UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

# Vorlesung
# Software aus Komponenten

## 1. Komponenten – Markt - Standards

Prof. Dr. Hans-Gert Gräbe
Wintersemester 2006/07

## Organisatorisches

- Termin: freitags 7:30 – 9:00 Uhr, Kl. HS Härtelstraße
- Abschluss: Klausur 1 und 2 Anfang Februar
  - Klausur 1 (90 min), Ergebnis anrechenbar als
    - Prüfung im Modul 10-202-2311, wenn Seminarschein vorliegt
    - PL im Bachelor- oder Masterstudiengang Informatik (alt)
  - Klausur 2 (45 min), Ergebnis anrechenbar als
    - studienbegleitende Modulprüfung im Kernfach Angewandte Informatik (Diplom-Informatiker)
    - APL (Master-Studiengang)
  - Alle anderen Anrechnungen bedürfen vorheriger Absprache
- Ansprechpartner
  - Prof. Dr. Gräbe <graebe@informatik.uni-leipzig.de>
  - Sprechzeiten: montags 13-14 Uhr oder nach Vereinbarung per Mail
- Informationen: Über die Webseiten der Abteilung oder meine Webseite am Institut

# Inhalt dieser Vorlesung

1. Komponenten, Markt, Standards
2. Grundlegende Prinzipien der Wiederverwendung
   - Komponenten, Objekte, Moduln, Schnittstellen
   - Die Vererbungsproblematik
3. Komponentenmodelle und –plattformen
   - Grundlagen
   - OMG und CORBA
   - Sun und Java
   - Microsoft und .Net
   - Vergleich
4. Komponenten und Architekturen
5. Komponenten und IT-Professionals

Die Vorlesung orientiert sich am Buch [Szyperski, 2002].

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

## Warum Software aus Komponenten?

- „Stehen auf den Schultern von Riesen"
- praktische Wiederverwendung von Softwareteilen
  - kein wiederholtes Schreiben gleichartigen Codes
  - andere Formen: Wiederverwendung von Quellcode, Verwendung von Bibliotheken, Höhere Abstraktionsprinzipien (Design, Architektur)
- Amortisation von Investitionen durch Mehrfachverwendung
- typischer ingenieur-technischer Zugang
  - Vorteile: höhere Qualität, schnellere Entwicklung, flexiblere Reaktion auf wechselnde Anforderungen
- Einsatz wiederverwendbarer Softwareteile ist auch innerhalb eines Unternehmens sinnvoll
  - erfordert konzeptuelle Vorbereitung und Standardisierung
- Softwarekomponenten und Markt

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

## Software-Erstellung als ingenieurtechnischer Prozess

**Standardsoftware**

hohe Stückzahl,
große Einsatzbreite

Parallelen zum
Werkzeugmaschinenbau

**Auftragssoftware**

geringe Stückzahl,
spezielle Einsatzbedingungen

Parallelen zum Anlagenbau

**Software als Produkt**

Komponenteneinsatz innerhalb eines Unternehmens im Rahmen von Produktlinien

**Software als Prozess**

Komponenten als unternehmens-übergreifende Infrastruktur: Bausteine, aus denen Applikationen gefertigt sind

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

# Was sind Software-Komponenten?

- „Components are for composition"
  - Komposition erlaubt es, präfabrizierte „Dinge" wiederzu-verwenden, indem sie in immer neuen Kombinationen zusammengesetzt werden.
  - Allgemeiner Wiederverwendungsansatz ist für unsere Betrachtungen zu breit
  - Ansatz muss den Aspekt „Aufbau aus Teilen" stärker berücksichtigen
    - es reicht nicht, monolithische Software für Wiederverwendung zu „zerstückeln"
    - Wiederverwendungsansatz muss bereits bei der Erstellung der Komponenten berücksichtigt werden, nicht erst a posteriori
    - Funktionalität muss genügend allgemein geplant sein, um Wiederverwendung in ausreichender Anzahl verschiedener Kontexte zu ermöglichen
    - Verallgemeinerung muss genügend speziell sein, um Wiederverwendung praktisch sinnvoll zu ermöglichen
  - Wiederverwendung soll über Firmengrenzen hinaus möglich sein

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Definition des Begriffs „Software-Komponente“ ist auf unterschiedlichen Abstraktionsebene möglich

> Software-Komponenten sind ausführbare Software-Einheiten, die unabhängig hergestellt, erworben und konfiguriert werden und aus denen sich funktionierende Gesamtsysteme zusammensetzen lassen. [Szyperski]

- Zusammensetzungsaspekt steht im Vordergrund
  - dafür sind Standards erforderlich:
    - Einordnung in ein spezielles Komponentenmodell
    - ausführbar auf einer speziellen Komponentenplattform
- So zusammengesetzte Systeme heißen Komponenten-Software
- Unabhängigkeit und Ausführbarkeit
  - Unabhängigkeit von Entwicklung und Herstellung
  - Abstraktionen auf Quellcode-Ebene ausgeschlossen
  - älteste Beispiele für Software-Komponenten: Bibliotheken

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

# Vorteile des Komponentenansatzes

- Die Software-Krise großer monolithischer Anwendungen
- in-House-Lösung: Modularisierung und objektorientierte Ansätze
  - Problem: Wiederverwendung nur eigener Expertise
  - für komplexe Produkte muss die Expertise auf der ganzen Front vorgehalten werden
  - maximal von großen Firmen (SAP. Microsoft, etc.) zu schultern
  - nur für Software-Produkte sinnvoll einsetzbar
- Vorteil von Open-Source-Projekten ist weniger die Quelloffenheit als die Verfügbarkeit von qualitativ hochwertigen Software-Bausteinen
  - typisches Ergebnis arbeitsteiligen Vorgehens wie im Wissenschaftsbetrieb üblich
  - Alle entwickelten Ingenieurdisziplinen verwenden Komponententechnologien
- Komponenten-Software ist unter diesem Blickwinkel **der** Ausweg aus der Software-Krise

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

# Komponentenmarkt

- Ansatz brummt richtig, wenn die erforderlichen Komponenten in guter Qualität und ausreichend breiter Funktionalität verfügbar sind
  - Modell Komponentenmarkt vs. Open Source und GPL

- Komponentenmärkte existieren erst in Ansätzen
  - Warum Mehrzahl?
  - technologische und ökonomische Rahmenbedingungen erforderlich
    - technologische Bedingungen existieren seit Ende der 60er Jahre
    - Markt erfordert Standard, da Komponenten nur in einer entsprechenden Infrastruktur (Komponentenplattform) operieren können. Standardsetzung ist ein politisches Problem
  - Beispiel: Markt für VBX

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Warum ist Komponenten-Software auch in Prä-Marktphase attraktiv?
  - Vorbereitung auf kommenden Markt
  - Übergang zu komponentenbasierter Software ist aufwändig
  - Setzen von Standards und Sammeln von Kompetenz

- Die selbstverstärkende Wirkung eines Komponenten-Markts

- Wird es wirklich dauerhaft Komponenten-Märkte geben?
  - Der infrastrukturelle Charakter einer Software-Komponenten-Szene
  - Ökonomische Besonderheiten von Software und deren inhärente Tendenz zur Monopolisierung