Bundling auf Softwaremärkten
Eine technische Sicht

Michael Goedicke

Institute for Computer Science and Business Information Systems

University of Duisburg-Essen

Campus Essen

Technische – Softwarenentwickler – Sicht auf die folgenden Fragen

- Welche produktionsseitigen Besonderheiten bestehen bei der Entwicklung von Software?
- Ist der Trend zur Systemintegration zwingend mit verstärkter Bündelung bislang eigenständiger Programme verbunden?
- Lassen sich bestimmte Funktionen ohne weiteres aus Plattformprodukten herauslösen?
- Welche Ansprüche stellen Anwender an die Softwareindustrie?

Software ist ein immaterielles Gut mit einigen schwierigen Eigenschaften

Korrektheit	Verifizierbarkeit	Verstehbarkeit
Zuverlässsigkeit	Wartbarkeit	Interoperabilität
Robustheit	Wiederverwendbarkeit	Produktivität
Benutzerfreundlichkeit	Portierbarkeit	Pünktlichkeit

- Es ergeben sich besonders hohe Herausforderungen für die Entwicklung langlebiger Softwareprodukte und -Komponenten
- Wenn die Art der Benutzung bekannt ist kann man bei systematischem Vorgehen ist das Engineering „beherrschbar"

allerdings

- Produkte mit vielfältiger Funktionalität sind nur schwer in dieser Hinsicht beurteilbar (was ja auch gewünscht ist)

→ Der Schlüssel ist eine „saubere" und stabile Softwarearchitektur für Systeme und Komponenten

Die Softwarearchitektur definiert eine Blaupause für die Konstruktion des Systems

- Klare und präzise Schnittstellen
- Schnittstellen sollten „schmal" sein
- Klare und abgegrenzte Aufgaben für Komponenten, und Teilsysteme in der Architektur

Herausforderung: Software auf so einer Basis zu konstruieren, dass

- entstehende Produkte allgemein verwendbar sind
- die Produkte einfach mit anderen kombinierbar und erweiterbar sind
- in den geplanten Anwendungsszenarien korrekt, performant und zuverlässig funktionieren

Zwei Sichten des Softwareentwicklers

a) Konstrukteur einer Softwarekomponente / Softwareprodukt

b) Verwendung / Erweiterung einer Komponente / eines Systems

Jeder Entwickler nimmt immer beide Rollen an!

Die Rolle des Konstrukteurs

- Schnittstellen stellen die stabilen Teile eines Systems dar

  ⇒ Die Erfüllung von Vorgaben durch Schnittstellen kann die Freiheiten eines Entwicklers einschränken

- Über die Zeit gesehen werden Änderungen an Komponenten / Systemen erfolgen

  ⇒ Änderungswünsche einerseits und Stabilität von Schnittstellen können in Konkurrenz treten

  Allerdings: Die Softwaresysteme, die erfolgreich und langlebig waren besaßen in der Regel eine klare Softwarearchitektur mit solchen klar definierten Schnittstellen

Rolle des Verwenders von Software-Komponenten

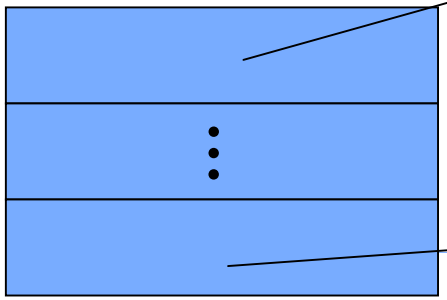
- Die eigene Entwicklung basiert auf einem fremden Produkt

  ⇒ Stabilität der Schnittstellen und Frameworks sehr wichtig

- Wünschenswert: Zugriff auf alle Details eines Systems

  ⇒ Kontrolle der Integrität wird mit zunehmendem Detailierungsgrad schwieriger

⇒ Zielkonflikte zwischen den beiden Rollen



Höhere Schichten: größere Komfort geringe Kontrolle über Details

Niedrige Schichten: geringer Komfort größere Kontrolle der Details

Zukünftige Entwicklungen (1)

- **Services** Serviceorientierte Architekturenn (SOA), Software as a Service (SaaS) …

  Charakteristik: die Schnittstellen werden noch wichtiger, da in der Regel die Realisierung des Service nicht mehr beim Anwender liegt

  Für den Konstrukteur wird die Flexibilität bei der Realisierung vielleicht sogar noch größer.

  Die Frage bleibt, was ist eine nützliche Schnittstelle

  Außerdem: je nach dem Grad der Öffentlichkeit der Schnittstelle(Service) wird die Forderung nach Stabilität eher größer

Zukünftige Entwicklungen (2)

- **Neue Hardwareentwicklungen**: das Mooresche Gesetz gilt in Teilen nicht mehr: die Erhöhung der Rechenleistung wird absehbar nur durch die **Erhöhung der Anzahl der Rechnerkerne** erzielt

  derzeitige Softwaresysteme sind selten auf diese Möglichkeiten eingestellt

  ⇒ ungenutzte Resourcen

  die Konstruktion angepasster Softwarearchitekturen wird erheblich komplexer werden

  ⇒ könnte einen Trend zu größeren Komponenten auslösen

Schlussfolgerungen

- Die technischen Risiken von Architektur-Entscheidungen sind nicht immer leicht zu bestimmen
- Es wird ein großer Bedarf herrschen, verschiedene Schnittstellen für ein Softwareprodukt öffentlich zugänglich zu machen
- Der Trend Software auf der Basis von Komponenten und Services zu Systemen zu konfigurieren wird sich weiter erhöhen

Die Fragen

- Welche produktionsseitigen Besonderheiten bestehen bei der Entwicklung von Software?

→ *Zielkonflikte zwischen den Rollen des Konstrukteurs*

- Ist der Trend zur Systemintegration zwingend mit verstärkter Bündelung bislang eigenständiger Programme verbunden?

→ *Vermutlich wird SOA etc. eher den Bedarf an Schnittstellen erhöhen*

- Lassen sich bestimmte Funktionen ohne weiteres aus Plattformprodukten herauslösen?

→ *Dies hängt stark von der Konstruktion und Architektur ab*

- Welche Ansprüche stellen Anwender an die Softwareindustrie?

→ *s.o. Mashup und vergleichbare Techniken erfordern viele kleine Services*

Herzlichen Dank für Ihre Aufmerksamkeit