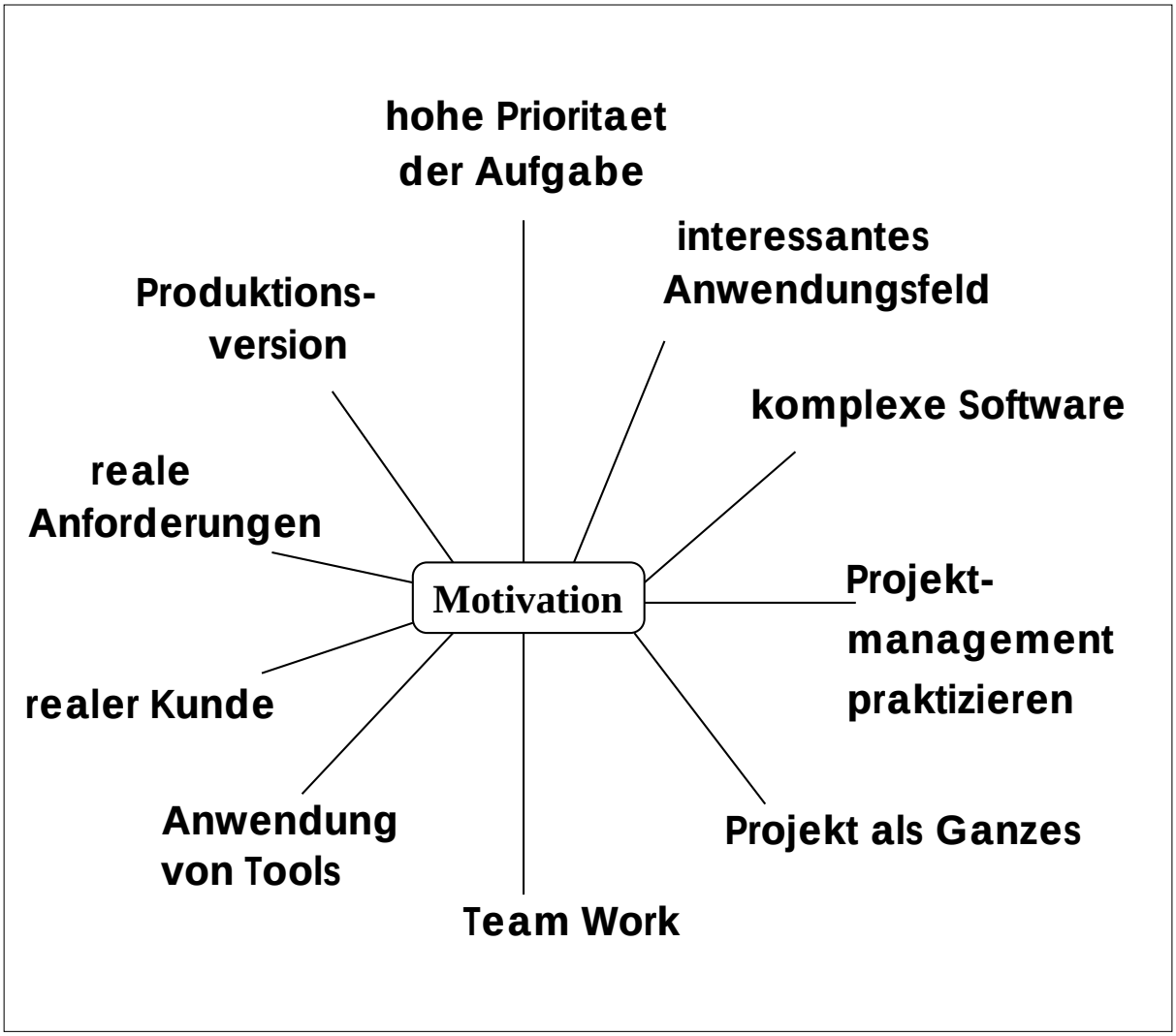
Multisemester-Projekt: Statistik



→ Gesamtzahl der Studenten: 9 + 1 + 9 + 1 + 18 = 38

M o t i v a t i o n

durch reale Reengineering-Projekte



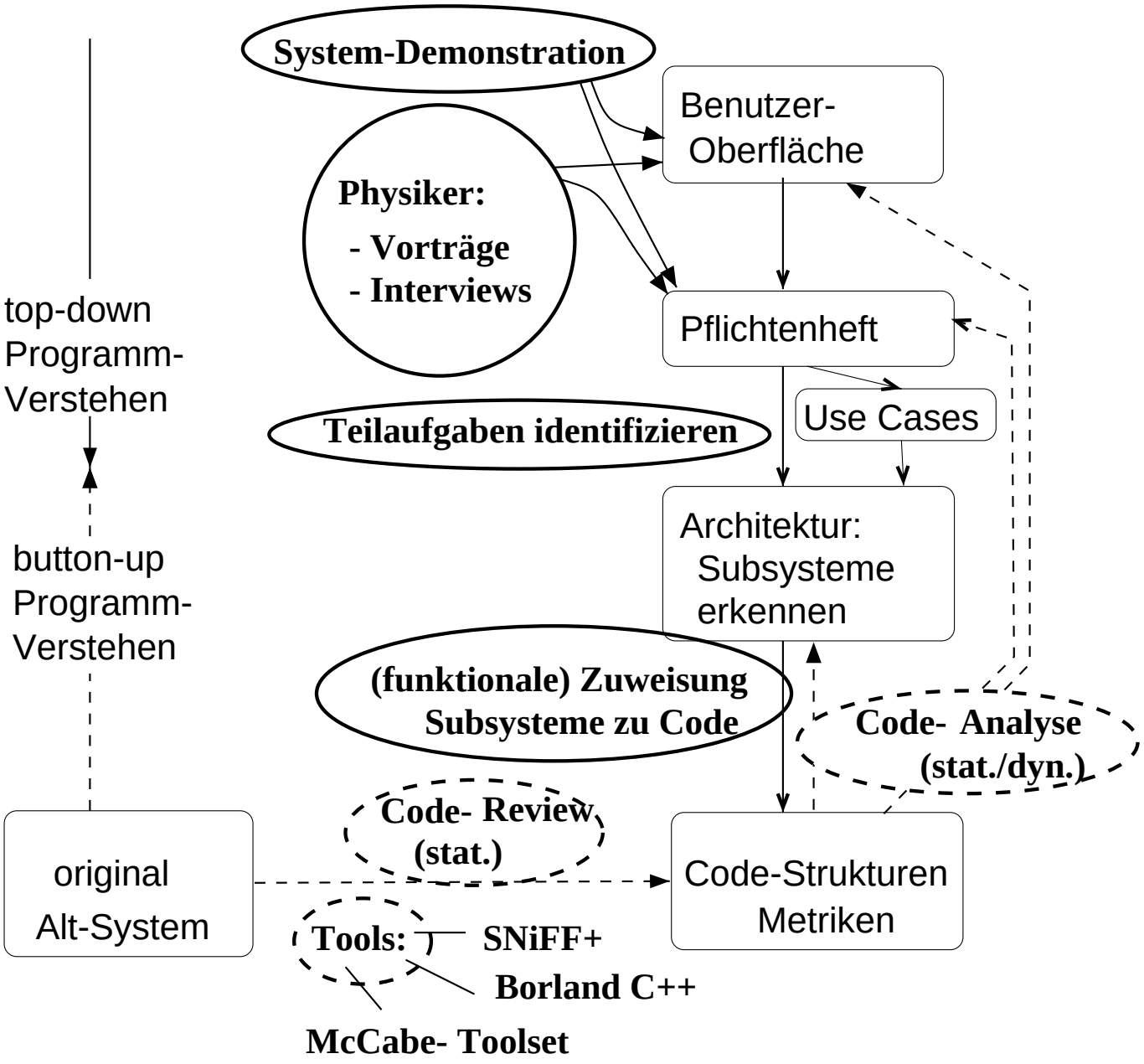
1. Motivation durch reale Reengineering-Projekte

2. Reengineering-Aktivitäten

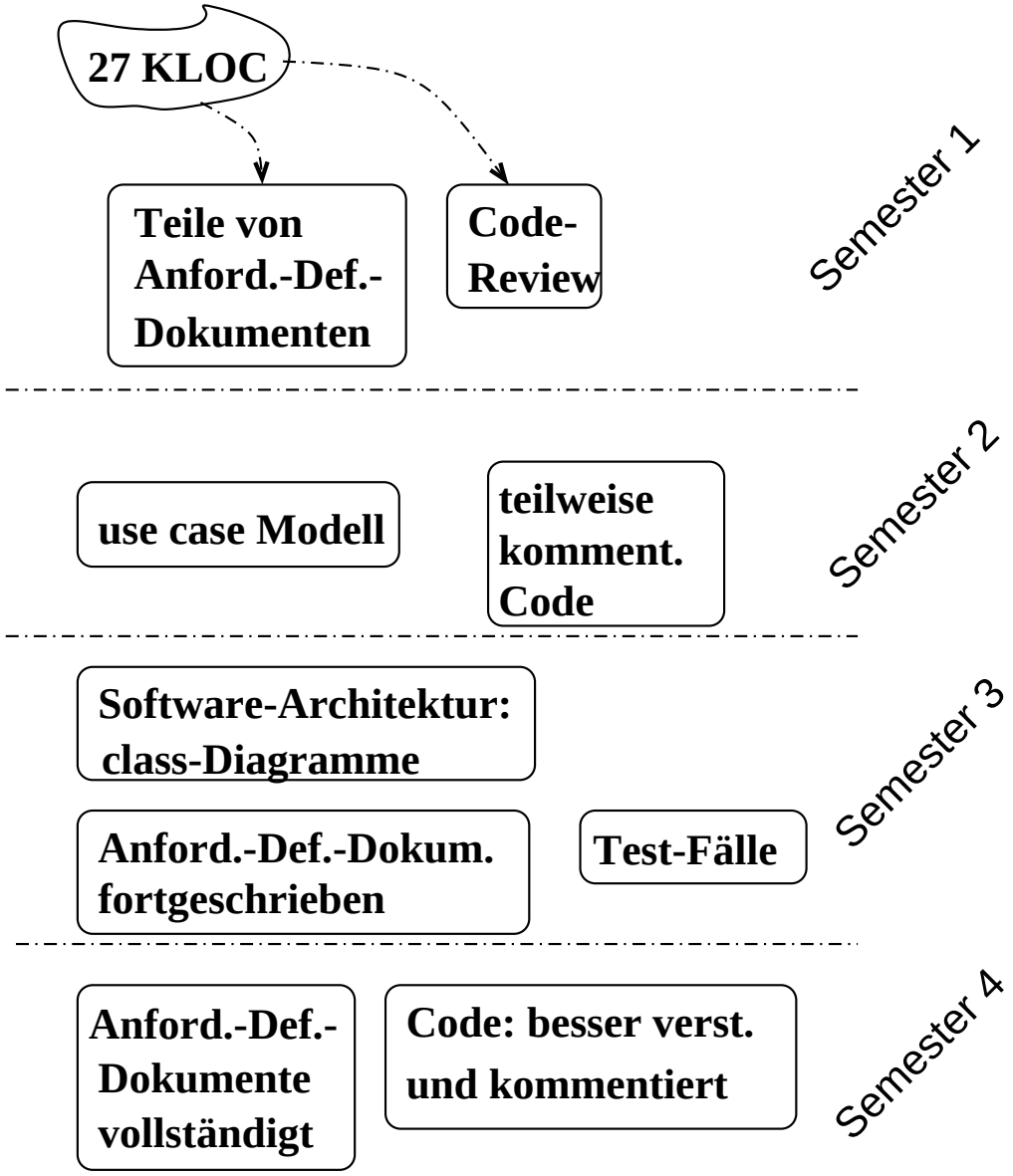
3. Probleme mit Reengineering-Tools

4. Pläne und Probleme

Projektaktivitäten: Reverse Engineering



Projekt-Wiederverwendung: über Jahre + wachsende Reife



-> Das Projekt wächst von Semester zu Semester

-> Die Abschnitte bauen aufeinander auf

-> Wiederverwendung: jedes Semester auf höherem Reifeniveau

Projektaktivitäten: Sonstiges

Erweiterungen:

- neue Geräte
- Automatisierung bisher manueller Vorgänge

Testsystem:

- Regressionstest
- Umgebungssimulation

Projektmanagement:

- Versionsverwaltung
- Web-Repository: alle Dokumente
- Team Work
- Treffen, Protokolle
- Kontakt mit realen Kunden / Nichtinformatiker

1. Motivation durch reale Reengineering-Projekte

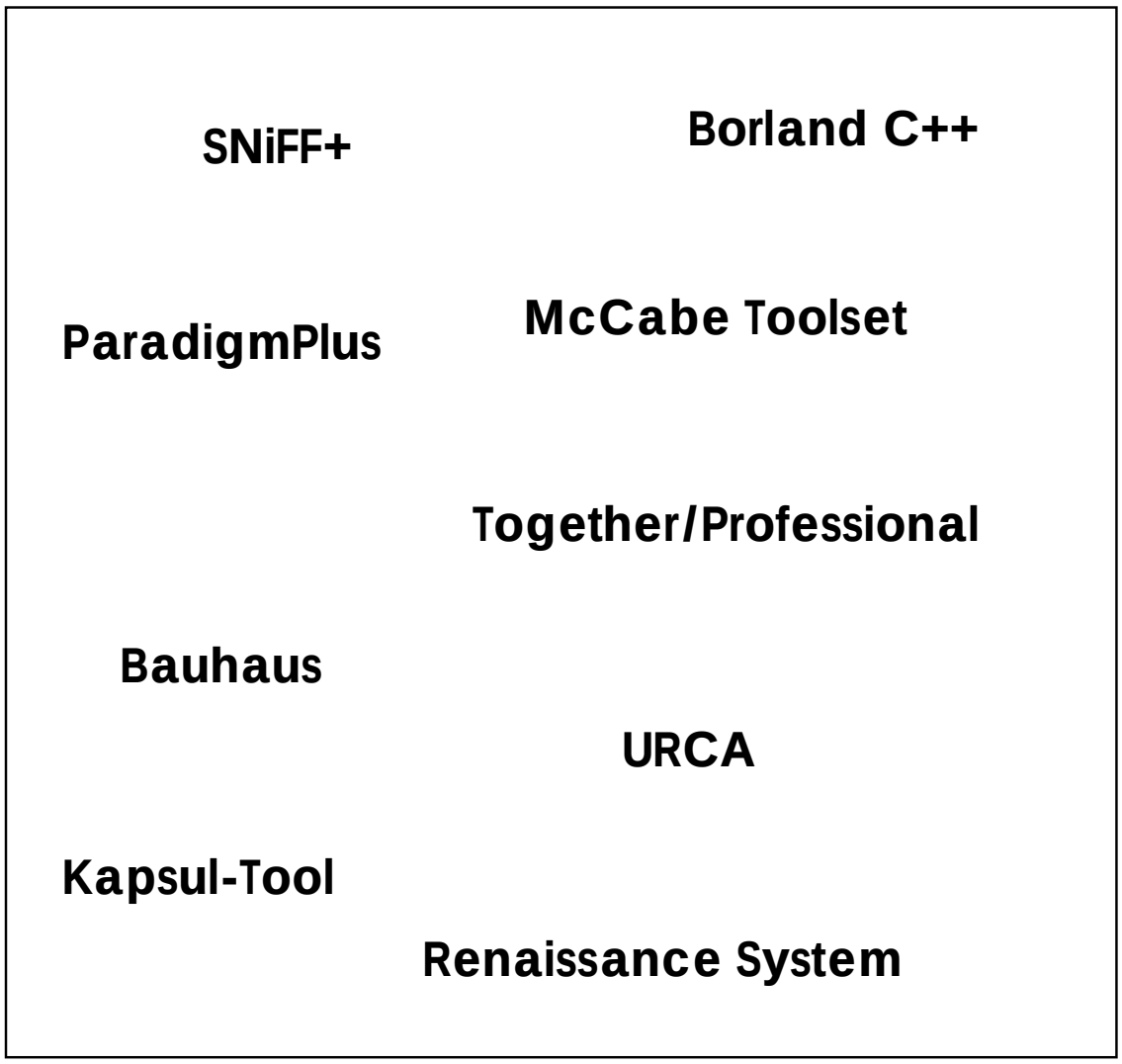
2. Reengineering-Aktivitäten

3. Probleme mit Reengineering-Tools

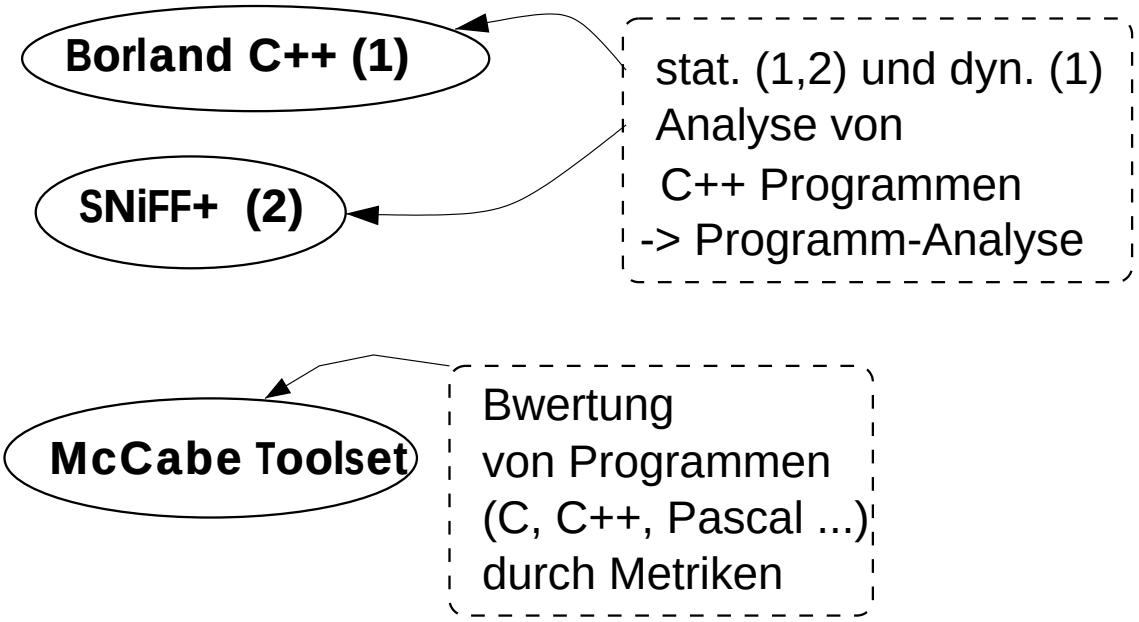
4. Pläne und Probleme

Reverse Engineering Tools

(von uns betrachtet)



Reverse Engineering Tools: unterschiedliche Funktionalität (1)



ParadigmPlus

Rational Rose

Together/Professional

Bauhaus

URCA

Kapsul-Tool

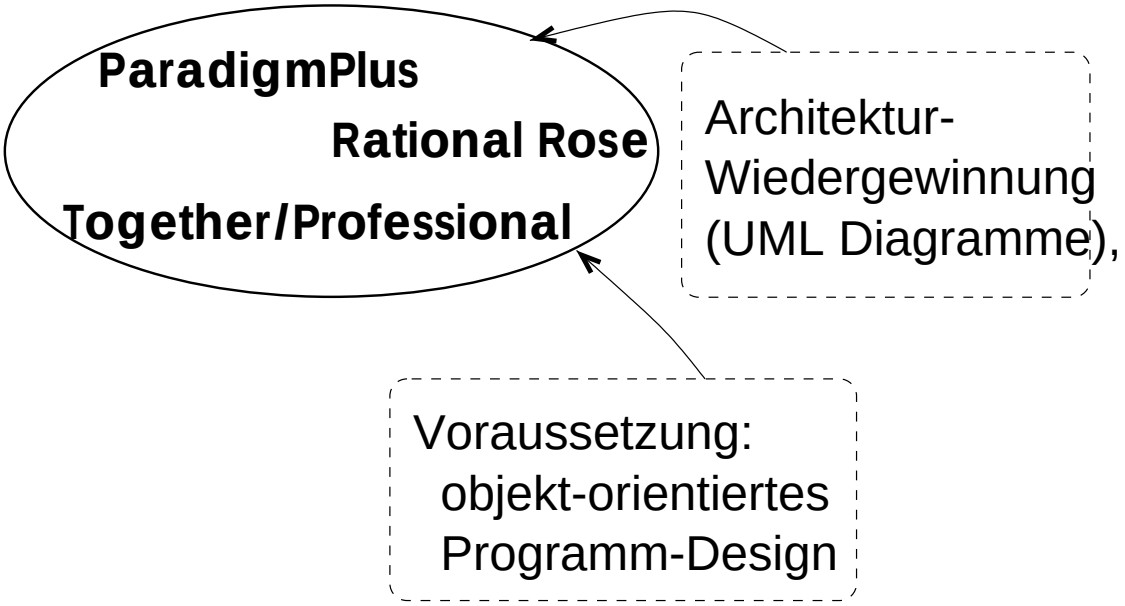
Renaissance System

Reverse Engineering Tools: unterschiedliche Funktionalität (2)

Borland C++

SNiFF+

McCabe Toolset



Bauhaus

URCA

Kapsul-Tool

Renaissance System

Reverse Engineering Tools: unterschiedliche Funktionalität (3)

Borland C++

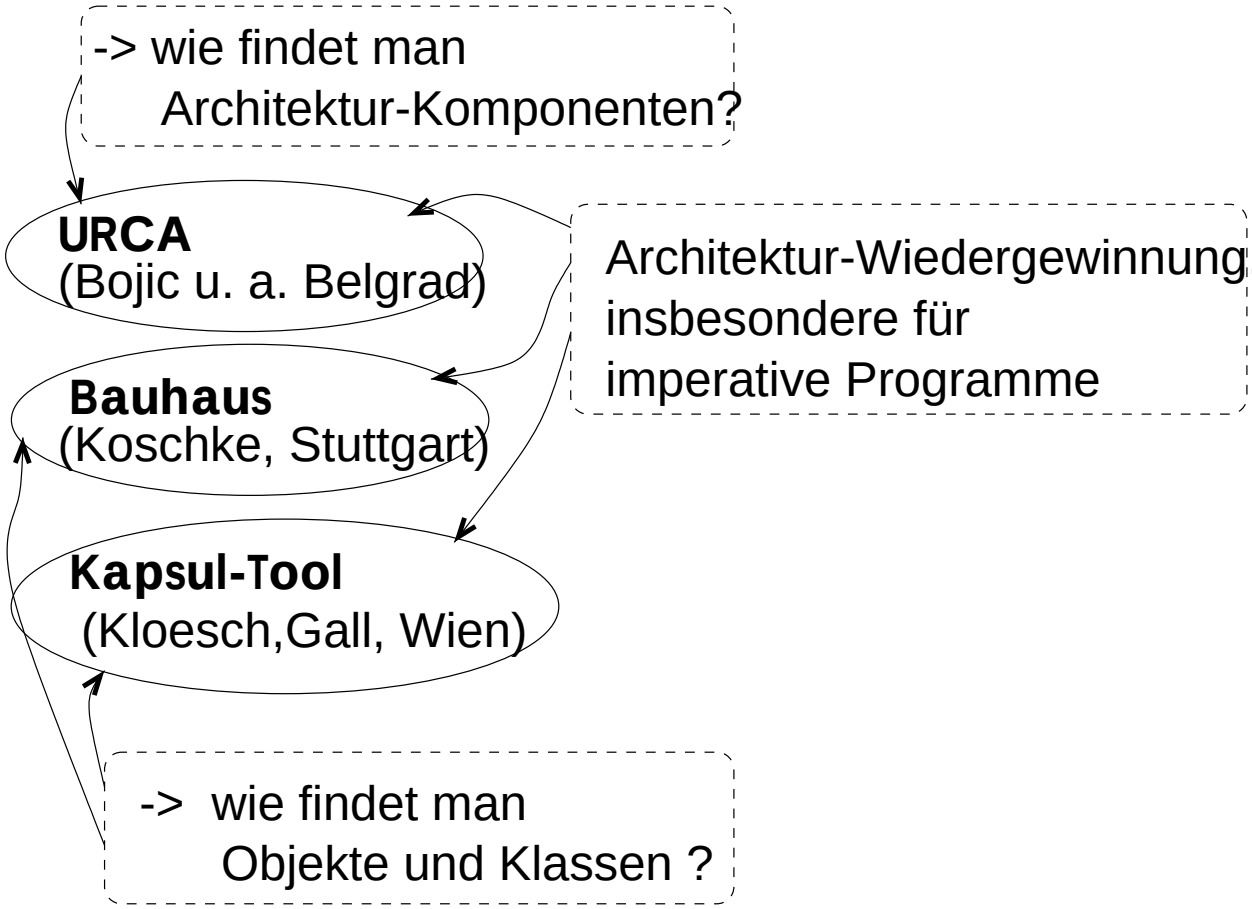
SNiFF+

McCabe Toolset

ParadigmPlus

Rational Rose

Together/Professional



Reverse Engineering Tools:
tatsächlicher Einsatz

Borland C++ √√

SNiFF+ √√

McCabe Toolset √

ParadigmPlus √

Together/Professional ?

Bauhaus √̸

URCA √̸

Kapsul-Tool ?

Renaissance System ?

√√ eingesetzt
√ teilweise eingesetzt
√̸ nicht einsetzbar
? nicht untersucht

1. **Motivation durch reale Reengineering-Projekte**

2. **Reengineering-Aktivitäten**

3. **Probleme mit Reengineering-Tools**

4. **Pläne und Probleme**

Pläne und Probleme

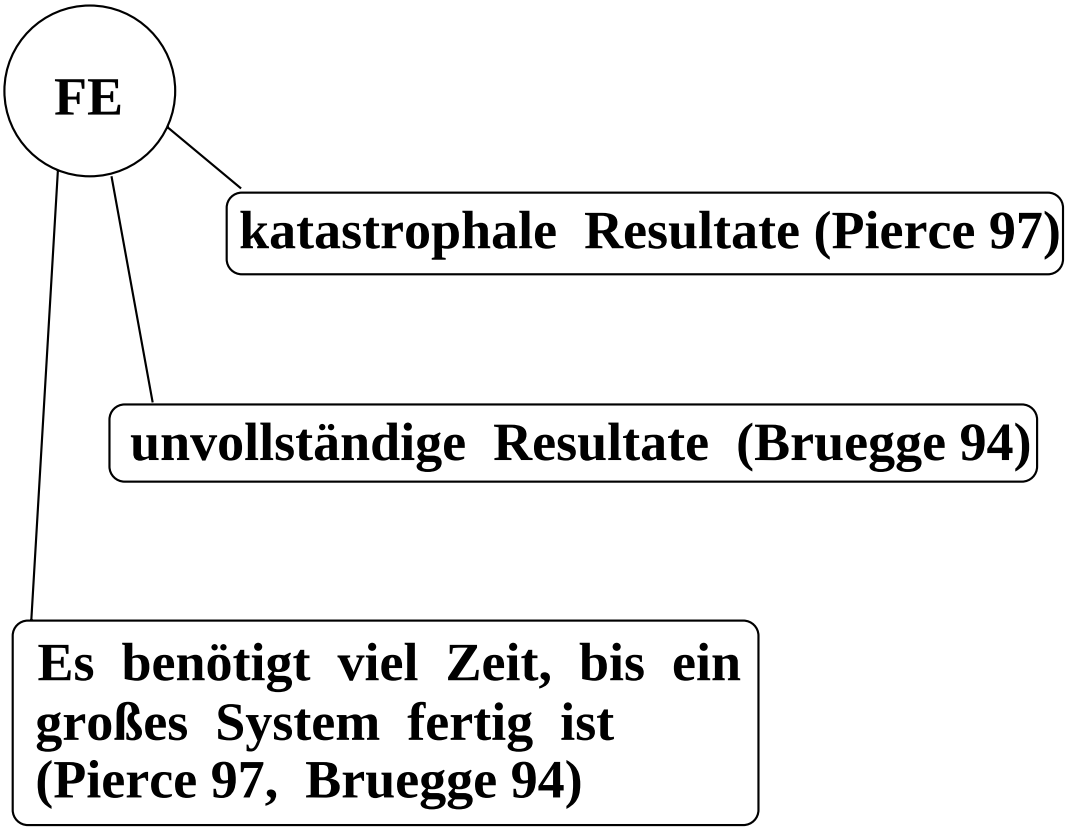
- Gravierendstes Problem:

  SW-Architektur s e h r schlecht

  (Komponenten ohne klare Schnittstellen,
  .c- .h-File-Strukturen mangelhaft,
  Mischung von Oberfläche und Funktionalitat,
  Mischung von imperativem und oo Stil)

- Testsystem, insb. automatischer Regressionstest
- dazu: Umgebungssimulation
- Portierung: Borland C++ nach Visual C++
  (vorläufig gescheitert)
- Erweiterungen: Wünsche der Physiker
- Suche nach geeigneten Tools
  (insb. zur Architektur-Verbesserung, s. o.)

Vorzüge des RE gegenüber dem FE in Hochschul-Projekten (1)



Vorzüge des RE gegenüber dem FE in Hochschul-Projekten (2)