

## Association for Information Systems AIS Electronic Library (AISeL)

---

Wirtschaftsinformatik Proceedings 2001

Wirtschaftsinformatik

---

September 2001

# Gibt es eine "Softwarekomponenten Industrie"? Ergebnisse einer empirischen Untersuchung

Andreas Dietzsch

*Technische Universität Dresden*, [andreas.dietzsch@wise.wiwi.tu-dresden.de](mailto:andreas.dietzsch@wise.wiwi.tu-dresden.de)

Werner Esswein

*Technische Universität Dresden*, [werner.esswein@wise.wiwi.tu-dresden.de](mailto:werner.esswein@wise.wiwi.tu-dresden.de)

Follow this and additional works at: <http://aisel.aisnet.org/wi2001>

---

### Recommended Citation

Dietzsch, Andreas and Esswein, Werner, "Gibt es eine "Softwarekomponenten Industrie"? Ergebnisse einer empirischen Untersuchung" (2001). *Wirtschaftsinformatik Proceedings 2001*. 50.

<http://aisel.aisnet.org/wi2001/50>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2001 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

In: Buhl, Hans Ulrich, u.a. (Hg.) 2001. *Information Age Economy*; 5. Internationale Tagung  
Wirtschaftsinformatik 2001. Heidelberg: Physica-Verlag

ISBN: 3-7908-1427-X

© Physica-Verlag Heidelberg 2001

# Gibt es eine „Softwarekomponenten Industrie“? Ergebnisse einer empirischen Untersuchung

**Andreas Dietzsch, Werner Esswein**

Technische Universität Dresden

*Zusammenfassung: Die Konstruktion von Softwaresystemen aus Komponenten wird bereits seit dem Ende der 1960er Jahre thematisiert. Den Ausgangspunkt dieser Diskussion bildete McIlroys Vision einer Softwarekomponenten fertigenden Industrie, die Integratoren die Komponenten für eine industrielle Softwareproduktion bereitstellt. Der vorliegende Beitrag stellt die Ergebnisse einer in Deutschland erhobenen Studie vor, deren Ziel es war zu untersuchen, ob sich ein Komponentenmarkt bereits etablieren konnte.*

*Schlüsselworte: Komponentenbasierte Softwareentwicklung, Componentware, Komponenten-Industrie*

## 1 Einleitung

„Software component technology is one of the most sought-after and at the same time least understood topics of the software field.“ [Szyp98, S. XIII]

Der Ausgangspunkt dieses Beitrags ist die von McIlroy formulierte Vision einer „Softwarekomponenten Industrie“, die eine wesentliche Voraussetzung für die Industrialisierung der Softwareentwicklung ist. Er beschreibt eine Trennung von Komponenten-Herstellern und –Integratoren. Für gleiche Aufgaben werden durch die Komponenten-Hersteller verschiedene Komponenten-Typen angeboten. Integratoren setzen Softwaresysteme aus diesen austauschbaren „Baugruppen“ zusammen [McIl76, S. 89].

Der vorliegende Beitrag untersucht, inwieweit sich ein Komponentenmarkt – die Voraussetzung für das Bestehen einer solchen Industrie – bereits etablieren konnte und welche Faktoren sich dabei fördernd oder behindernd auswirken. Dazu wird zunächst die in der Literatur beschriebene Sicht auf komponentenbasierte Softwareentwicklung beschrieben. Darauf aufbauend werden geeignete Fragestellungen abgeleitet. Zu deren Beantwortung wurde 1999/2000 eine Studie durchgeführt, deren Ergebnisse hier teilweise vorgestellt werden.

Die Untersuchung volks- oder betriebswirtschaftlicher Aspekte, wie z. B. die vorherrschende Wettbewerbsstruktur oder Geschäftsmodelle, sind nicht Gegenstand dieses Beitrags.

## 2 Komponentenbasierte Softwareentwicklung in der Theorie

**Wiederverwendung** ist das Nutzen existierender Artefakte bei der Entwicklung von Software [Lim98, S. 7]. Zu solchen Artefakten zählen wiederverwendbare Softwarekomponenten (reusable software components (RSC)). Dies sind Entwürfe, Code oder andere Ergebnisse des Prozesses der Softwareentwicklung, die zur Wiederverwendung bestimmt sind [NATO91, S. 3-1]. Im folgenden werden die Begriffe Softwarekomponente und Komponente synonym gebraucht.

### 2.1 Einführung

Auf der 1968er NATO Konferenz in Garmisch skizzierte *McIlroy* seine Vision einer auf „massenproduzierten“ Softwarekomponenten basierenden Softwareproduktion mit dem Hauptziel, wohlstrukturierte Systeme zu erstellen, die aus verständlichen und wiederverwendbaren Teilen bestehen [McIl76, S. 89ff]. Wie mit Bausteinen sollen dabei Softwaresysteme unter Nutzung von Komponenten zusammengesetzt werden.

Komponenten bilden damit den Kern kompositorischer Wiederverwendung. Bei dieser Entwicklungsstrategie werden kleine Teile als invariant betrachtet und durch variable Funktionalitäten zu einem System verbunden. Existierende Komponenten werden so als Bausteine für neue Systeme verwendet. Diese werden dann als **Componentware** bezeichnet. Eine solche Sicht auf die Konstruktion von Softwaresystemen ist ebenfalls bei Geschäftsobjekten, Komponenten-Frameworks und Prozeßbausteinen wiederzufinden [Grif99, S. 28; Pree99, S. 59; Krus96; Lang97; Remm97].

Die komponentenbasierte Entwicklung von Softwaresystemen setzt weit mehr als das Vorhandensein von Komponenten voraus. In Analogie zum allgemeinen Verständnis von Technologie [PlSa96, S. 7] ist **Komponententechnologie** die Art und Weise der Entwicklung von Softwaresystemen mit Komponenten und die Gesamtheit des dazu erforderlichen Wissens. Dieses umfaßt sowohl technische Aspekte, wie die Spezifikation des Aufbaus eines Softwaresystems aus Komponenten und des Zusammenwirkens der einzelnen Komponenten, als auch nicht-technische Aspekte, wie geeignete Organisationsformen und Wirtschaftlichkeitsmodelle. Die dabei diskutierten Fragestellungen werden nachfolgend dargestellt.

## 2.2 Technische Aspekte

### 2.2.1 Überblick

Eine einheitliche Definition von Softwarekomponenten fehlt bislang. Bei der Beschreibung von Merkmalen solcher Komponenten wird häufig deren Vorliegen als **Softwarestück** genannt, das **in binärer Form** existiert und mit Komponenten anderer Hersteller kombinierbar ist [Micr95, S. 11]. Allgemein sind Komponenten Softwarebausteine, die in ihrem Umfang zwischen kompletten Applikationen und einzelnen Klassen bzw. Prozeduren einzuordnen sind [Döme97, S. 11]. *Bronsard et al.* ergänzen diese Sicht um die Eigenschaft einer Komponente als „... a significant functional unit of a system.“ [BBKL97, S. 20] Sie betonen damit, daß es sich bei Komponenten sowohl um systemnahe als auch um an fachlichen Problemlösungen orientierte Elemente handelt.

Den Charakter von Komponenten als Wiederverwendungsansatz betont *Same-tinger*, der Softwarekomponenten als **in sich geschlossene, klar identifizierbare** Artefakte auffaßt, die spezifische Funktionen beschreiben oder ausführen und neben klaren Schnittstellen eine angemessene Dokumentation sowie einen definierten Wiederverwendungsstatus besitzen [Same96, S. 6].

Eine abgeschlossene Gestaltung ist Voraussetzung für die **unabhängige Verteilung** von Komponenten. Eine Komponente ist dann abgeschlossen, wenn deren Verwendung möglich ist, ohne daß das Vorhandensein anderer Komponenten vorausgesetzt wird [Same97, S. 68]. Die Definition einer oder mehrerer Schnittstellen, über die die Komponenten genutzt werden können, ermöglicht eine solche Gestaltung von Komponenten und damit die Kombination von Komponenten ohne Kenntnis über deren Umsetzung [EiNi99, S. 364ff; GrPf98, S. 38].

Damit wird die **Erstellung und Verteilung durch Dritte** möglich. Diesen Aspekt betont *Szyperski*. Er betrachtet Softwarekomponenten als „... a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.“ [Szyp98, S. 34]

Um Softwaresysteme mit Hilfe von Komponenten erstellen zu können, müssen diese interagieren können. Die **Interoperabilität** von Softwarekomponenten ist dazu über verschiedene Systemplattformen, Netzwerke, Programmiersprachen und Werkzeuge hinweg zu gewährleisten. Hieraus leitet sich die Notwendigkeit geeigneter Middleware und Komponentenplattformen ab. Neben Komponenten sind ebenfalls Architekturen, Skripte und „Klebstoff“ (glue) im Rahmen einer komponentenbasierten Softwareentwicklung anzuwenden. Als Klebstoff werden Abstraktionen bezeichnet, die notwendig sind, um Komponenten trotz inkompatibler Schnittstellen transparent miteinander zu verbinden [Schn99, S. 16f].

### 2.2.2 Abgeleitete Fragestellungen

Aus der Vielzahl der existierenden Definitionen leitet sich die Frage ab, ob die in der Literatur formulierten Eigenschaften von Komponenten, in der Praxis ebenfalls als charakteristisch angesehen werden. Weiter ist zu untersuchen, welche Technologien als komponentenorientiert in diesem Sinne gelten.

Bezogen auf heute verfügbare Technologien ist zu hinterfragen, ob deren Unterstützung der Verwaltung und Komposition von Komponenten mit den in der Literatur formulierten Vorstellungen korrespondiert.

Unter dem Aspekt einer „Softwarekomponenten Industrie“ ist festzustellen, in welcher Form die Weitergabe von Komponenten erfolgt. Insbesondere ist es von Interesse, ob Komponenten als eigenständige Produkte kommerziell verwertet oder als feste Bestandteile von Systemen ausschließlich mit diesen vermarktet werden.

## 2.3 Wirtschaftliche Aspekte

### 2.3.1 Überblick

Eine wesentliche Motivation zum Einsatz von Komponenten ist die Möglichkeit, mit dieser Form der Wiederverwendung die Produktivität der Entwicklung und die Qualität von Softwaresystemen zu erhöhen. Nach Meinung der Autoren ist die Erfüllung dieser Erwartungen wesentlich für das Etablieren komponentenbasierter Softwareentwicklung.

*Lim* untersucht die Effekte von Wiederverwendung in zwei Unternehmen und stellt die Ergebnisse seiner Studie in [Lim94] vor. Er stellt fest, daß die Entwicklungsdauer und Markteinführungszeit um 42% reduziert und die Produktivität der Softwareentwicklung zwischen 40% und 57% erhöht werden konnte. Daß die Qualität von Softwaresystemen durch die Verwendung ausgereifter Artefakte, wie z. B. Softwarekomponenten, steigt, bestätigt diese Untersuchung mit der Feststellung, daß die Fehlerrate der Systeme, die durch die untersuchten Firmen entwickelt wurden, um bis zu 51% sank [Lim94, S. 24]. Auch wenn diese Ergebnisse nicht unmittelbar auf komponentenbasierte Softwareentwicklung übertragen werden können, wird doch die Dimension der Potentiale von Wiederverwendung im allgemeinen und dem Einsatz von Komponenten im besonderen deutlich.

Neben Aufwandssenkungen bei der Erstellung von Softwaresystemen können ebenfalls die Wartungskosten gesenkt werden. So können fehlerhafte Systemteile ausgetauscht werden, ohne das Gesamtsystem neu übersetzen zu müssen. Darüber hinaus sind modular aufgebaute Softwaresysteme flexibel an veränderte oder neue Anforderungen anpaßbar.

Da Komponenten eine Form der Wiederverwendung sind, ist davon auszugehen, daß sich diese Erfahrungen in ihrer Grundaussage auch auf das Gebiet der komponentenbasierten Softwareentwicklung übertragen lassen.

Auch wird in der Literatur angeführt, daß die durch Wiederverwendung verursachten Kosten wesentliche wirtschaftliche Hindernisse bei der Einführung sein können. Zu diesen Kosten zählen z. B. Aufwendungen für die Entwicklung und den Einsatz wiederverwendbarer Artefakte sowie die Definition und Implementierung von Wiederverwendungsprozessen. Die Kosten der Erstellung wiederverwendbaren Codes können so zwischen 111% und 480% der Kosten von konventionellen Code liegen [Lim94, S. 26]. Daraus leitet sich die Erwartung ab, daß Wiederverwendung nur langfristig wirtschaftlich realisiert werden kann.

### **2.3.2 Abgeleitete Fragestellungen**

Überwiegend werden der Wiederverwendung im allgemeinen und damit ebenfalls dem Einsatz von Komponenten vielfältige wirtschaftliche Potentiale zugeschrieben. Um diese zu konkretisieren, sind die mit der Nutzung von Komponenten verbundenen Erwartungen zu hinterfragen. Dabei ist zwischen technischen und nicht-technischen Erwartungen zu unterscheiden.

Da die Entscheidung für oder gegen den Einsatz einer bestimmten Technologie nicht zuletzt unter Gesichtspunkten der Wirtschaftlichkeit zu treffen ist, muß festgestellt werden, welche Auswirkungen auf den Verlauf des Entwicklungsaufwands durch den Einsatz von Komponenten erwartet werden. Daneben sind weitere wirtschaftliche Erwartungen, wie z. B. die Verkürzung von Markteinführungszeiten, Qualitätssteigerungen und die Erleichterung des Wissenstransfers festzuhalten.

Mit Blick auf eine „Softwarekomponenten Industrie“ ist zu hinterfragen, wieviele der Unternehmen die Vermarktung von Komponenten als eigenes Geschäftsfeld ansehen bzw. entwickeln wollen.

## **3 Studie zur komponentenbasierten Softwareentwicklung**

### **3.1 Aufbau und Umfang der Studie**

Im Rahmen einer an der TU Dresden durchgeführten Studie wurden 2347 Unternehmen der Softwarebranche angeschrieben. Die Anzahl der Ausfälle, z. B. durch falsche Adressierung, lag bei 192. Von den verbleibenden 2155 Angeschriebenen gab es 71 Rückmeldungen, was bei dem beschriebenen Umfang der Stichprobe

einer Ausschöpfungsquote von 91,82% und einer Rücklaufquote von 3,29% entspricht. Zur Erhebung wurde ein WWW-basierter Fragebogen genutzt, in dessen Abschnitten Fragen zu existierenden Technologien, Methoden, Programmiersprachen, Gründen und Auswirkungen des Einsatzes von Komponententechnologien, Potentialen des Komponentenansatzes sowie zu demoskopischen Merkmalen zu beantworten waren.

Aufgrund des Mangels an empirischen Untersuchungen auf diesem Gebiet trägt diese Studie explorativen Charakter. Nachfolgend werden ausgewählte Ergebnisse der Umfrage dargestellt.

## **3.2 Ergebnisse der Studie**

### **3.2.1 Charakterisierung der Unternehmen**

Die Untersuchung der entwickelnden Unternehmen war nicht der Schwerpunkt der Untersuchung. Dennoch ist es als überraschend zu bezeichnen, daß ein Zusammenhang zwischen dem Grad der Nutzung von Komponenten und der Anzahl der bearbeiteten Entwicklungsprojekte nicht bestätigt werden konnte. Die Hypothese, daß zwischen dem Zielbereich der entwickelten Software (z. B. Industrie, Verwaltung, Medizin, Militär) und dem Grad des Einsatzes von Komponenten ein Zusammenhang existiert wurde, ebenfalls verworfen.

### **3.2.2 Das Komponentenverständnis**

In Abschnitt 2.2 wurde gezeigt, daß die Literatur keine gemeinsame Definition des Komponentenbegriffs bereitstellt. Aus diesem Grund wurde untersucht, welche Eigenschaften mit Komponenten assoziiert werden [Abbildung 1]. Mehrfachantworten waren dabei möglich.

Als charakteristisches Merkmal wurde der Besitz eigener Ressourcen und einer Schnittstelle am häufigsten (73%) genannt. Ebenfalls als charakteristisch, jedoch mit deutlichem Abstand, werden das Lösen eines Problems und die Möglichkeit der unabhängigen Verteilung angesehen. Daraus kann abgeleitet werden, daß das Komponentenverständnis sowohl systemnahe Komponenten (z. B. für grafische Benutzerschnittstellen) als auch solche Bausteine einschließt, die Lösung fachlicher Probleme implementieren.



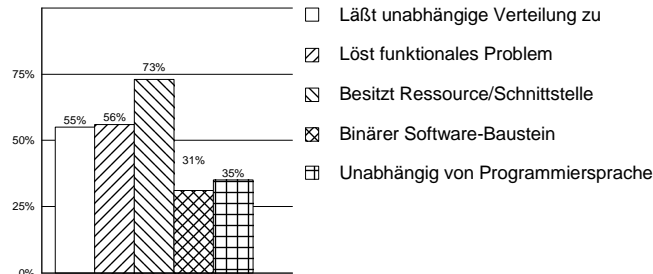


Abbildung 1: Eigenschaften von Komponenten

Daß eine Komponente in binärer Form vorliegt und unabhängig von einer bestimmten Programmiersprache ist, stellt, abweichend von den in der Literatur formulierten Anforderungen, nur für 31% bzw. 35% der Befragten ein charakteristisches Merkmal dar.

Der zentrale Aspekt einer Komponente ist damit deren Abgeschlossenheit, insbesondere in funktionaler Hinsicht. Dagegen wird der Aspekt des „Plug-and-Play“ von Komponenten [MeLi98; SML99], die binär codiert sind und unabhängig von einer spezifischen Programmiersprache in ein System integriert werden können, (noch) nicht als zentrales Problem angesehen.

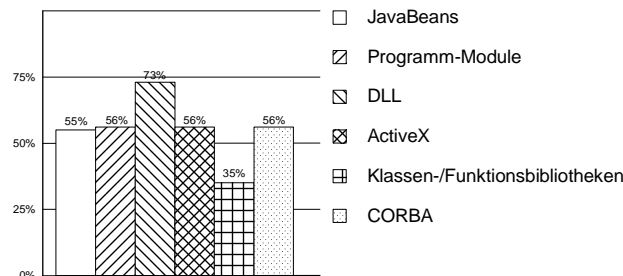


Abbildung 2: Komponententechnologien

In diesem Kontext wurde ebenfalls untersucht, welche Produkte durch die Befragten als komponentenorientiert aufgefaßt werden [Abbildung 2]. Insbesondere im Vergleich zu JavaBeans, ActiveX und CORBA ist hier die überdurchschnittlich häufige Nennung von Dynamic Link Libraries (DLL) interessant. Gängige Komponententechnologien werden in den dargestellten Eigenschaften ausgeglichen bewertet. Die Abgrenzung zu Klassen- und Funktionsbibliotheken ist deutlich. Überraschend ist bei diesen jedoch die insgesamt hohe Häufigkeit der Nennung als komponentenorientierte Technologie.

### 3.2.3 Komponentenangebot

Lediglich 20% der befragten Unternehmen an, ihre Komponenten außerhalb eigener Projekte zu verwerten. Auf eine direkte kommerzielle Verwertung von Komponenten wird weitgehend verzichtet. Es wurde festgestellt, daß Komponenten überwiegend gebündelt mit eigenen oder fremden Produkten vertrieben werden [Abbildung 3]. Die Vermarktung von Komponenten als selbständiges Geschäftsfeld aufzubauen, planen 27% der Unternehmen. Von diesen sehen jedoch nur 40% darin einen Kernbereich zukünftiger Geschäftstätigkeit. Aufgrund dieses Mangels an Komponenten-Herstellern kann vermutet werden, daß sich noch kein funktionierender Markt für Komponenten etablieren konnte.

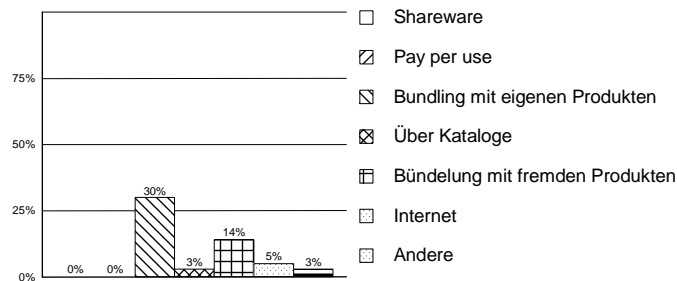


Abbildung 3: Distribution erstellter Komponenten

Deshalb wurde weiterhin untersucht, ob die Nachfrage nach Komponenten bzw. die in deren Einsatz gesetzten Erwartungen die Vermutung rechtfertigen, daß eine erfolgreiche Etablierung eines Marktes für Komponenten möglich ist.

### 3.2.4 Nachfrager von Komponenten

Annähernd 57% der befragten Unternehmen gaben an, Komponenten von Fremdfirmen zu beziehen. Gleichzeitig wird das Angebot an Komponenten von 65% der Unternehmen jedoch als zu gering bewertet. Die technischen und wirtschaftlichen Erwartungen, die mit dem Einsatz von Komponenten verknüpft sind, geben Abbildung 4 und Abbildung 5 wieder. Mehrfachantworten waren möglich.

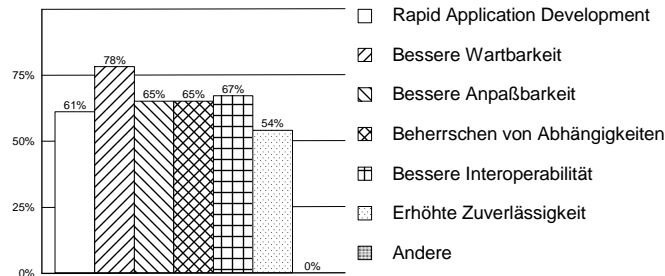


Abbildung 4: Technische Erwartungen an den Einsatz von Komponenten

Die dargestellten Erwartungen stehen nicht im Widerspruch zu den in der Literatur beschriebenen Potentialen komponentenbasierter Softwareentwicklung. Als Indiz für den Zustand heutiger Systeme mögen die mit 78% und 67% am häufigsten genannten Erwartungen gesehen werden, die Wartbarkeit könne durch den Einsatz von Komponenten verbessert und Abhängigkeiten in den Systemen besser beherrscht werden.

Neben diesen Erwartungen an technische Effekte wurden ebenfalls die vermuteten wirtschaftlichen Potentiale des Einsatzes von Komponenten hinterfragt [Abbildung 5]. In einem engen Zusammenhang sind hier die Nennung von einer kürzeren Markteinführungszeit sowie einer gesteigerten Produktivität zu sehen. Daneben sind die Verbesserung der Produktqualität und die Senkung der Entwicklungskosten häufig genannte Erwartungen. In diesem Zusammenhang ist die geringe Häufigkeit der Nennung der Bereiche „economies of scale“ sowie „Differenzierungspotential“ auffällig.

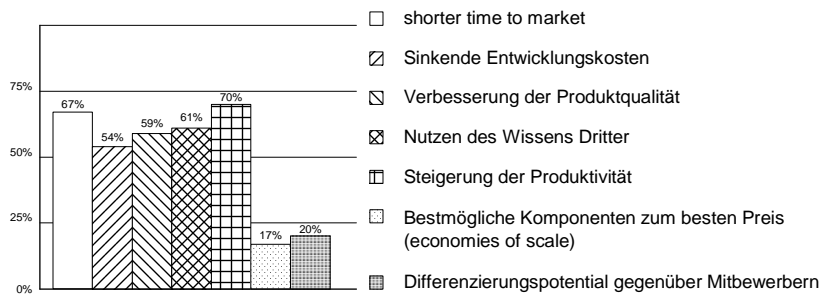


Abbildung 5: Ausgewählte wirtschaftliche Erwartungen an den Einsatz von Komponenten

Für das Etablieren von Komponententechnologien ist der erwartete Verlauf des Entwicklungsaufwands von entscheidender Bedeutung. Dabei wird bei 48% der befragten Unternehmen von einer Steigerung des Aufwands ausgegangen, 37%

erwarten, daß dieser sinkt und 15% rechnen nicht mit einer Veränderung des Entwicklungsaufwands.

Als Gründe eines steigenden Aufwands werden vor allem die Anlaufinvestitionen genannt, also die Schaffung einer Infrastruktur für den Einsatz von Komponenten und deren Wiederverwendung. Deutlich wird aber auch, daß durch die Einführung der Komponententechnologie eine Änderung im Entwurf der Software und im Ablauf ihrer Entwicklung erwartet wird [Abbildung 6].

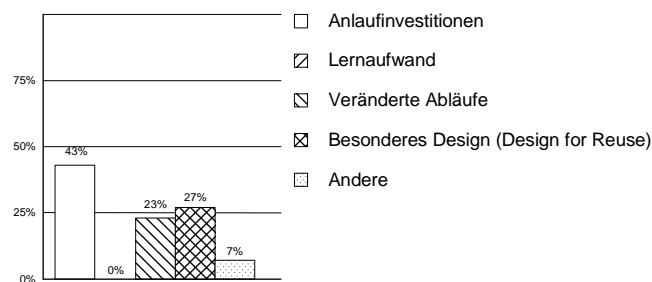


Abbildung 6: Gründe für die erwartete Zunahme des Entwicklungsaufwands

Daß nicht mit einem steigenden Lernaufwand gerechnet wird, da diesem keine Auswirkungen auf einen steigenden Aufwand zugesprochen werden, ist als interessantes Phänomen zu werten, dessen Klärung auf Basis des zugrundeliegenden Datenmaterials jedoch nicht möglich ist.

15% der befragten Unternehmen gehen von einem gleichbleibenden Entwicklungsaufwand aus, was im wesentlichen damit begründet wird, daß Komponententechnologie „... nichts substantiell Neues ...“ sei (67%).

Kostensenkende Effekte nach einer Etablierung komponentenbasierter Softwareentwicklung werden von 37% der befragten Unternehmen erwartet. Als Gründe werden die Möglichkeit zum „Nachnutzen bestehender Lösungen“ (52%) sowie die „Vereinfachung der Entwicklung“ (43%) genannt. Neben der Wiederverwendung wird also ein wesentlicher Effekt von Komponenten in der Kapselung der Komplexität der bereitgestellten Lösungen gesehen.

### 3.2.5 Erfahrungen beim Einsatz von Komponenten

Die Unternehmen wurden ebenfalls danach befragt, welche Erfahrungen beim Einsatz von Komponententechnologien bereits gesammelt wurde. Insbesondere wurde untersucht, in welchem Umfang die mit diesem Einsatz verbundenen Erwartungen erfüllt wurden. Dabei war festzustellen, daß die positiven Antworten überwiegen. So die Erwartungen des überwiegenden Teils der Befragten (83%) „voll“ (2%), „gut“ (33%) oder „im allgemeinen erfüllt“ (48%). Für alle Erwartungen, die nicht mindestens „gut“ erfüllt wurden, sollte die getroffene Einschätzung

begründet werden. Als wesentlich wurde dabei die ungenügende Reife der verfügbaren Komponententechnologien angegeben. Als weitere Ursachen werden jedoch auch ein unzureichendes technologisches Verständnis der Entwickler sowie das Fehlen geeigneter Werkzeuge angeführt [Abbildung 7].

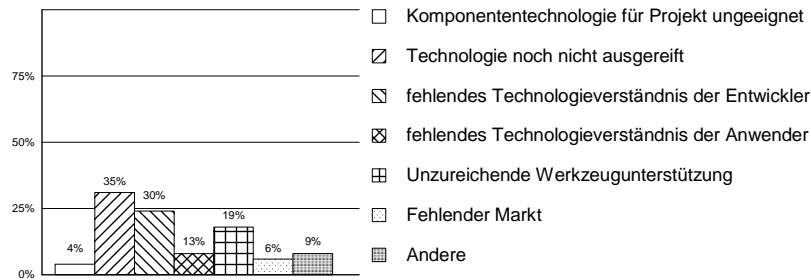


Abbildung 7: Gründe für das Nichterfüllen der Erwartungen

Die Möglichkeit der Verbesserung der Qualität und das mehrmalige Verwenden existierender Lösungen gelten neben der Steigerung der Produktivität als Haupteffekte der Wiederverwendung [Abschnitt 2.3]. Für die Befragung wurden diese Merkmale als „Einfachheit der Lösung“, „Wiederverwendung“ und „Qualitätssteigerung“ abgebildet. Die Beantwortung erfolgte in Verbindung mit der Frage nach der Nichterfüllung der Erwartungen.

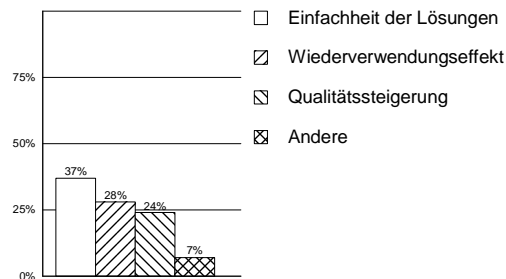


Abbildung 8: Spezifische Bereiche, in denen Erwartungen nicht erfüllt wurden

Es wird deutlich, daß die Erwartungen in diesen spezifischen Bereichen in erheblichem Umfang nicht erfüllt wurden. Da Komponenten auf die Wiederverwendung bei der Softwareentwicklung fokussieren, ist besonders hervorzuheben, daß annähernd 30% der Befragten ihre **Erwartungen an die Wiederverwendbarkeit** durch den Einsatz von Komponenten **nicht erfüllt** sahen [Abbildung 8].

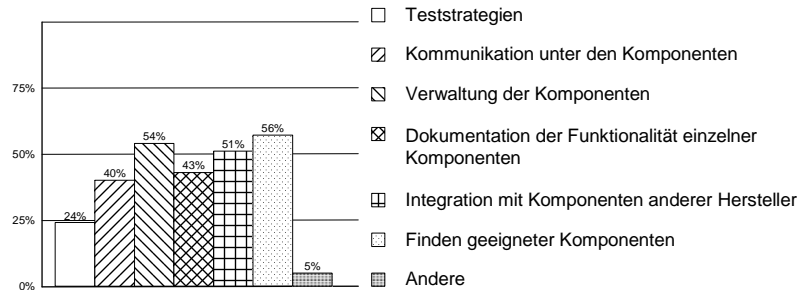


Abbildung 9: Probleme beim Projekteinsatz von Komponenten

Aus Abbildung 9 geht hervor, daß die Organisation von Komponenten, deren ausreichende Dokumentation sowie eine unzureichende Interoperabilität als Hauptursachen für Probleme beim Einsatz von Komponenten in Entwicklungsprojekten angesehen werden. An dieser Stelle beschränkt sich das Technologieverständnis der Anbieter offensichtlich auf die Bereitstellungen von Techniken und Werkzeugen zur Kombination von Komponenten innerhalb eines spezifischen Systems.

## 4 Zusammenfassung

Mit den Ergebnissen der in diesem Beitrag vorgestellten Studie konnte ein Eindruck vom Stand der Entwicklung einer „Softwarekomponenten Industrie“ in Deutschland gewonnen werden. Der dargestellte Rücklauf läßt jedoch keine Interpretation der Ergebnisse als repräsentativ zu.

Als Ergebnis der Befragung ist festzustellen, daß das in der Praxis anzutreffende Komponentenverständnis weitgehend mit in der Literatur vertretenen Meinungen korrespondiert. Auch die Erwartungen an den Einsatz von Komponenten entsprechen in technischer und wirtschaftlicher Hinsicht den in der Literatur dargestellten Potentialen. Es muß jedoch auch festgestellt werden, daß Erwartungen dort in erheblichem Umfang nicht erfüllt wurden, wo Komponenten mit dem Ziel der Wiederverwendung und Qualitätssteigerung eingesetzt wurden. Als Gründe sind das Fehlen von Infrastrukturen für den Einsatz von Komponentenansätzen zu nennen. Dazu zählt insbesondere die Unterstützung der Entwickler durch geeignete Entwicklungsmethoden und Werkzeuge.

Die Ergebnisse lassen damit den Schluß zu, daß trotz der Vielzahl an theoretischen Überlegungen zum Thema komponentenbasierter Softwareentwicklung, diese Technologie noch nicht die Reife für einen breiten und erfolgreichen Einsatz in der Praxis besitzt. Die Probleme, die im Einsatz von Komponenten innerhalb

konkreter Projekte auftraten, lassen den Schluß zu, daß die heute am Markt verfügbaren Systeme die Interoperabilität von Komponenten verschiedener Hersteller nicht ausreichend unterstützen. Aus dieser Sicht ist die abwartende Haltung auf Seiten potentieller Anbieter verständlich, die das Feld einer reinen Komponentenproduktion nur zögernd betreten. Von einer funktionierenden „Softwarekomponenten Industrie“ kann deshalb zum jetzigen Zeitpunkt nicht gesprochen werden.

## Literatur

- [BRSV98] Bergner, K.; Rausch, A.; Sihling, M.; Vilbig, A.: A Componentware Methodology based on Process Patterns. <http://www.forsoft.de/publikationen/TUM-I9823.ps>, 1998, Abruf am 2000-11-23.
- [BRSV99] Bergner, K.; Rausch, A.; Sihling, M.; Vilbig, A.: Componentware - Methodology and Process. <http://www.forsoft.de/publikationen/BRSV99c.ps>, 1999, Abruf am 2000-12-07.
- [BBKL97] Bronsard, F.; Bryan, D.; Kozaczynski, W.; Liongosari, E. S.; Ning, J. Q.; Olafsson, A.; Wetterstrand, J. W.: Toward Software Plug-and-Play. In: SSR '97. Proceedings of the 1997 Symposium on Software Reusability, May 17-19, 1997, Boston, MA. ACM Press, New York, 1997, S. 19-29.
- [Döme97] Dömer, F.: Componentware - Auf ins Legoland. In: Diebold Management Report (1997) 1, S. 11-14.
- [EiNi99] Eicker, S.; Nietsch, M.: Standards zum objektorientierten Paradigma. In: Wirtschaftsinformatik 41 (1999) 4, S. 358-370.
- [Grif98] Griffel, F.: Componentware: Konzepte und Technologien eines Softwareparadigma. dpunkt-Verlag, Heidelberg, 1998.
- [Grif99] Griffel, F.: Komponenten - Softwarebausteine des nächsten Jahrtausends? In: OBJEKTSpektrum, 6 (1999) 1, S. 24-29.
- [GrPf98] Gruntz, D.; Pfister, C.: Komponentensoftware und ihre speziellen Anforderungen an Standards. In: OBJEKTSpektrum 5 (1998) 4, S. 38-42.
- [HoMi93] Hochmüller, E.; Mittermeir, R.: Rahmenbedingungen für erfolgreiches Software-Reuse. <http://www.ifi.uni-klu.ac.at/Publications/pubfiles/psgzfiles/1993-02-HoMi.ps.gz>, 1993, Abruf am 2000-08-23.
- [JGJ97] Jacobson, I.; Griss, M.; Jonsson, P.: Software Reuse - Architecture, Process and Organization for Business Success. Addison-Wesley, Harlow et al., 1997.
- [Krus96] Kruse, C.: Referenzmodellgestütztes Geschäftsprozeßmanagement: Ein Ansatz zur prozessorientierten Gestaltung vertriebslogistischer Systeme. Gabler, Wiesbaden, 1996.
- [Lang97] Lang, K.: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen. Deutscher Universitätsverlag, Wiesbaden, 1997.

- [Lim94] Lim, C. W.: Effects of Reuse on Quality, Productivity and Economics. In: IEEE Software, 11 (1994) 5, S. 23-30.
- [Lim98] Lim, W. C.: Managing Software Reuse. Prentice-Hall, Upper Saddle River, 1998.
- [McIl76] McIlroy, M. D.: „Mass Produced“ Software Components. In: Naur, P.; Randell, B.; Buxton, J. N. (Hrsg.): Software engineering: concepts and techniques, Proceedings of the NATO conferences. Mason/Charter, New York, 1976, S. 88-95.
- [MeLi98] Mezini, M.; Lieberherr, K.: Adaptive Plug-and-Play Components for Evolutionary Software Development. <ftp://ftp.ccs.neu.edu/pub/people/lieber/appcs.pdf>, 1998, Abruf am 2000-12-15.
- [Micr95] o.V.: The Component Object Model Specification, Draft Version 0.9. <http://www.microsoft.com/Com/resources/comdocs.asp>, 1995, Abruf am 1999-07-29.
- [NATO91] o.V.: NATO Standard for Management of a Reusable Software Component Library. NATO Communications and Information Systems Agency, 1991.
- [PISa96] Pleschak, F.; Sabisch, H.: Innovationsmanagement. Schäffer-Poeschel, Stuttgart, 1996.
- [Pree99] Pree, W.: Wiederverwendung von Architekturen durch Frameworks. In: Computerwoche, (1999) 20, S. 58-60.
- [Remm97] Remme, M.: Konstruktion von Geschäftsprozessen: Ein modellgestützter Ansatz durch Montage generischer Prozeßartikel. Gabler, Wiesbaden, 1997.
- [Same96] Sametinger, J.: Classification of Composition and Interoperation. Poster Presentation at OOPSLA '96. <http://sydney.swe.uni-linz.ac.at/Staff/JohannesSametinger/papers/96.oopsla.long.pdf>, 1996, Abruf am 1999-07-29.
- [Same97] Sametinger, J.: Software Engineering with Reusable Components. Springer, Berlin et al., 1997.
- [SML99] Seiter, L.; Mezini, M.; Lieberherr, K.: Dynamic Component Gluing. <ftp://ftp.ccs.neu.edu/pub/people/lieber/DynCompGlue/DynCompGlue.pdf>, 1999, Abruf am 2000-12-15.
- [Szyp98] Szyperski, C.: Component software: beyond object-oriented programming. Addison-Wesley, Harlow et al., 1998.
- [Schn99] Schneider, J.-G.: Components, Scripts, and Glue: A conceptual framework for software composition. Dissertation an der Universität Bern, Philosophisch-naturwissenschaftliche Fakultät, 1999.