

# Spezifikation der Performance- Eigenschaften von Softwarekomponenten



Andreas Schmietendorf, Arbeitsgruppe Software-Technik

André Scholz, Arbeitsgruppe Wirtschaftsinformatik

*Otto-von-Guericke-Universität Magdeburg, Fakultät Informatik*

# Agenda

- ◆ Motivation und Ausgangssituation
- ◆ Qualitätsklassen für Software-Komponenten
- ◆ Spezifizierung und Vertrieb
- ◆ Performance-Analyse von Komponenten
- ◆ Abbildung von Qualitätseigenschaften
- ◆ Zusammenfassung und Ausblick

# Motivation zur Untersuchung I

- ◆ Black-Box Sicht bei Anwendung von Komponenten
- ◆ Komponenten-basierte Softwareentwicklung
  - bis zu 80% auf der Basis von Komponenten und Services
  - qualitative Eigenschaften weitgehend unberücksichtigt
- ◆ Komponentenbeschreibungen inkl. Qualität
  - Vorschlag der Schweizer Bank
  - Vorschlag bei der Deutschen Telekom

## Motivation zur Untersuchung II

- ◆ Konsequenzen unperformanter IV-Systeme
  - Airport Denver: 2 Mrd. US\$
  - AOL USA: 320 Mill. US\$
  - geringe Kundenzufriedenheit, Image-Schäden
- ◆ Performance der Geschäftsprozesse als Erfolgsfaktor:
  - E-Commerce (Studien: Gartner-, Meta-Group,...)
- ➔ nur 10% aller SW-Projekte berücksichtigen Aufgaben des PE

## Ausgangssituation

Auszug aus einem technischen Report des SEI:

*No standard of practice exist for how to evaluate the performance of a software component. As reliance on COTS increases, the risk of discovering performance problems after it is too late may increase for systems for which performance predictability is critical.*

# Qualitätsbewertung nach ISO 9126

- ◆ Übertragbarkeit (Anpassbarkeit, Installierbarkeit),
- ◆ Anwendbarkeit (Erlernbarkeit, Beherrschbarkeit,...),
- ◆ **Effizienz (raum-, zeit- und ressourcenbezogen),**
- ◆ Funktionalität (Angemessenheit, Interoperabilität,...),
- ◆ Zuverlässigkeit (Fehlertoleranz,...),
- ◆ Wartbarkeit (Analysierbarkeit, Änderbarkeit, ...)

# Qualitätsklassen für Komponenten

Übertragbarkeit Q1	Anwendbarkeit Q2	Effizienz Q3	Funktionalität Q4	Zuverlässigkeit Q5	Wartbarkeit Q6
Anpassbarkeit Installierbarkeit	Erlernbarkeit Beherrschbarkeit Verständlichkeit	Raumbezogen Zeitbezogen Ressourcenbezogen	Angemessenheit Interoperabilität Genauigkeit	Fehlertoleranz Fehlerhäufigkeit Fehlerverfolgbarkeit	Analysierbarkeit Änderbarkeit Stabilität Testbarkeit

Beschreibung als 6-Tupel:

*Komponenten<sub>n</sub> (q1, q2, q3, q4, q5, q6)*

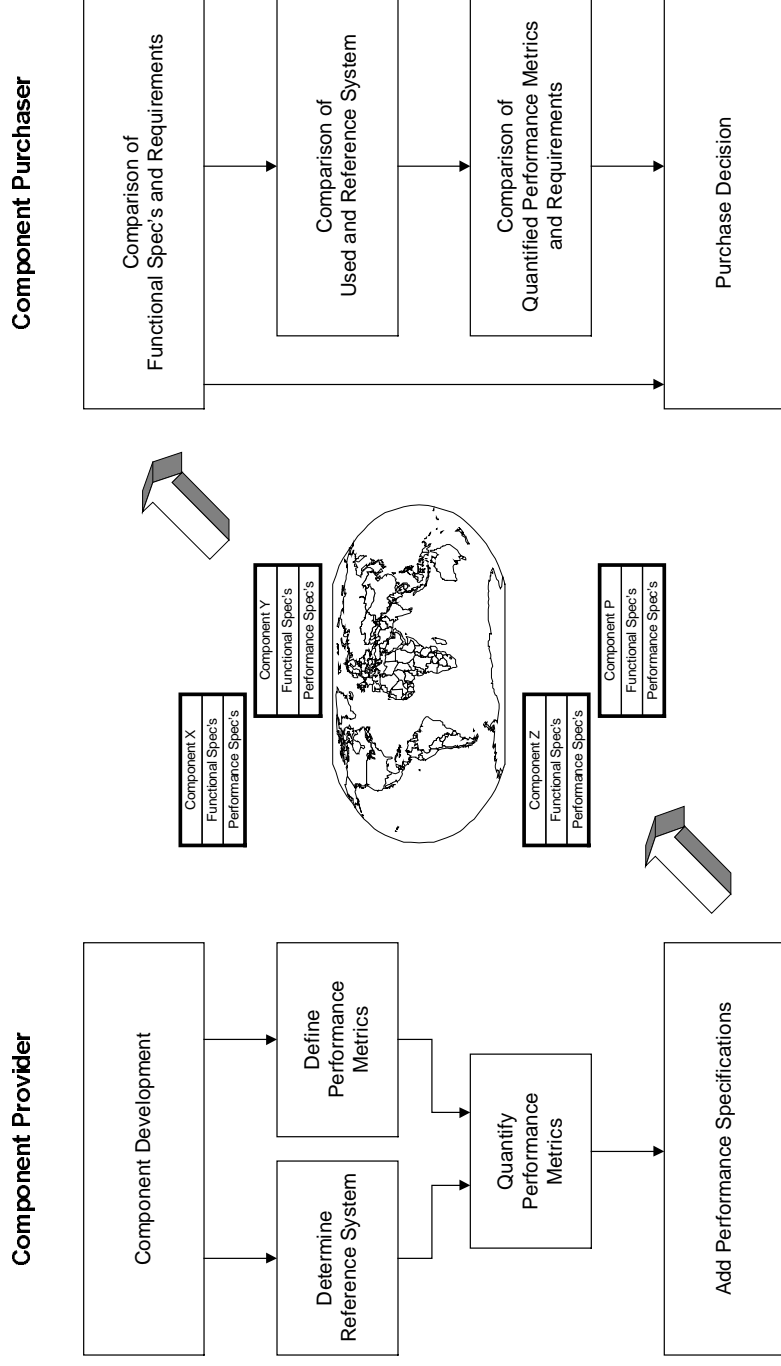
- differenzierte Beschreibung der Komponenten-Qualität
- Berücksichtigung von Skaleneigenschaften

# Abbildung von Qualitätseigenschaften

- ◆ **Komponenten-Repositories**
  - flache Textdatei
  - HTML-Datei mit Link auf die Komponente
  - ↳ keine „maschinelle“ Verwendung der Angaben
- ◆ **Integration in die Komponente**
  - Zugriff aus Entwicklungsumgebungen heraus
  - direkte Überführung in die Programmausführung



# Vertrieb von Komponenten I

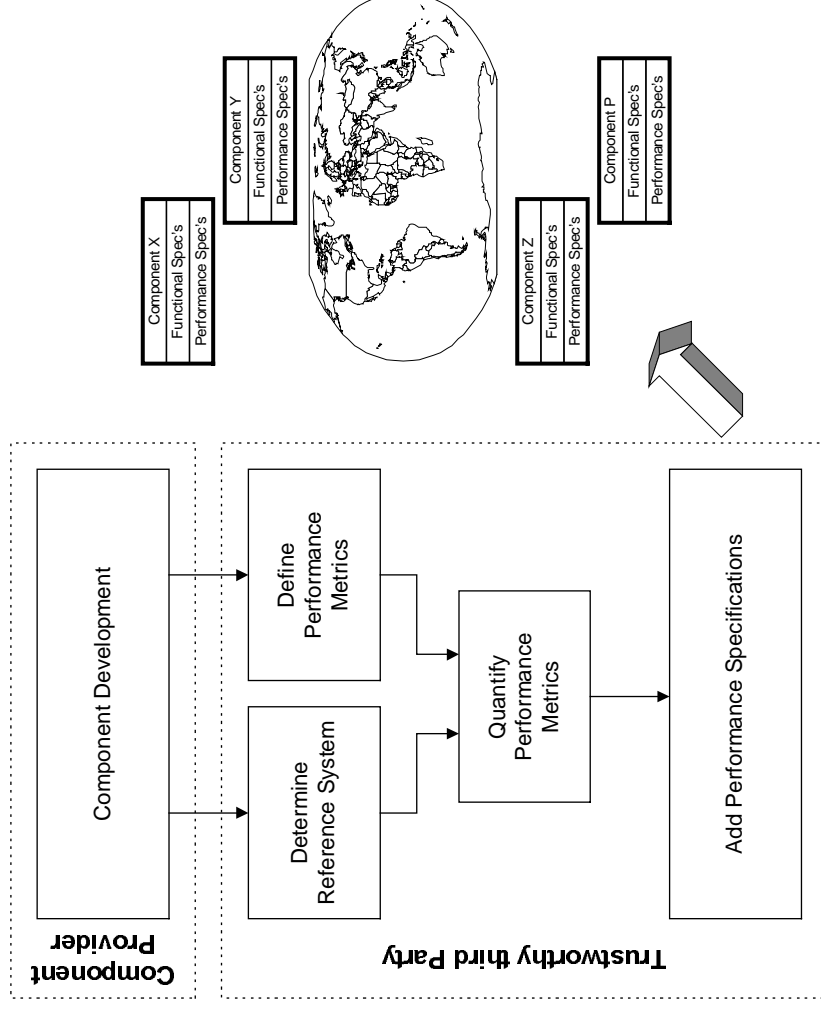


## Vertrieb von Komponenten II

Einführung einer vertrauenswürdigen Instanz:

- ◆ Spezifikation der Vorgehensweise
  - ◆ Verifizierung von Performance-Parametern
    - Arbeitslast
    - Ressourcenbedarf
    - Leistungseigenschaften
  - ◆ Vergabe eines „Komponenten-Zertifikats“
- ⇔ Vorgehensweise vergleichbar der von Benchmark-Organisationen

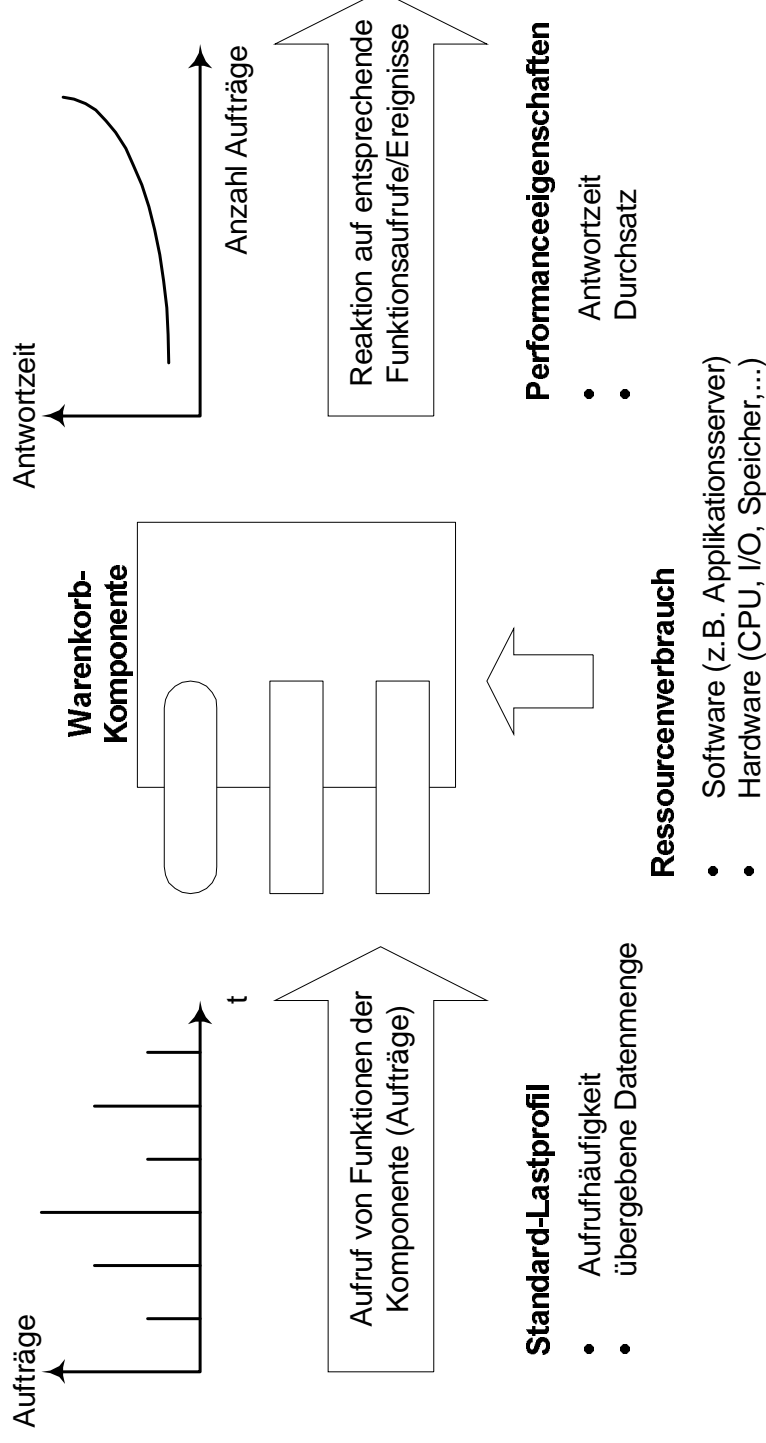
# Vertrieb von Komponenten III



# Methodik der Performanceanalyse I

- ◆ Entwicklung eines „typischen“ Lastprofils
- ◆ Bereitstellung einer Laufzeitumgebung
- ◆ Testdaten generieren
- ◆ Aufzeichnung elementarer Funktionsaufrufe
- ◆ Daten-Substitution der Funktionsaufrufe
- ◆ Konfiguration des Lasttreibers (Lastmix einstellen)
- ◆ Ausführung des Benchmarks
- ◆ Ergebnisauswertung
- ◆ Hinterlegung der Performanceinformationen

# Methodik der Performanceanalyse II

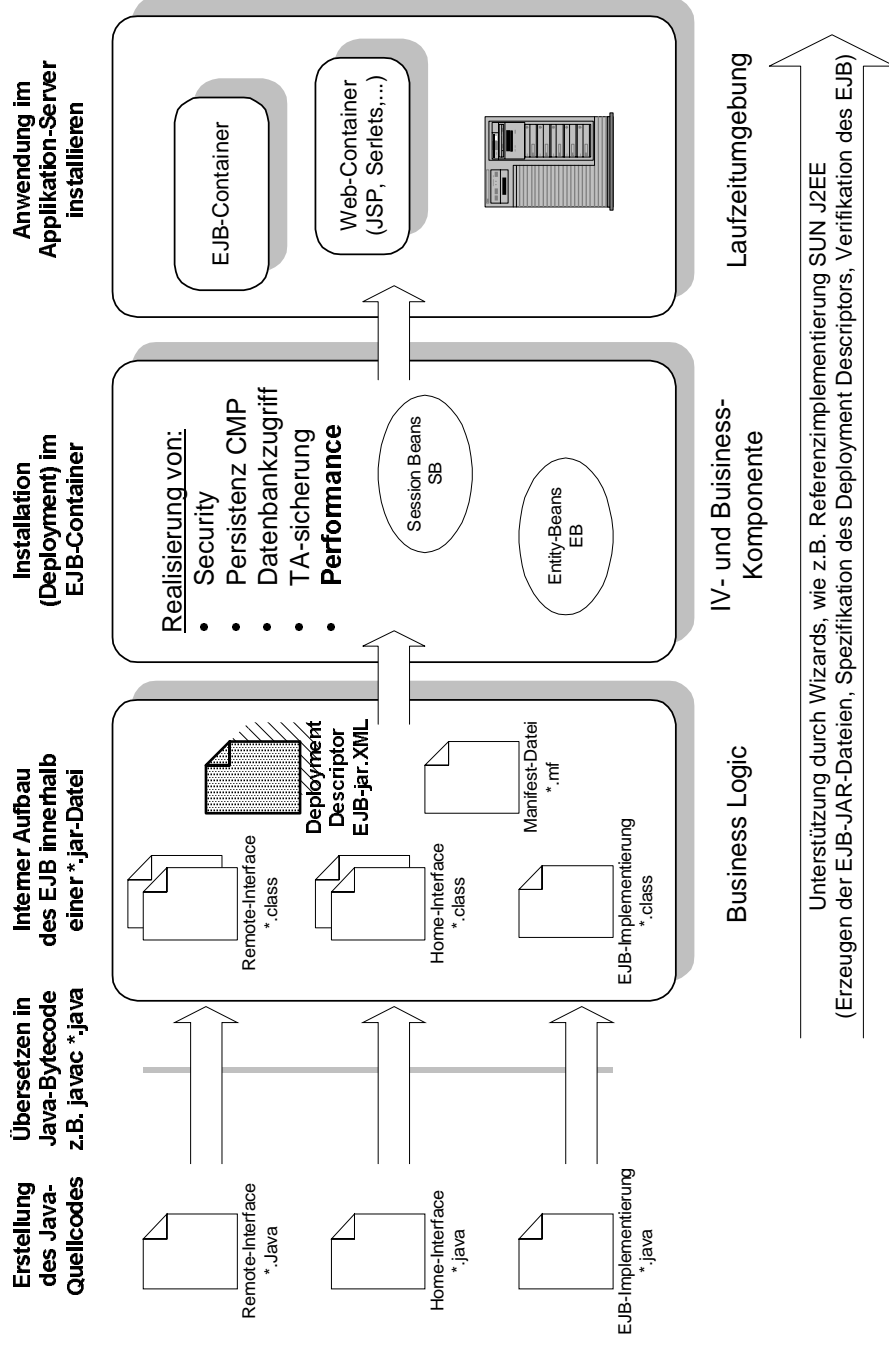


# Methodik der Performanceanalyse III

Determinierung des Ressourcenbedarfs :

- ◆ HW-Bezug schränkt die Anwendbarkeit ein
  - CPU-Auslastung
  - Speicherverbrauch
  - I/O-Durchsatz
- ◆ Alternative durch Bezug auf Standardbenchmarks
  - SPEC Standard Performance Evaluation Cooperation
    - SPECjvm98
    - SPECjbb2000
  - TPC Transaction Processing Council
    - TPC-W

# Erweiterter XML-DD für EJB's I



## Erweiterter XML-DD für EJB's II

Idee/Konzept:

- ◆ Deployment-Descriptor als „Beipackzettel“
  - Einführung eines <quality>-Tag
  - Bezug auf entsprechende Methode <method>-Tag
  - Abbildung der Qualitätsklassen durch <Qn>-Tag's
- ◆ Spezifizierung beim Deploy-Vorgang
  - Erweiterung der Wizard-Funktionalitäten



## Erweiterter XML-DD für EJB's III

### Vorteile:

- ◆ Komponenteninterne Beschreibung
- ◆ Funktionalitäten zu Laufzeit
  - Service-Level-Agreements
  - Accounting
  - Load-Balancing

### Voraussetzung:

- ◆ Berücksichtigung innerhalb der SUN-Spezifikation
- ◆ Umsetzung durch Container/Application-Server

# Erweiterter XML-DD für EJB's IV

```
<ejb-jar>
...
<quality>
  <RS>CPU: Risc P5C-233 SPEC:156</RS>
  <METHOD NAME="Kunde_hinzufügen">
    <Q1>...</Q1>
    <Q2>...</Q2>
    <Q3>
      <Q31>Response Time: 1.5s</Q31>
      <Q32>CPU %: 70%</Q32>
      <Q33>...</Q33>
    </Q3>
    <Q4>...</Q4>
    <Q5>...</Q5>
    <Q6>...</Q6>
  </METHOD>
</quality>
</enterprise-beans>
</ejb-jar>
```

# Zusammenfassung

- ◆ Komplexität der Performance-Spezifizierung
- ◆ ungelöste Problemstellung
  - im wissenschaftlichen- und
  - im technologischen Kontext
- ◆ Qualitätsklassen als Klassifizierungsrahmen
- ◆ Vision eines weiteren Auswahlkriterium
- ◆ Methodik der Performance Analyse
- ◆ Hinterlegung der Informationen am Bsp. von EJB's

## Ausblick I

- ◆ Prototypische Implementierung des Konzeptes
- ◆ Instrumentierung von Komponenten mittels ARM
  - Erfassung der kompletten Interaktionskette
- ◆ UML-Design inkl. Performance Annotationen
  - Komponenten-Diagramm
  - Interaktionsdiagramme (Abstraktion auf Komponenten)
  - Deployment-Diagramme (Interaktion mit Ressourcen)

## Ausblick II

- ◆ Modellbezogene Performance-Analyse
- ◆ Komponentenbezogene Performance-Patterns
  - Interaktion verschiedener Komponenten
  - Verbindungen zwischen Komponenten
    - Client-Seite
      - synchrone Kommunikation
      - asynchron Kommunikation
    - Server-Seite
      - Single-Threaded vs. Multi Threaded
      - Thread-Pool vs. Thread creation by request

## Weitere Informationen

- ◆ Andreas Schmietendorf
  - [schmiete@ivs.cs.uni-magdeburg.de](mailto:schmiete@ivs.cs.uni-magdeburg.de)
- oder
  - [a.schmietendorf@telekom.de](mailto:a.schmietendorf@telekom.de)
- ◆ André Scholz
  - [ascholz@iti.cs.uni-magdeburg.de](mailto:ascholz@iti.cs.uni-magdeburg.de)