

# OBJEKTSpektrum

Die Zeitschrift für Software-Engineering und -Management

- **Weshalb Applikationen modernisieren?**
- **Strukturierte Analyse zur Optimierung von Anwendungsportfolios**
- **Wie wichtig ist der Code?**
- **Steuerungsoptionen für die Entwicklung im Team**
- **Wie migriere ich ohne Gefahr für das Alltagsgeschäft?**

NOCH NIE WAR AUSWANDERN SCHÖNER:

**WIR BRINGEN SIE NACH JAVA – MIT IBM EGL.**



 **PKS**  
BUILDING BRIDGES

## IHR PERSÖNLICHER AUSWANDERUNGSTEST:

- Ihre User wollen eine moderne Benutzeroberfläche.
- Ihre Windows®- und Web-Anwendungen wollen sich integrieren.
- Ihre Legacy-Anwendungen wollen hierbleiben.
- Ihr Know-how will seinen Wert behalten.
- Ihr Betrieb soll beim Migrieren einfach weiterlaufen.

## AUSWERTUNG

Schon bei zwei Kreuzchen lohnt sich ein Gespräch mit uns: 0751 56140229

### Tipp für Skeptiker:

Lesen Sie die Berichte von Auswanderern auf [www.pks.de](http://www.pks.de)



IBM® **Rational**® software

**PKS Software GmbH**

Georgstraße 15, D-88214 Ravensburg

Tel: +49 (0) 751 56140-0, [www.pks.de](http://www.pks.de)

<b>inhaltsverzeichnis</b> .....	<b>3</b>	Nachhaltig modern .....	<b>23</b>
<b>editorial</b>		Mit Anwendungsmodernisierung zu mehr Datenqualität .....	<b>24</b>
Modernisierung von Legacy Systemen .....	<b>4</b>	Mit Jazz und Rational Team Concert zum Dream Team werden .....	<b>26</b>
<b>einführung</b>		<b>migration &amp; consolidation</b>	
Anwendungsmodernisierung – Motivation und Positionierung .....	<b>5</b>	Application Modernization by Migration .....	<b>27</b>
<b>application portfolio</b>		Dynamische Software-Infrastrukturen auf Basis agiler IBM Mainframe Lösungen .....	<b>28</b>
Wie optimieren Sie Ihr Anwendungs- portfolio? .....	<b>6</b>	Migrationen ohne Risiko für das Business .....	<b>31</b>
<b>application renovation</b>		Impressum .....	<b>32</b>
Modernisierung von IT-Applikationen – Wege zum Erfolg .....	<b>10</b>	<b>education &amp; skill enablement</b>	
Die Wahrheit liegt im Code .....	<b>11</b>	Software-Evolution in Forschung und Praxis .....	<b>33</b>
Flexible Modernisierung von Anwendungen mit ARC .....	<b>12</b>	Mainframeausbildung im Fokus – 600 Universitäten sind weltweit dabei .....	<b>34</b>
Abhängigkeiten erkennen – Codeanalyse mit Rational Asset Analyzer .....	<b>14</b>	<b>Anzeigen:</b>	
Save Our Assets – Alte Systeme im neuen Gewand .....	<b>15</b>	IBM Deutschland GmbH .....	<b>36</b>
HATS – zwei Funktionen unter einem Hut .....	<b>18</b>	PKS GmbH .....	<b>2</b>
Genauso wie früher, nur besser .....	<b>19</b>	Q four GmbH .....	<b>22</b>
Ein neues Gesicht für die Host- Anwendungsentwicklung .....	<b>21</b>	Quintec GmbH .....	<b>9</b>
		SIGS DATACOM GmbH .....	<b>13</b>
		SoforTe GmbH .....	<b>29</b>
		Tricept GmbH .....	<b>35</b>

## Liebe Leserinnen und Leser,

Anwendungsmodernisierung wird oft mit dem Kunstwort „Software-Archäologie“ gleich gesetzt. Mit virtuellem Pinsel und Pinzette stößt der Entwickler dabei durch die verstaubten Programmzeilen und versucht so die ehemals strahlende Anwendung wieder in das rechte Licht zu rücken.

Genau diese Sichtweise wollen wir in der Ihnen vorliegenden Sonderbeilage mit dem Themenschwerpunkt „Business Application Modernization“ korrigieren. Heute stehen uns vielfältige Methoden und Werkzeuge zur Verfügung, die die Anwendungsmodernisierung effizient und kostengünstig machen.

Wir wünschen Ihnen beim Lesen der Fachbeiträge viel Freude und hoffen Ihnen damit eine neue Perspektive auf das Thema zu eröffnen!



**Ihr Markus Weyerhäuser**  
Leiter IBM GBS Kompetenzzentrum  
für Anwendungsmodernisierung  
(markus.weyerhaeuser@de.ibm.com)



**Ihre Susanne Herl**  
Redaktionsleitung OBJEKtspektrum  
(susanne.herl@sigs-datacom.de)

PS.: Wenn wir Ihnen mit diesem Sonderheft nicht alle Ihre Fragen beantwortet haben sollten, dann lassen Sie es uns gerne wissen!

Unter [www.ibm.com/software/de/rational/solutions/em](http://www.ibm.com/software/de/rational/solutions/em) können Sie weitere Informationen abrufen.

# Modernisierung von Legacy Systemen – Statusaufnahme und Handlungsempfehlungen

*Dieses Sonderheft ist der Thematik Business Application Modernization gewidmet. Es geht also darum „alte“ Fachanwendungen zu modernisieren. Eine „alte“ (oder Legacy) Anwendung ist dadurch gekennzeichnet, dass sie historisch gewachsen ist und eine meist unzureichende technische Dokumentation hat. Im Rahmen einer Studie haben wir im letzten Jahr die Landschaft solcher Systeme im deutschsprachigen Bereich (DACH) analysiert. Die Studie wurden unter den Namen „Strategische Planung und Einsatz von Legacy Systemen“ in Dezember 2009 veröffentlicht und hatte das Ziel, die Diskrepanz zwischen strategischer IT-Ausrichtung, dem Einsatz von Legacy Systemen und den dafür notwendigen Maßnahmen, insbesondere im Bereich der Personalentwicklung, nachzuweisen. Wir wollten auch aufzeigen, dass diese Problematik in Verbindung mit einem fehlenden Lehrangebot von Hochschulen mittel- bis langfristig zu unternehmensgefährdenden Problemen und zu weiter steigenden Kosten führen wird. Nicht zuletzt sollte die Studie auch das hohe Risikopotential bei der aktuellen Altersstruktur der Legacy-Experten im Unternehmen evaluieren und die Thematik der Legacy Systeme als eine aktuelle Problemstellung der Informatik und der Wirtschaftsinformatik wieder in den Vordergrund stellen.*

Bei der Analyse des universitären Lehrangebots im Bereich Legacy Systeme und Sprachen haben wir festgestellt, dass nur 17 von 198 deutschen Hochschulen Legacy-System- bzw. Mainframe-Lehrinhalte vermitteln. Hierbei legen die wenigen Lehrangebote primär ihren Schwerpunkt auf die Wartung und Administration von Mainframe-Technologien, aber weniger auf die Software-Sanierung und Programmierung.

Bei der Analyse der personellen Ressourcen für die Wartung und Weiterentwicklung von Legacy Systemen haben wir festgestellt, dass nahezu 80 % der Unternehmen Probleme mit ihren personellen Ressourcen haben und über 50 % keine spezifischen Maßnahmen ergreifen bzw. sich nur auf externe Ressourcen verlassen. Somit wurde eine der wichtigsten Annahmen der Studie bestätigt und das Risikopotential der aktuellen Situation aufgezeigt.

Unsere Studie ergab, dass Legacy Systeme in allen Anwendungsbereichen wie Vertrieb, Finanz- und Rechnungswesen, Personalwirtschaft und Produktion nahezu gleich vertreten sind. Das bedeutet, dass sich die aufgezeigten Risikopotentiale nicht auf einige wenige Kernaufgaben beziehen, sondern fast jeden Ablauf in einem Unternehmen infrage stellen können. Schwerpunktmäßig werden Legacy Systeme mehrmals in der Woche eingesetzt, somit rückt hier der Usability-Aspekt in den Vordergrund, der bei einer weiterführenden Untersuchung Optimierungspotentiale freigeben könnte.

Bei der eigenen Beurteilung von Legacy Systemen durch die Unternehmen selbst fällt besonders auf, dass die Modularität solcher Systeme als schlecht bis mangelhaft bewertet wurde. Vor dem Hintergrund der häufigen Integration von Legacy Systemen über Service-orientierte Architekturen (SOA) wird das Ausmaß dieser Problematik deutlich, die eine erfolgreiche Einbindung über SOA verhindert. Innerhalb von Integrations- und Modernisierungsprojekten sollte somit der Fokus auf diese nachträgliche Modularisierung gelegt werden.

Bei der Verwendung von Legacy Systemen treten insbesondere drei Probleme in den Vordergrund: Der Schulungsaufwand, die Nutzung von Insellösungen und die schlechten Schnittstellen. Diese Probleme werden mehrheitlich als Gründe angegeben, die die Nutzung der Legacy Systeme gänzlich verhindern oder erschweren.

Nahezu die Hälfte der Unternehmen sieht Legacy Systeme für den aktuellen Unternehmenserfolg als wichtig an. Bei der Frage, ob die Informatikstrategie der Unternehmen Vorgaben für den zukünftigen Einsatz von Legacy Systemen enthält, gaben über 30 % an, dass dort keine Vorgaben spezifiziert sind. Weitere 5 % gaben an, dass Legacy Systeme langfristig weiterhin Bestandteil ihrer IT bleiben werden.

Über 60 % der Unternehmen planen in den nächsten Jahren eine umfassende Modernisierung ihrer Legacy Systeme. Dieses Sonderheft soll dabei als Leitfaden dienen. Die Kernaufgaben eines IT-Verantwortlichen im Kontext solcher Modernisierungen müssen also in zwei Themenbereichen betrachtet werden – personell und technologisch. Auf der personellen Seite gilt es die Ausbildungslücke zu schließen. Kurzfristig kann dies nur unter Einsatz externer Kompetenz im Bereich Legacy Systeme und Integrationstechnologien erfolgen. Geeignete Hilfe kann hier im Rahmen von Kooperationsprojekten mit Technologieberatungen, Universitäten und Forschungseinrichtungen bereitgestellt werden. Mittel- und langfristig sollten durch gemeinsame Ausbildungsinitiativen zwischen Industrie und Hochschulen qualifizierte Hochschulabsolventen in ausreichender Zahl ausgebildet werden.

Die technologische Lücke kann nur durch ein verstärktes Engagement in Forschungs- und Entwicklungsaktivitäten geschlossen werden. Die aktuellen Erkenntnisse der Wissenschaft im Bereich SOA und „Internet der Dienste“ sollen besser auf Legacy Systeme übertragbar sein. Hier sind auch Kooperationen zwischen Industrie und Hochschulen ein geeignetes Mittel solche Themen anzugehen. Dabei ist auch die Politik gefragt mit geeigneten Förderschwerpunkten im Kontext des „Internet der Dienste“ Zeichen zu setzen.

Wir wünschen Ihnen viel Spaß beim Lesen! ■

**Prof. Dr. Hermann Krallmann**

Fachgebiet Systemanalyse, Technische Universität Berlin  
(hkr@sysedv.tu-berlin.de)

**Dr. Vladimir Stantchev**

Fachgebiet Systemanalyse, Technische Universität Berlin  
(vladimir.stantchev@tu-berlin.de)

**Dipl.-Inf. Martin Kolbe**

CIO, Kühne + Nagel International AG, Hamburg  
(martin.kolbe@Kuehne-Nagel.com)

# Anwendungsmodernisierung – Motivation und Positionierung

*Anwendungsmodernisierung ist ein weites Feld. Dies gilt nicht nur für die Vielfalt der Begriffe, die im Zusammenhang mit dem Thema verwendet werden, sondern vor allem auch für die Vielzahl an Technologien, Produkten, Werkzeugen und Infrastrukturkomponenten, die eine Rolle spielen und beherrscht werden müssen. Oft ist von „Legacy“-Anwendungen die Rede – dies beschreibt die tatsächliche Situation allerdings nur mangelhaft, denn immer öfter stehen gerade Kernanwendungen, die einen wichtigen Beitrag zum Geschäftserfolg leisten, im Mittelpunkt der Modernisierungsüberlegungen. Umso wichtiger ist es, ein klares Verständnis über vorhandene Optionen und mögliche Maßnahmen zu haben, die den zukünftigen Umgang mit den wichtigen und geschäftskritischen Anwendungen planbar machen.*

Die wesentlichen Auslöser für eine Anwendungsmodernisierung sind in vielen Fällen vergleichbar: geringe Flexibilität der Anwendungen, operative Probleme im laufenden Betrieb sowie Skill- und Know-how-Mangel bezüglich Wartung und Weiterentwicklung. Diese Probleme treten dabei meist gekoppelt mit einer schlechten Dokumentation auf allen Ebenen in Erscheinung. Die geringe Flexibilität der Anwendungen ist eine kritische Eigenschaft – gerade in wirtschaftlich schwierigen Zeiten, in denen sich Geschäftsmodelle ändern und sowohl M&A- als auch Carve-Out-Szenarien zur Tagesordnung gehören. Die Integrationsfähigkeit mit internen und externen Anwendungen wird ebenso hoch bewertet. Wenn dazu noch eine mangelhafte Verfügbarkeit und eine Häufung von Ausfällen der Anwendung kommen, wird es höchste Zeit über Alternativen nachzudenken. Gleiches gilt, wenn sich die Suche nach dringend benötigtem Skill immer schwieriger gestaltet, weil die Kernmannschaft das Pensionsalter erreicht. Die Anzahl der Experten für Programmiersprachen wie Assembler, APL, RPG, aber auch in zunehmendem Maße für die Sprachen Cobol und PL/I nimmt rapide ab.

Stellt sich die Frage, warum überhaupt modernisiert werden soll, und ob nicht die Einführung einer Standardsoftware oder eine komplette Neuentwicklung bessere Alternativen sind, die einer Modernisierung der Altanwendung vorzuziehen wären.

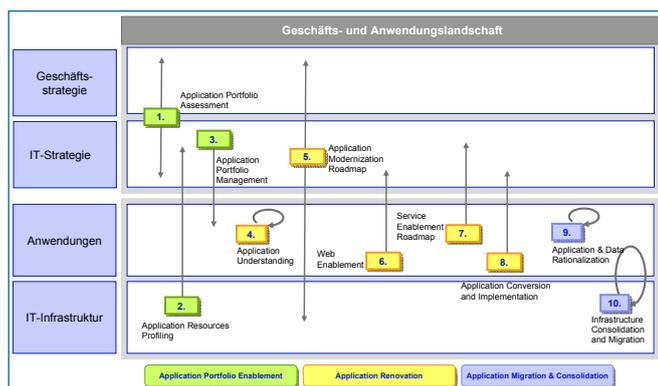


Abb. Einstiegsunkte

Die Einführung einer Standardsoftware ist vor allem dann sinnvoll, wenn es sich um so genannte „Commodity Software“ handelt. Also Software, die keinen oder einen nur sehr geringen wettbewerbsdifferenzierenden Charakter besitzt – beispielsweise der Bereich Personalverwaltung oder Finanzbuchhaltung. Trifft dies nicht zu, läuft man sehr schnell Gefahr, die vielen Spezialfälle und Sonderanforderungen des Kerngeschäfts in einer Standardsoftware nachprogrammieren zu müssen, und entfernt sich dabei weit vom Standard. Die Abhängigkeit vom Lieferanten der Standardsoftware wird oft als weiterer Nachteil empfunden.

Für die Abbildung wettbewerbsdifferenzierender Prozesse ist eine Neuentwicklung gegebenenfalls die bessere Alternative. Dabei müssen allerdings die fachlichen Anforderungen bekannt sein, und die Erstellung der neuen Anwendung muss mit vertretbarem Aufwand auch leistbar sein. Oftmals scheitern Neuentwicklungen ebenso an der langen Entwicklungszeit, während die Altsysteme parallel weiterentwickelt, gewartet und betrieben werden müssen.

In vielen Fällen ist dann eine Modernisierung die bessere Alternative. Oftmals ist es sogar die einzige praktikable Lösung, insbesondere da es mittlerweile sehr unterschiedliche Ansätze und viele Optionen gibt. Methodiken und Vorgehensmodelle der Anwendungsmodernisierung haben sich stark weiterentwickelt. Gleiches gilt für Werkzeuge und Infrastrukturkomponenten. So stehen heute weitgehend skalierbare Lösungsansätze im Vordergrund. Schrittweise Modernisierungen – oft mit mehreren Jahren Laufzeit – senken das Risiko drastisch. Die Palette reicht von Web-Enabling der Anwendung, über das Service-Enablement bis hin zur punktuellen Modernisierung des bestehenden Quellcodes und des Reengineering. Sollen Anwendungen im Wesentlichen ohne Veränderung an Funktionalität und Datenstruktur bei Beibehaltung des Quellcodes auf eine modernere Entwicklungsumgebung oder Infrastruktur portiert werden, so stehen Lösungen der Daten-, Sprach- oder Plattform-Migration beziehungsweise des Re-Hostings zur Verfügung. Diese können heute über Factory-Ansätze oft mit einem hohen Automatisierungsgrad realisiert werden.

Bevor ein Modernisierungsprojekt aufgesetzt und initiiert wird, empfiehlt sich in jedem Falle die Betrachtung des Gesamtportfolios, um potentielle Synergien identifizieren und heben zu können.

Damit die unterschiedlichen Modernisierungsansätze und Möglichkeiten transparent werden, hat sich die Orientierung an einem Modernisierungsframework in der Praxis bewährt. Das hier dargestellte Framework (siehe Abb. Einstiegsunkte) stellt verschiedene Einstiegsunkte dar, die je nach Zuständigkeit angeordnet sind. Alle Einstiegsunkte stehen gleichberechtigt nebeneinander – können allerdings miteinander kombiniert und in einer bestimmten Reihenfolge abgearbeitet werden. Hinter jedem Einstiegspunkt sind das spezifische Vorgehensmodell, die Methodik, der Werkzeugkasten sowie einsetzbare Assets und Beschleuniger hinterlegt.

**Markus Weyerhäuser**  
Leiter IBM GBS Kompetenzzentrum für Anwendungsmodernisierung

**Dr. Uwe Gross**  
Leiter IBM GBS Application Innovation Services

**Jürgen Henn**  
Leiter IBM GBS IT Strategy & Transformation

**Dr. Steffen Graf**  
Bereichsleiter IBM GBS Application Modernization

# Wie optimieren Sie Ihr Anwendungsportfolio?

IT-Funktionen sind heute der Lebensnerv der meisten Unternehmen. Gleichzeitig sind die Zielsetzungen an die IT aber herausfordernd: Wie können Kosten reduziert und gleichzeitig der Service und Nutzen für das Geschäft verbessert werden? Dies ist insbesondere eine Herausforderung für den Bereich der Anwendungen, die direkte Schnittstelle zwischen Business und IT sowie einer der Hauptkostenfaktoren der IT im Unternehmen. Von besonderer Bedeutung ist es daher, das Anwendungsportfolio zu optimieren. Die Zielsetzung ist, Redundanzen und Vielfalt zu reduzieren, damit Sie Kosten senken und refokussieren können. Eine solche zielgerichtete Kosten- und Wertoptimierung kann nur auf der Basis einer entsprechenden Bewertung erfolgen.

komponenten des CBM gemäß der geschäftlichen Bedeutung klassifiziert. Zur Abdeckung der geschäftlichen Kernbereiche mit IT-Lösungen erstellt man im Anschluss den Bebauungsplan auf Basis der CBM, ordnet so die Anwendungen den Geschäftsfunktionen zu und gewichtet die „Qualität“ und die Kosten der Anwendungen gemäß der geschäftlichen Bedeutung und der Anforderungen (siehe Abb. 2).

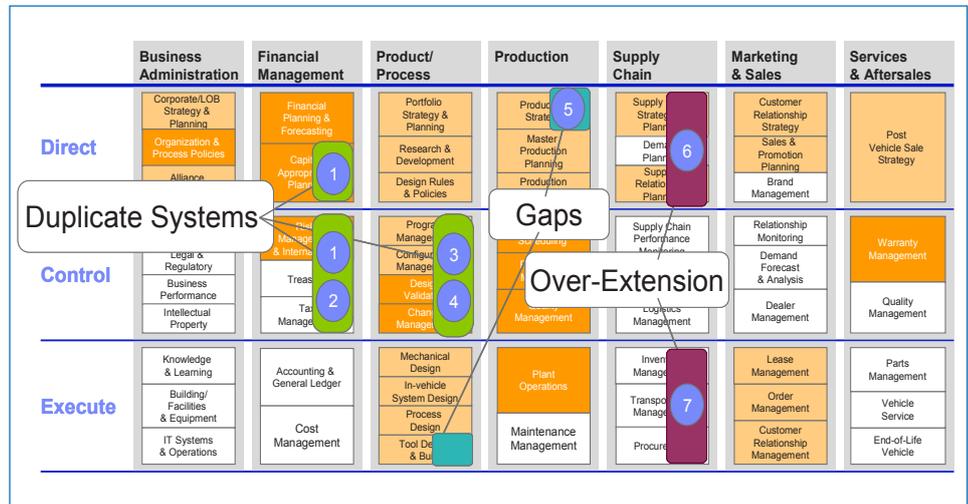


Abb. 2 Bebauungsplan auf Basis einer CBM (Beispiel)

Eine strukturierte Analyse zur Optimierung des Anwendungsportfolios wird Anwendungsportfolio-Assessment genannt.

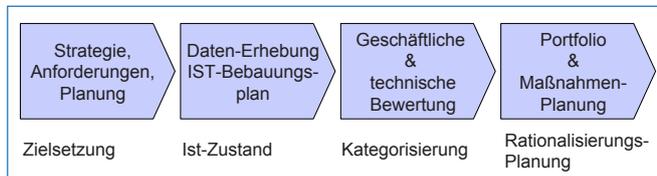


Abb.1 Schematischer Ablauf einer Anwendungsportfolio-Analyse

Ein Anwendungsportfolio-Assessment läuft üblicherweise nach folgendem Schema ab:

Wesentlich ist es, dabei zwei Aspekte zu betrachten: den Wertbeitrag der Anwendungen im Portfolio (ermittelt durch die geschäftliche Bewertung oder „Business Assessment“) als auch die finanziellen und strategischen Implikationen aus der Technologie und der Umsetzung der Anwendungen (die technische Bewertung oder „technical assessment“).

## Geschäftliche Bewertung

Die geschäftlichen Anforderungen bilden die Basis für die Bewertung der Anwendungen. Ein nützliches Werkzeug zur strukturierten Darstellung des Geschäftsmodells ist das von IBM entwickelte Component Business Model (CBM). CBM ist das Modell eines Unternehmens als Zusammenspiel von einzelnen, nicht überlappenden Komponenten. Jede Komponente ist hierbei eine logische Gruppierung von Mitarbeitern, Technologien und Ressourcen, die einen spezifischen Wertbeitrag leistet sowie unabhängig operieren kann.

Um die geschäftlichen Anforderungen zu konsolidieren und auf das Geschäftsmodell abzubilden, werden die Geschäfts-

Zur geschäftlichen Bewertung der Anwendungen dienen zum einen die Kosten, u.a.

- Entwicklungskosten
- Betriebskosten,
- Lizenzkosten und
- Trainingskosten

und zum anderen der „Business Value“, z.B.

- Abgleich mit Geschäftsstrategie und -zielen,
- Effizienz- und Umsatzsteigerungen oder
- Kundenunterstützung.

Zudem erhebt man zusätzlich Kriterien der funktionalen Qualität (Verfügbarkeit, Benutzerfreundlichkeit, etc.) und der Datenqualität (Richtigkeit, Verfügbarkeit, etc.) und verfeinert damit die Bewertung des geschäftlichen Nutzens der Anwendungen.

Eine solche Basis der geschäftlichen Anwendungsbewertung bildet die Grundlage, um die Anwendungen zu klassifizieren, die Handlungsoptionen hinsichtlich Nutzen, Kosten und Risiken zu bewerten und den ROI (Return of Investment) zu berechnen.

## Technische Bewertung

Neben der Untersuchung des geschäftlichen Nutzens von Anwendungen sollten im Rahmen eines Anwendungsportfolio-Assessments technische Aspekte betrachtet werden. Auf diese Weise ist eine gesamtheitliche Analyse von Anwendungen möglich.

Folgende Struktur zur Gliederung der technischen Aspekte hat sich in der Praxis bewährt:

- Wartungsverhalten
- Technische Umsetzung
- Technische Qualität
- Schnittstellen
- Input-/Output-Management
- Betrieb
- Ressourcen und Skills
- Dokumentation

Unter dem Wartungsverhalten einer Anwendung werden die Aspekte zusammengefasst, die die ausreichende Abdeckung der geschäftlichen Anforderungen sicherstellen. Dazu zählen insbesondere die Anzahl der Change Requests pro Jahr, die Anzahl an Releases pro Jahr sowie der Aufwand je Release.

Die technische Umsetzung beschreibt die Systemkomponenten, auf denen eine Anwendung basiert. Wesentliche Bereiche sind – je nach konkretem Umfeld – Hardware-Plattform(en), Betriebssystem, Programmiersprache, Benutzerschnittstelle, Datenhaltungssysteme, Transaktionsmonitor, Kommunikationssoftware sowie Werkzeuge zum Systems Management (z.B. Monitoring, Sicherheit).

Die technische Qualität von Anwendungen stellt eine wesentliche Entscheidungsgrundlage für die Beurteilung von Anwendungen dar. Beurteilt werden dabei Faktoren wie die Einhaltung unternehmensspezifischer Architekturvorgaben und Programmierstandards, die Häufigkeit und Tragweite von Programmfehlern, die zu Systemabstürzen führen, und die Zukunftssicherheit der Technologie. Darüber hinaus kann man die Komplexität von Programmen mittels unterschiedlicher Indikatoren einschätzen.

Anzahl und Art der Schnittstellen sind ein wichtiger Faktor bei der Analyse von Anwendungen. Im Wesentlichen untersucht man die Schnittstellen zwischen einzelnen Anwendungen, zu Datenbanken und Dateien sowie zu externen Partnern. Allen Arten von Schnittstellen sind grundlegende Bewertungsfaktoren gemeinsam: Anzahl, Häufigkeit der Aufrufe, durchschnittliches Datenvolumen je Aufruf, Art (synchon/asynchron), Kontrollfluss sowie Sicherheitsanforderungen.

Aus dem Input-/Output-Management einer Anwendung lassen sich wertvolle Erkenntnisse gewinnen. Im Rahmen eines Application Portfolio Assessments sollte insbesondere analysiert werden, welche Medien (z.B. Papier, Scan, RFID) zur Ausgabe sowie zur Erfassung genutzt werden, welche Werkzeuge eingesetzt werden und wie hoch das Volumen je Medium ist. Darüber hinaus sind tiefer gehende Analysen des Druckumfelds lohnend. Faktoren wie beispielsweise das Druckvolumen im zentralen bzw. im dezentralen Bereich sowie die Verteilung von Druckstücken können wesentliche Kostentreiber darstellen.

Der Betrieb von Anwendungen muss bei einem Anwendungsportfolio-Assessment intensiv betrachtet werden. In diesen Bereich fallen insbesondere tatsächliche und geforderte Verfügbarkeit, Service-Level-Vereinbarungen sowie Verbrauch an Hardware-Ressourcen wie z. B. CPU-Zeit oder Bandbreiten im Netzwerk.

Die Ressourcen und Skills, die in die Entwicklung einer Anwendung gesteckt wurden und werden, können als Indikator für Umfang und Komplexität dienen. Die aktuelle und zukünftige Verfügbarkeit der Skills für die Wartung einer Anwendung, ist darüber hinaus ein wichtiger Faktor zur Beurteilung der Zukunftssicherheit von Anwendungen.

Das Vorliegen einer aussagekräftigen Dokumentation ist ein wichtiger Faktor bei der Anwendungsanalyse. Dies gilt insbesondere vor dem Hintergrund der Tatsache, dass viele „Programmierer der ersten Stunden“ bereits im Ruhestand sind oder in den nächsten Jahren in den Ruhestand gehen werden. Wesentliche Teile einer Dokumentation sind Grob- und Fein-Design, Datenmodelle, Benutzerhandbücher, Schulungsunterlagen etc.

In Abhängigkeit von konkreten Projektzielen legt man fest, wie detailliert für jeden der o. g. Punkte eine Analyse zu erfolgen hat. Die Einzelergebnisse werden anschließend zu einer Gesamtaussage je Anwendung zusammengeführt. Die unterschiedliche Relevanz einzelner Punkte kann man in konkreten Kundensituationen mit Hilfe von Gewichtungsfaktoren abbilden. Aus den Einzelanalysen für jede Anwendung lassen sich Schlussfolgerungen und Handlungsempfehlungen für ein gesamtes Anwendungsportfolio ableiten und priorisieren. Dabei spielen in der Regel Kostenaspekte eine zentrale Rolle. Ein fundiertes Anwendungsportfolio-Assessment bildet einen exzellenten Einstieg in ein aktives Anwendungsportfolio-Management.

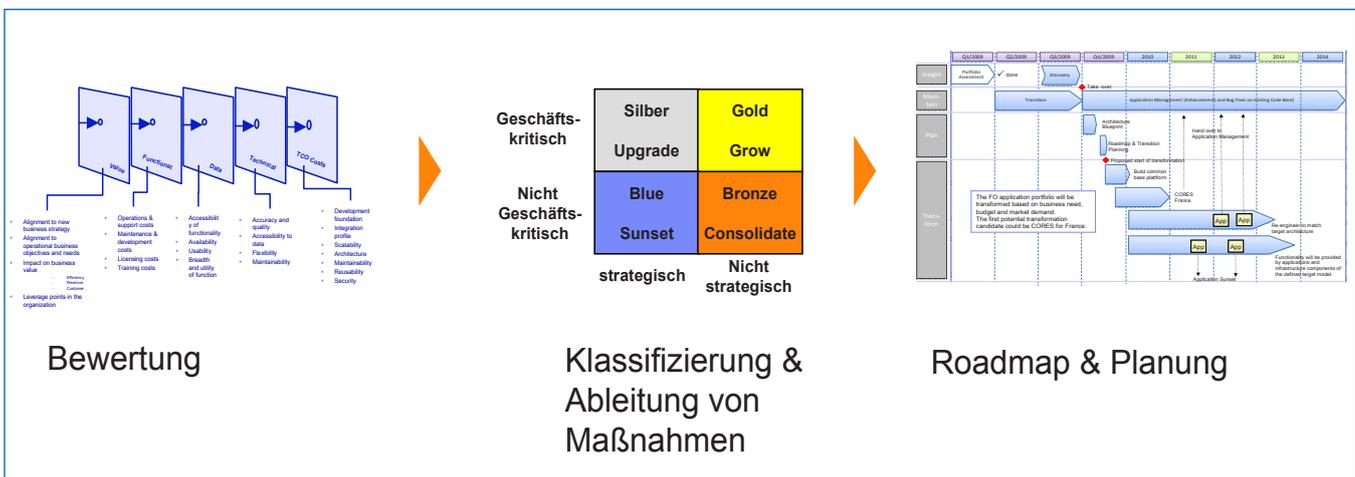


Abb. 3 Ableitung von Handlungsempfehlungen

## Aufstellung des Portfolios und Ableitung von Empfehlungen

Die Bewertung der Anwendungen mit den abgestimmten Kriterien erlaubt es, die Anwendungen in speziellen Portfolios zu positionieren. Damit wird die Ableitung von Maßnahmen wirkungsvoll unterstützt:

- Realisierung kurzfristiger Kosteneinsparungen durch abgestufte Service Level und somit Refokussierung und Rationalisierung der Service-Kosten: Mit der Bewertung von Anwendungen kann man den notwendigen Betriebs- und Service-Aufwand für jede Anwendung neu bewerten. Üblicherweise werden die Service-Klassen Gold, Silber, Bronze und Blau definiert und die Anwendungen gemäß ihrer geschäftlichen Relevanz und weiterer Kriterien eingeordnet. Erfahrungen zeigen Einsparungspotenziale von 10-30% bei den Betriebs- und Wartungskosten.
- Rationalisierung des Anwendungsportfolios durch Verringerung von Redundanz, Konsolidierung und Ausrichtung an den geschäftlichen Prioritäten  
Durch konsequente Rationalisierung können erfahrungsgemäß Einsparungspotenziale von bis zu 20% realisiert werden.
- Ableitung von fokussierten Maßnahmen zur Modernisierung einzelner Applikationen zur Sicherung der IT-Unterstützung und Eindämmung zukünftiger Kosten und Risiken.  
Wesentlicher Nutzen aus Modernisierungsmaßnahmen ist die Reduktion von Risiken durch veraltete Anwendungen und Technologien sowie die Konsolidierung und Zukunftssicherung des Technologie-Portfolios.

Die Rationalisierung des Anwendungsportfolios kann man aufgrund der Ergebnisse gezielt planen. Redundanzen und Konsolidierungspotenzial wurden identifiziert sowie die Fokussierung auf geschäftliche Prioritäten definiert.

Ein wesentliches Ergebnis ist die Identifikation von Modernisierungsprojekten auf der Basis der Ergebnisse des Anwendungsportfolio-Assessments (siehe Abb. 3).

### Markus Weyerhäuser

Leiter IBM GBS Kompetenzzentrum für Anwendungsmodernisierung  
(markus.weyerhaeuser@de.ibm.com)

### Dr. Ralf Jungclaus

Senior Managing Consultant,  
IBM Global Business Services  
(ralf.jungclaus@de.ibm.com)

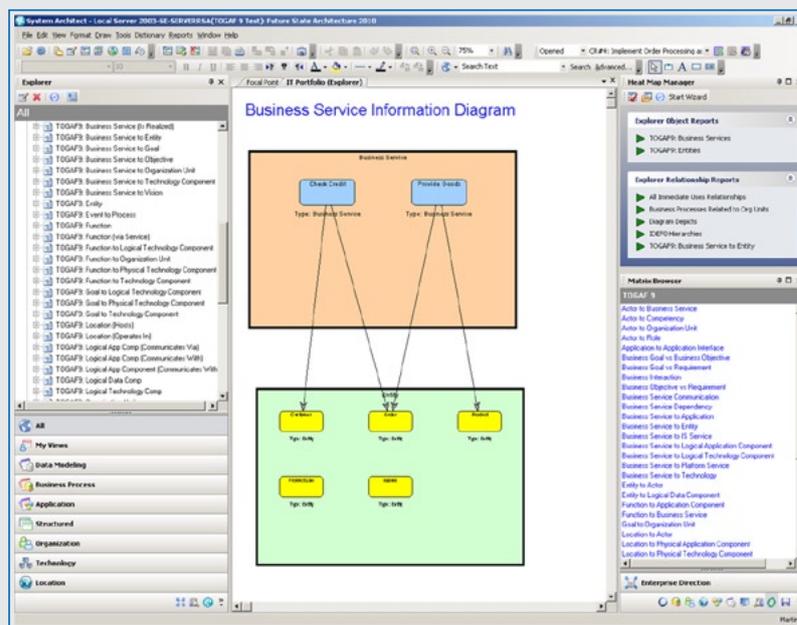
### Dr. Ulrich Herrmann

Executive IT Architect,  
IBM Global Business Services  
(ulrich.herrmann@de.ibm.com)

IBM Rational System Architect fasst die Informationen rund um Enterprise Architecture Management (EAM) zentral zusammen, speichert diese in einer Datenbank und ermöglicht so übergreifende Auswertungen über alle Bereiche der EAM, angefangen von Unternehmenszielen und Strategien über Prozesse und Applikationen bis hin zu Infrastruktur und Datenmodellen.

Dazu unterstützt er Standardframeworks und -methoden wie TOGAF, Zachman, ITIL, NGOSS, NAF, FEAF und demnächst auch CBM und IFW/IAA. Methoden wie BPMN, UML, IDEF, ER sind standardmäßig enthalten. IBM Rational System Architect kann aber auch beliebig angepasst und mit kundenspezifischen Frameworks, Methoden und Metamodellen bestückt werden. Attribute können frei definiert werden. Reporting und Publishing lassen sich flexibel konfigurieren, um alle Fragen aus dem EAM beantworten zu können.

Vorgefertigte Analysen erleichtern das Auswerten der Informationen und erlauben die visuelle Kommunikation unterschiedlichster Fragestellungen wie z. B. Auswirkungsanalysen und Anwendungslandschaften.



Ebenso lässt sich IBM Rational System Architect gut mit anderen Werkzeugen integrieren. Man kann z. B. über WebSphere Business Modeller Prozesse in Web Services überführen. Ebenso ist möglich, Blueprint-Informationen aus einer Tivoli CMDDB automatisiert für die As-Is – To-Be Planung zu synchronisieren. Verteilte Teams werden über Web Clients eingebunden und können direkt ihre aktuellen Inhalte beisteuern.

Umfangreiche Web-Publishing-Möglichkeiten helfen dabei, die Erkenntnisse und Informationen aus dem Enterprise Architecture Management wiederum für internes und/oder externes Marketing zu nutzen.

# APPLIKATIONSLANDSCHAFTEN ANALYSIEREN, VERMESSEN, DOKUMENTIEREN

Schnelle Reaktionen auf Marktereignisse, Gesetzesänderungen und Managementvorgaben bei gleichzeitig hohem Kostendruck erfordern flexibel steuerbare Geschäftsprozesse, die zunehmend automatisiert werden. Umfassendes, stets aktuelles und effizient auswertbares Wissen über den hochkomplexen Gesamtzusammenhang ist Voraussetzung für eine wirtschaftliche Business Application Modernization.

**AREDIS/ApplicationMiner** kombiniert Methoden und Techniken aus den Bereichen Compilerbau, Data-Mining und Softwarevisualisierung zu einer Plattform mit umfassender Funktionalität zur Analyse, Vermessung und Dokumentation bestehender Applikationslandschaften.

**Extraktoren** selektieren und übernehmen die Objekte des Application Portfolio aus den Entwicklungs- und Betriebsumgebungen.

**Analysatoren** zerlegen die extrahierten Objekte in ihre grammatischen Elemente und rekonstruieren die vielschichtigen Bezüge zwischen Prozessen, Programmen, Datenbanken, Laufzeitverhalten und Projekten präzise in einer multidimensional organisierten Wissensbasis. Das Schema ist flexibel ergänzbar. Neue Objekte und Fakten werden inkrementell in den Kontext einbezogen.

**Visualisierer** kombinieren die Ergebnisse der Analysatoren zu aussagekräftigen Sichten und präsentieren diese anschaulich in tabellarischer und grafischer Form.

Der **Leitstand** bietet Funktionen zur effizienten Auswertung der Wissensbasis. *Drill-down-* und *Roll-up-*Operationen ermöglichen Recherchen auf unterschiedlichen Abstraktionsebenen – Programm- und Datenbankdetails auf der untersten Ebene und die Zusammenhänge zwischen Prozessen, Applikationen und Projekten auf der obersten Ebene.

Der **Grafikeditor** unterstützt die Umsetzung rekonstruierter Strukturelemente in grafische Symbole und deren Zuordnung zu logischen Sichten – von der Architektur einzelner Applikationen bis zum Bebauungsplan der Applikationslandschaft.

Der **Dokumentengenerator** erzeugt eine technische Dokumentation mit vorgebbarem Inhalt wie z. B. Programmstrukturen, Datenbankstrukturen und Projektorganisation in den Formaten HTML und PDF.

Auswählbare Inhalte der Wissensbasis können in den Formaten „XMI“ oder „Relational“ exportiert und mit einem **CASE-Tool** wie **IBM Rational Rose®** oder einem **Repository** wie **ASG-Rochade®** weiterverarbeitet werden.

## Technische Eigenschaften:

- Analysatoren für die Sprachen **COBOL, PL/1, REXX, JCL, C, C++, C#, VisualBasic, Pascal, Java**, für verbreitete Datenbanksysteme und für strukturierte Dokumente
- Metriken zur Messung genereller und individueller Größen-, Komplexitäts- und Qualitätsmerkmale
- Unterstützung des Qualitätsmodells für Software ISO IEC 9126
- offen dokumentierte Schnittstellen und deshalb leicht in bestehende Entwicklungsumgebungen wie z. B. Eclipse integrierbar
- Betriebsfähigkeit der Extraktoren und Analysatoren unter IBM-z/OS®, MS-Windows®, UNIX®, Sun-Solaris® und Linux®
- Bewältigung immenser Datenmengen

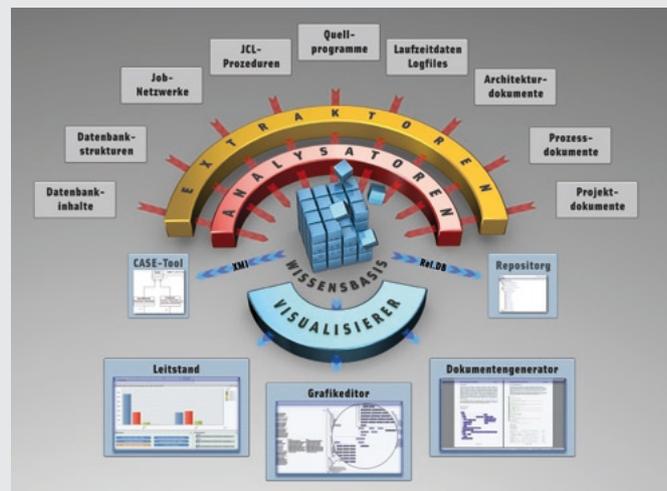


Abb. AREDIS / ApplicationMiner – Funktionsübersicht

- bedarfsgerecht einstellbar und skalierbar vom Einplatzsystem bis zu flächendeckenden Client/Server-Netzwerken

## Anwendungen:

- Schaffung von Transparenz als Voraussetzung für eine wirtschaftliche Weiterentwicklung und Modernisierung
- effiziente Projektsteuerung durch wissensbasierte Zusammenarbeit
- automatisierte Nachdokumentation der Programmstatik, -dynamik und -logik
- automatische Qualitätsprüfung von Quellprogrammen auf Konformität mit vorgegebenen Architektur- und Code-Konventionen
- Codereview mit Abhängigkeits- und Seiteneffektanalysen
- Unterstützung moderner Regelwerke wie CMMI®, ISO 9000®, ITIL®, RUP®, V-Modell® sowie der IT-Compliance und der IT-Governance

Banken, Versicherungen, Behörden, Handels- und Industrieunternehmen nutzen AREDIS/ApplicationMiner zur Beschleunigung zeitaufwendiger Routineaufgaben und zur Steigerung der Software-Qualität.

Nennen Sie uns Ihren konkreten Bedarf. Wir liefern maßgeschneiderte Lösungen.



Quintec Informationstechnologie GmbH

Kirchstraße 12, 53840 Troisdorf

Telefon: 02241 9826 15

E-Mail: info@quintec-it.de

Internet: www.quintec-it.de

# Modernisierung von IT-Applikationen – Wege zum Erfolg

IT-Modernisierungsprojekten sind komplex und der Projekterfolg ist eng verknüpft mit der Fähigkeit den Geschäftsbetrieb aufrecht zu erhalten, besonders wenn es um geschäftskritische Anwendungen geht. Auf welcher Basis entscheidet man nun bei welcher Anwendung sich eine Modernisierung lohnt bzw. andere Maßnahmen sinnvoller angewandt werden sollten? Eine Möglichkeit ist die Durchführung eines Portfolio Assessments so wie auf Seite [Seitenverweis einfügen] beschrieben. In vielen Fällen drängen sich bestimmte Modernisierungsmaßnahmen aber gerade zu auf. Aber wie nun starten? In welchen Stufen geht man vor? Wie sieht die Modernisierungsstrategie aus? Welche Technologien kommen zum Einsatz? Zur Beantwortung dieser Fragestellungen lohnt es sich, eine Strategiephase vorzuschalten.

Wesentlicher Bestandteil dieser ersten Projektphase ist es, eine detaillierte Bestandsaufnahme durchzuführen, sich Klarheit über das gewünschte Zielbild zu verschaffen und den Weg dorthin genau zu beschreiben. Dabei werden auch alternative Szenarien beschrieben und bewertet. Das Ergebnis dieser Phase dient dem gemeinsamen Verständnis – sowohl innerhalb des Teams als auch als Basis für die Kommunikation zu anderen Geschäftsbereichen und Entscheidungsträgern. Bewährt hat sich ein dreistufiges Vorgehen. Dabei müssen die einzelnen Phasen nicht streng chronologisch durchgeführt werden. Eine teilweise Überlappung der Phasen bzw. ein inkrementelles Vorgehen ist durchaus gewünscht und empfehlenswert.

## Phase 1 - Durchführung der Bestandsaufnahme

In Phase 1 erfolgt eine umfassende Bestandsaufnahme. Dabei wird ein IT-übergreifender Blick gewählt, das heißt auch Geschäftsstrategie und fachliche Aspekte spielen eine Rolle und werden untersucht. Ziel ist es, das Problemfeld aus den unterschiedlichen Blickwinkeln zu verstehen, die wichtigsten Treiber und Kennzahlen zu identifizieren und eine Inventarisierung durchzuführen. Aus diesem Grund werden sowohl fachliche, technische als auch strategische Voraussetzungen auf den unterschiedlichen Detailebenen untersucht und dokumentiert. Während der Phase der Fachanalyse wird zum einen der genaue Umfang des Modernisierungsvorhabens abgesteckt und zum anderen die fachlichen Voraussetzungen erhoben, die als Input für Zielarchitektur und Handlungsempfehlungen dienen. Dazu gehören beispielsweise die Identifizierung von Fachkomponenten, Geschäftsprozessen, Use Cases und Datenmodellen. Während der Aufnahme der IT-relevanten Artefakte ist ein Abgleich mit den Ergebnissen der Fachanalyse durchaus sinnvoll. Ein wesentlicher Aspekt der Bestandsaufnahme ist die projektspezifische Identifikation relevanter KPIs und deren Quantifizierung. Beispiele für gängige KPIs sind z.B. die Anzahl der Programme, die Anzahl der Code-Zahlen, verwendete Programmiersprachen, Anzahl Bildschirmmasken und Anzahl der Tabellen.

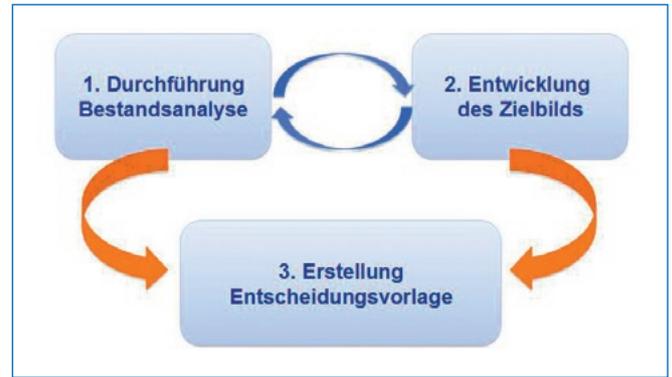


Abb. Phasenmodell

## Phase 2 - Entwicklung des Zielbildes

Nach der Bestandsanalyse erfolgt die Entwicklung des Zielbildes. Das Zielbild zeigt die zukünftige Architektur sowohl aus fachlicher als auch IT-Sicht auf, inklusive Anwendungsarchitektur und operationalem Modell. Je nach Aufgabenstellung können auch relevante IT-Prozesse wie z. B. Anforderungsmanagement, Build- und Release-Management oder auch Test- und Entwicklungsprozess definiert werden. Auch hier wird man im Sinne eines Zielbildes keine detaillierten Abläufe beschreiben, sondern die wesentlichen Merkmale, Eigenschaften und Rahmenbedingungen definieren. Bestandteile der Facharchitektur sind die wesentlichen Fachkomponenten, Geschäftsprozesse und Use Cases, die umgesetzt werden sollen. Die IT-Architektur identifiziert und beschreibt Architekturprinzipien und Architekturentscheidungen. Die Ausrichtung der zukünftigen IT-Architektur wird sich oft an dem Blueprint einer Service-orientierten Architektur (SOA) orientieren. Auf Basis der Fach- und IT-Architektur können nun verschiedene tiefer gehende Darstellungen gewählt werden, die das Zielsystem weiter beschreiben und konkretisieren. So kann zum Beispiel ein Technologie-Blueprint erstellt werden, der die unterschiedlichen Technologien definiert, mit denen die Architekturschicht realisiert werden. Ähnliches kann auch für einzusetzende (Kauf-)Produkte und Technologien erfolgen.

## Phase 3 - Erarbeitung der Handlungsempfehlungen

Während das Zielbild die zukünftige Landschaft skizziert, wird in der dritten und letzten Phase beschrieben, wie dieses Zielbild erreicht werden soll. Wesentliches Ergebnis dieser Stufe sind Handlungsempfehlungen und eine detaillierte Roadmap. In dieser Phase werden unterschiedliche Modernisierungsszenarien und Optionen dokumentiert und bewertet, Chancen und Risiken beschrieben und Handlungsoptionen dokumentiert. Von hoher Bedeutung ist dabei auch die Erstellung einer Aufwandsschätzung und gegebenenfalls eines Business Case. Dabei wird auf die Ergebnisse der beiden ersten Phasen zurückgegriffen. Auf Basis der Handlungsempfehlungen und des geschätzten Aufwandes kann nun eine Roadmap erstellt werden. Ein Skill- und Ressourcenplan rundet die Entscheidungsvorlage ab. ■

### Markus Weyerhäuser

Leiter IBM Kompetenzzentrum für Anwendungsmodernisierung (markus.weyerhaeuser@de.ibm.com)

# Die Wahrheit liegt im Code

Ein solides Verständnis von Altanwendungen ist der Schlüssel zu deren erfolgreicher Modernisierung. Systematisches Vorgehen und Analysetools können dabei helfen, dieses Verständnis zu gewinnen. Liegt keine lückenlose Dokumentation der bisherigen Entwicklungsprozesse vor, muss Application Understanding, also die systematische Nachdokumentation, betrieben werden. Anhand des Source Codes kann man die ursprünglichen Regeln und Anforderungen nachvollziehen und eine sinnvolle Modernisierung des Systems starten.

## Ziele

Das Ziel der Analyse sollte man zu Beginn präzise festlegen. Wenn eine Anwendung z. B. durch Web Service Enablement (siehe „Save Our Assets – Alte Systeme im neuen Gewand“, S. 15) nach außen geöffnet werden soll, muss man lediglich die relevanten Schnittstellen identifizieren, nicht jedoch die Anwendung bis ins Detail aufgliedern.

Andere mögliche Ziele einer Analyse sind:

- **Erstellung eines Datenkatalogs**  
Welche Daten sind in meiner Anwendung hinterlegt – unter welchen, möglicherweise unterschiedlichen, Bezeichnungen? Wo in der Anwendung werden die Daten verändert, auf Grund welcher Eingaben?
- **Impact-Analyse**  
Wer ist betroffen, wenn ich eine Felddefinition ändere? Diese Frage war aus gegebenem Anlass zur Jahrtausendwende brandaktuell. Viele der heute gängigen Tools haben ihre Wurzeln im Y2K-Problem.
- **Ermittlung von Metriken**  
Welche Anwendungen sind am wartungsanfälligsten? Die Beantwortung dieser Frage sagt Ihnen, an welchen Stellen sich Investitionen in Wissenstransfer und (Nach-)Dokumentation lohnen. Die Werkzeuge errechnen dazu Kennzahlen, die Richtwerte für die Komplexität eines Programms liefern. Stellvertretend seien hier die Metriken von McCabe [1] und von Halstead [2] genannt.
- **Businessregel-Extraktion**  
Welche Geschäftsregeln – womöglich nirgendwo mehr zuverlässig dokumentiert – sind im Code verborgen?

Obwohl es unterschiedliche Szenarien und Motivationsfaktoren für das Thema Application Understanding gibt, hat sich ein Vorgehensmodell etabliert, das sich auf projektspezifische Anforderungen anpassen lässt (Abb. Vorgehensmodell).

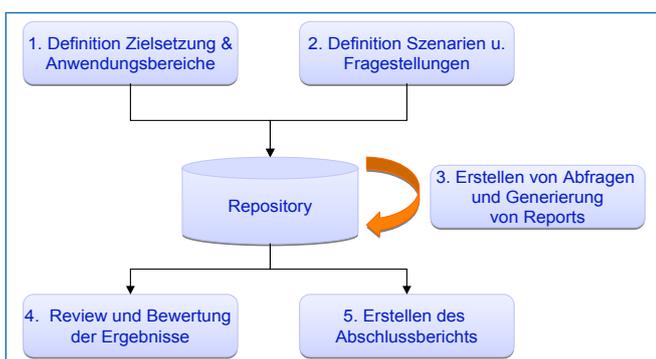


Abb. Vorgehensmodell

## Aufbau des Repositorys

Tools zur Code-Analyse verfügen über Scanner, die den Code zunächst – wie dies auch ein Compiler tun würde – in einen abstrakten Syntaxbaum überführen. Der wird weiter analysiert und dann in sprachunabhängiger Form in einem Repository abgelegt, das Dreh- und Angelpunkt für folgende Analysen ist. Für die Struktur solcher Repositories gibt es mittlerweile auch einen OMG Standard (vgl. [2]).

Es ist nicht damit getan, das Repository nur aufzubauen: der beste Datenbestand ist nutzlos ohne gute Werkzeuge zu seiner Aufbereitung. Die gängigen Tools setzen unterschiedliche Schwerpunkte. Eines der Werkzeuge aus dem IBM-Portfolio stellen wir auf der Seite 14 „Statische Code-Analyse mit Rational Asset Analyzer (RAA)“ ausführlicher vor.

## Möglichkeiten und Grenzen

Es ist kein leichtes Unterfangen, aus existierendem Source Code automatisch „lesbare“ Geschäftsregeln zu machen, die ein Businessanalyst ohne weiteres verstehen kann. Ein über die Jahre gewachsenes Softwaresystem ist das Ergebnis eines langen Prozesses – die Annahme, dass man auf Knopfdruck vom Endergebnis zurück zum Ausgangspunkt gelangen kann, ist naiv.

Tools können Ihnen als Entwickler oder Businessanalyst jedoch dabei helfen. Viele Geschäftsregeln, wie Validierungsprüfungen, lassen sich anhand typischer Muster im Code identifizieren und extrahieren. Unabdingbar ist die Erstellung eines Datenkatalogs, der Programmbezeichner in „Klartext“ übersetzt – „Zuzahlende Zinsen“ liest sich nun einmal leichter als ZUZAZI.

Im folgenden Praxisbeispiel haben wir die festen Bestandteile des Musters in rot, die variablen Bestandteile in blau hervorgehoben:

```

    IF E-ZINSSATZ > K-ZINSSATZ-MAX
    MOVE E-ZINSSATZ-X TO FEHLER-FELD
    MOVE 019 TO FEHLER-NR
    PERFORM FEHLERAUSGABE
    GO TO K999-EX.
  
```

Mit Hilfe des Datenkatalogs lässt sich dieses Code-Fragment in natürliche Sprache übersetzen: „Wenn Zinssatz > maximaler Zinssatz, dann Fehler 019“ (siehe Artikel „Flexible Modernisierung von Anwendungen mit ARC“, S. 12).

Wir wollen nicht verschweigen: Tools können enorm unterstützen, die Erfahrung von Experten lässt sich jedoch nicht ersetzen. Je höher der angestrebte Automatisierungsgrad, desto mehr Aufwand wird man in die Konfiguration der Tools investieren müssen. Dieser Aufwand wird aber in großen Projekten die Qualität der Ergebnisse deutlich verbessern. ■

## Literatur/Quellen

- [1] [www.literateprogramming.com/mccabe.pdf](http://www.literateprogramming.com/mccabe.pdf)
- [2] [adm.omg.org/](http://adm.omg.org/)

### Ulrich Seelbach

beschäftigt sich bei IBM Global Business Services seit über zehn Jahren mit Anwendungsmodernisierung, insbesondere mit Java-Technologie unter z/OS. (ulrich.seelbach@de.ibm.com)

### Markus Weyerhäuser

Leiter IBM GBS Kompetenzzentrum für Anwendungsmodernisierung (markus.weyerhaeuser@de.ibm.com)

# Flexible Modernisierung von Anwendungen mit ARC

Der IBM Analysis & Renovation Catalyst (ARC) ist ein effizientes und äußerst flexibles Werkzeug zur Analyse und Weiterentwicklung von bestehenden Anwendungen auf Unternehmensebene. ARC erfasst und beschreibt diese Anwendungen in einer plattformunabhängigen semantischen Darstellung, die auf einem Metamodell basiert. Damit erlaubt ARC, die Abhängigkeiten und Beziehungen zwischen verschiedenen Plattformen, Programmiersprachen oder allgemein technischen Artefakten zu modellieren. Aus dem Modell heraus können wiederum modernisierte Komponenten (Code, Daten, Benutzerschnittstellen) für eine spezifische Zielplattform generiert werden.

Anwendungen auf Unternehmensebene müssen von Zeit zu Zeit grundlegend modernisiert werden. Grund dafür sind meist veraltete Technologien und die sich daraus ergebenden Probleme in Betrieb, Wartung und Weiterentwicklung.

Prinzipiell wäre es viel effizienter und kostengünstiger, die bestehenden Anwendungen zu modernisieren, anstatt sie komplett zu ersetzen und neu zu entwickeln. Durch die Wiederverwendung bleiben bereits getätigte Investitionen in bestehende Anwendungen erhalten. Ein solcher Renovierungsansatz impliziert aber, dass die Semantik von Logik und Struktur der Anwendung verstanden und analysiert werden muss.

Der Kern einer solchen Analyse ist ein Mechanismus zur Erkennung und Analyse von Mustern. Diese Abstraktion der Funktionalität einer Anwendung erfolgt in Form von *semantischen Assoziationen*, also der Beschreibung von Beziehungen zwischen Objekten, die in einer Inventardatenbank gespeichert werden.

Aus der Inventardatenbank heraus wird die Funktionalität der Anwendung analysiert. Die fachliche Mitarbeit von Anwendungsexpert/innen ist dabei von großer Bedeutung, da die technischen Teile der Applikation verstanden und im Modell semantisch hinzugefügt werden müssen. Die Analyse ist ein iterativer Prozess, wobei es in jeder Iteration zu Anpassungen und Erweiterungen des Modells kommt.

Die Inventardaten werden allen Experten/innen z. B. über Web-Applikationen zur Verfügung gestellt. Das Modell kann dabei grafisch oder in Tabellenform dargestellt werden. Alle semantischen Assoziationen sind durch Hyperlinks realisiert, so dass eine einfache Navigation im Modell leicht möglich ist. Eine Ausgabe in Form von Tabellen-Reports oder UML-Diagrammen ist ebenfalls sinnvoll.

Ein semantisches Modell erlaubt die Integration verschiedener Aspekte des Systems – Anwendungslogik, Datenmodell, Datenzugriffe, Benutzerschnittstelle, Geschäftsprozesse, Komplexitätsmetriken – in eine einzige konsistente Ansicht. Aus dieser Analyse entsteht ein grundlegendes Verständnis über die Anwendung, was im nächsten Schritt eine sinnvolle Transformation in eine neue Technologie ermöglicht.

Das Modell in der Inventardatenbank ist ein fundamentaler Baustein für eine Software-Transformation. Analysemodule identifizieren Muster und Strukturen im plattformunabhängigen Modell. Aus diesen Mustern kann mithilfe von Generierungsmodulen Code für eine Zielplattform (z. B. Java) erzeugt

werden. Auch die Code-Generierung kann man dabei spezifisch an die Bedürfnisse des Kunden anpassen. Eventuell muss der transformierte Code noch manuell weiterbearbeitet oder ergänzt werden, doch der Wert des Modells liegt in der Möglichkeit, ein Applikationsgerüst automatisch zu erstellen.

ARC ist ein IBM Werkzeug, das diesen Modellierungsansatz implementiert, und bietet einen Baukasten mit compilerähnlichen Parsern und Modulen zur Abstraktion an, um die bestehende Anwendung in ein plattformunabhängiges Modell zu übersetzen.

Die Komponenten, aus denen ARC aufgebaut ist, sind Teil einer erweiterbaren und skalierbaren Architektur. Anwendungen mit mehreren Millionen Zeilen Code können in ARC problemlos erfasst werden. Die Übersetzungsmodule parsen und transformieren zurzeit schon Programmiersprachen, wie Cobol, PL/I, Java, SQL, JCL, CSP, EGL, Visual Basic oder Smalltalk in ein gemeinsames semantisches Metamodell. Neue Übersetzungsmodule für zusätzlich benötigte Programmiersprachen können in der ARC Technologie schnell und effizient entwickelt werden. Diese schnelle und flexible Anpassung an kundenspezifische Anforderungen ist eine wesentliche Stärke von ARC.

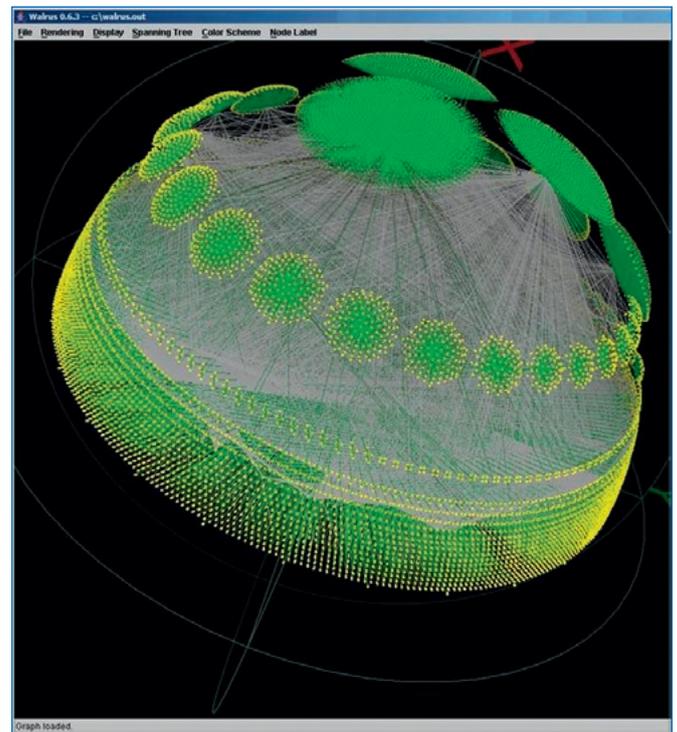


Abb. Modul-Daten Beziehungen (Cobol Anwendung)

Der Analysis & Renovation Catalyst wird von der IBM in internationalen Projekten erfolgreich eingesetzt und reduziert den erforderlichen Entwicklungsaufwand in einem Renovierungsprojekt substantiell im Vergleich zu einer kompletten Neuentwicklung der Anwendung. ■

## Robert Lojek

Senior IT Architect, Chief Technologist ARC  
(robert.lojek@de.ibm.com)

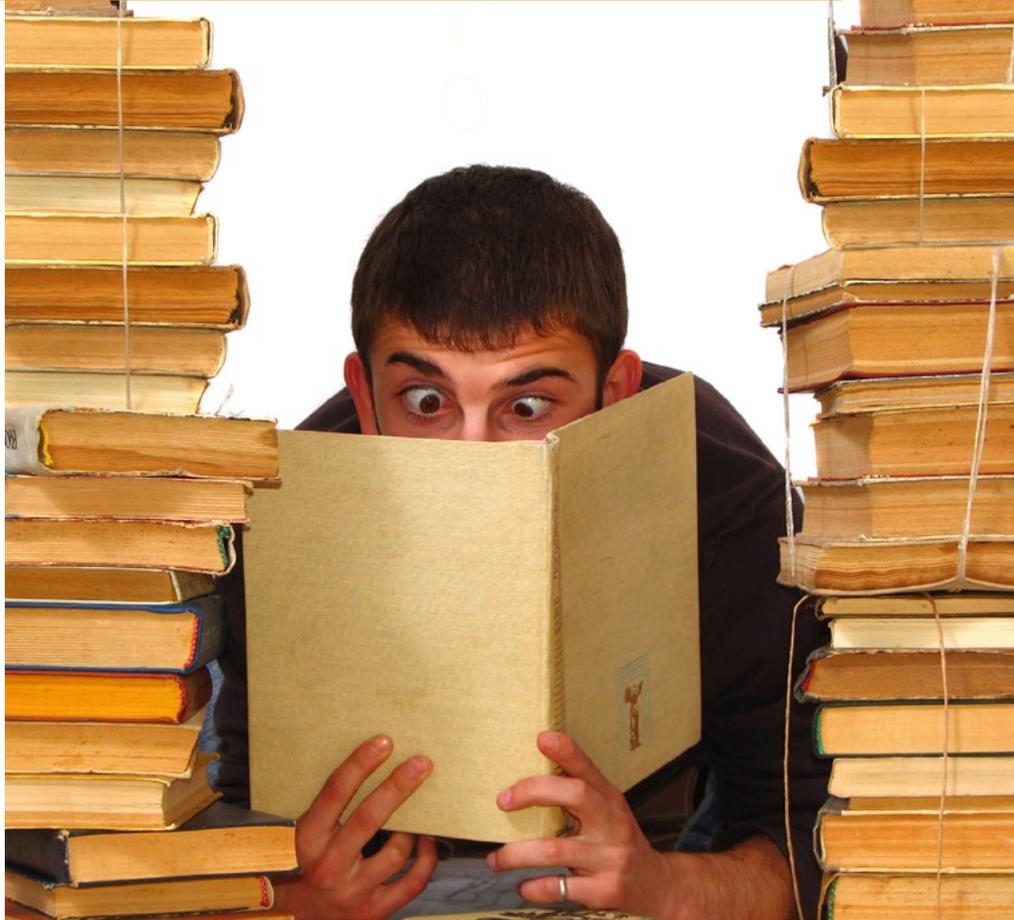
## Günter Hübner

IT Specialist (guenter.huebner@de.ibm.com)

## Rudolf Pailer

Senior IT Architect (rudolf.pailer@at.ibm.com)

# Sie suchen Fach-WISSEN?



## Dann besuchen Sie unsere große Wissensbibliothek mit Fachinformationen zu Ihren IT-Themen.

Hier finden Sie hochaktuelle Fachbeiträge, Praxisberichte, Vortragsfolien, Webcasts und White Paper unserer Experten zu den aktuellen und kommenden Technologien bei der Softwareentwicklung.

In der Rubrik Themenchannel haben wir hochwertige Fachinformationen zu Trendthemen wie SOA/BPM, Scrum/Agilität oder Software Architektur für Sie gesammelt, alles gebündelt und per Mausklick bequem abrufbar.

Alle Wissensbeiträge sind kostenfrei und rund um die Uhr auf dem Portal von SIGS DATACOM verfügbar und werden ständig für Sie aktualisiert.

Überzeugen Sie sich selbst unter  
[www.sigs-datacom.de/wissen](http://www.sigs-datacom.de/wissen)

**SIGS DATACOM**  
FACHINFORMATIONEN FÜR IT-PROFESSIONALS

SIGS DATACOM WISSEN – Die große Wissensbibliothek für IT-Professionals

# Abhängigkeiten erkennen – Code-Analyse mit Rational Asset Analyzer

*Schlechte Wartbarkeit, hohe Komplexität, zu wenig Dokumentation und redundanter Code sind bekannte Probleme bei historisch gewachsenen Anwendungen. Um solche Probleme besser in den Griff zu bekommen gibt es Werkzeuge, die helfen, diese monolithischen Anwendungen in Bezug auf Abhängigkeiten zu analysieren und die Ergebnisse über grafische Darstellungen greifbarer aufzubereiten. Eine Möglichkeit dazu liefert Rational Asset Analyzer, RAA.*

Bei der Analyse von Anwendungen unterscheidet man zunächst zwei Teilbereiche, zum einen die statische Code-Analyse und zum anderen die Laufzeitanalyse. Bei der statischen Analyse wird der Quellcode der Anwendungen benötigt und analysiert, bei der Laufzeitanalyse wird die Anwendung während des Betriebs untersucht. RAA legt hier einen Schwerpunkt auf die statische Code-Analyse. Für die folgenden Einsatzgebiete kann RAA eingesetzt werden:

- Für Mainframe und distributed Anwendungen in Cobol, PL/I, Assembler, Java EE, C/C++
- Als zentrales erweiterbares Repository mit Metadaten über Anwendungen
- Zur Verbesserung des Anwendungsverständnisses durch grafische und tabellarische Aufbereitung
- Zur Unterstützung bei Aufwandsschätzungen von Code Änderungen
- Zur Kontrolle von Komplexität der Anwendungen durch die Erfassung von Industriemetriken z. B. Halstead, McCabe
- Zur Erstellung von Dokumentation
- Zur Unterstützung des Entwicklungsprozesses als Integration in eine Eclipse IDE
- Als Basis für Modernisierungsprojekte
- Zur Optimierung von Code durch Erfassung von Dead Code

## Informationen greifbar aufbereitet

RAA liefert zum einen eine Browseroberfläche und kann zudem die Metriken und erkannten Abhängigkeiten in Form von Tabellen oder skalierbaren Vektorgrafiken (SVG) anzeigen. Das Datenbankmodell von RAA ist als Entity Relationship Model (ERM) in der Online Hilfe verfügbar und kann bei Bedarf erweitert werden. Es ist möglich, benutzerdefinierte Abfragen über Beziehungen von Assets über das gesamte Datenmodell mithilfe von Custom Queries zu realisieren. Diese können über einen Web-basierten Assistenten erstellt werden und als XML Format ausgetauscht werden. Alternativ zur Browser Oberfläche gibt es eine Integration in die Eclipse Oberfläche in Verbindung mit dem Werkzeug Rational Developer for System z, RDz.

Um eine neue Anwendung analysieren zu können, muss der entsprechende Quellcode der Anwendung zur Verfügung stehen. Dies kann z. B. ein Verzeichnis mit der entsprechenden Ordnerstruktur auf Windows oder AIX sein, oder ein PDS (Partitioned Dataset) auf dem Mainframe. Über drei Arten können Sourcen dem Repository neu hinzugefügt werden:

1. Initialer Ladevorgang aller benötigten Sourcen,
2. Partieller Ladevorgang bei Bedarf,
3. Inkrementeller Ladevorgang.

Der inkrementelle Ladevorgang kann soweit automatisiert werden, dass bei Code-Änderungen und einer Neu-Umwandlung des Programms genau dieses neu analysiert wird. Dabei bleibt das RAA Repository synchron mit dem aktuellen Stand der Anwendung. RAA kann optional auch Subsystem Information, z. B. für IMS (Information Management System), DB2, WebSphere MQ und WebSphere Application Server auswerten. Dabei steigt die Detailtiefe an Informationen über die Assets.

Nach einer erfolgreichen Analyse der Programme, Satzstrukturen (Copy Books oder Includes) werden die Meta-Informationen in eine DB2 Datenbank geladen. Im letzten Schritt werden die Bausteine analysiert, nach Typ, Sprache, Querverbindungen und in logische Einheiten gruppiert, z. B. eine Run Unit. Eine Run Unit besteht aus einem Hauptprogramm und allen Unterprogrammen der gesamten Aufrufkette.

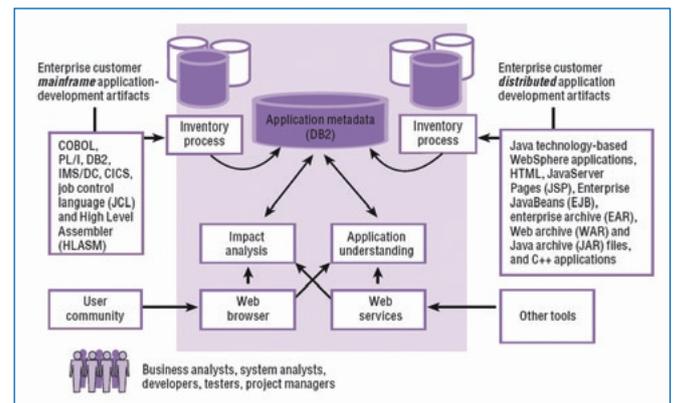


Abb. Übersicht von RAA

RAA liefert verschiedene Schnittstellen, um an die Metainformationen zu kommen: zum einem über Web Browser, über eine Web-Service-Schnittstelle zur Anbindung von externen Werkzeugen oder über eine SQL Schnittstelle direkt.

Über die Integration mit dem Entwicklungswerkzeug RDz kann der Entwickler während der Programmentwicklung die Visualisierungs- und Auswertungsmöglichkeiten von RAA nutzen.

## Fazit

Rational Asset Analyzer liefert einen transparenteren Durchblick in komplexe Anwendungen und kann über verschiedene Analysemöglichkeiten die Anwendungsentwicklung beschleunigen. Risiken bei Änderungen werden planbarer. Gerade bei Modernisierungsprojekten und bei dem Wunsch nach mehr Transparenz bei Outsourcing von Anwendungen an externe Dienstleister leistet RAA gute Dienste. ■

### Benjamin Storz

ist akkreditierter IT Specialist bei IBM und beschäftigt sich mit dem Thema Anwendungsentwicklung und -modernisierung für Mainframe Systeme. (benjamin.storz@de.ibm.com)

# Save Our Assets – Alte Systeme im neuen Gewand

*Save Our Assets – das ist das Motto der meisten Unternehmen bei der Einführung einer Service-orientierten Architektur (SOA). Die im Unternehmen bestehenden, heterogenen Anwendungslandschaften sind meist über viele Jahre gewachsen. Sie beinhalten die missionskritischen Applikationen als Herzstück für das tägliche Geschäft. Diese Anwendungen sind oft groß und komplex und bilden wichtige Funktionalitäten ab, deren Neuentwicklung sehr kosten- und zeitintensiv wäre. Zudem ist die Dokumentation meist nicht vollständig und veraltet, teilweise gibt es kaum noch Mitarbeiter, die sich en detail mit der Struktur und dem Code der Anwendungen auskennen. Um die Investitionen in diese Anwendungen zu schützen, stellt sich daher für jedes Unternehmen an einem gewissen Punkt die Frage: Wie integriere ich meine bestehenden Applikationen in die Service-orientierte Anwendungslandschaft? Wie mache ich die dort verfügbare Funktionalität und Daten innerhalb neuer SOA-basierter Anwendungen nutzbar?*

Beim nicht-invasiven Service Enablement nimmt man Code-Anpassungen in der Regel nur vor, um die technischen Möglichkeiten für die Bereitstellung eines Service zu schaffen, z. B. das MQ Enablement einer Anwendung für das IBM IMS (Information Management System) oder CICS (Customer Information Control System). Es geht dabei nicht um Neuentwicklung von Funktionalität, Renovierung bestehender Anwendungen (Application Renovation/Conversion) oder Transformation auf moderne Plattformen und Programmiersprachen (Migration) (siehe Artikel „Genauso wie früher, nur besser“, S. 19). Migration oder Renovierung können jedoch mitunter die Voraussetzung sein, überhaupt klar definierte Service-Schnittstellen bereitstellen zu können.

Durch Service Enablement werden Reichweite und Lebenszyklus von Altanwendungen erweitert bzw. verlängert. Es lassen sich prinzipiell drei Arten unterscheiden:

- UI-Wrapping oder Screen Scraping – Screens auf moderne grafische oder Web-Service-Schnittstellen mappen. Dies ist mit aktuellen Tools relativ einfach durchführbar, jedoch nur sinnvoll bei UI-intensiven Applikationen.
- Data-Wrapping – neue Schnittstellen für Legacy-Datenstrukturen entwickeln, um direkten Zugriff per SQL oder XML-Technologien zu ermöglichen.
- Business-Logic-Wrapping – auf vorhandene Funktionen via individuell entwickelter Wrapper, Adapter oder (Web-)Service-Schnittstellen zugreifen. Moderne Plattformen wie Java EE und .NET, aber auch z. B. CICS bieten Unterstützung für Services out-of-the-box.

Ein Beispiel für das Business-Logic-Wrapping findet sich in Abb. 1. Hier wird eine IMS-Anwendung via SOAP Gateway angebunden.

Für eine direkte Integration sind Altanwendungen oft schlecht geeignet. So zum Beispiel, wenn keine saubere Trennung zwischen Präsentationslogik und Businesslogik existiert oder wenn die Granularität der Schnittstellen nicht den Anforderungen an eine Service-Schnittstelle aus Geschäftssicht genügt.

Zudem gibt es zahlreiche nicht-funktionale Faktoren, die beim Service Enablement unbedingt zu berücksichtigen sind. Dazu zählen Aspekte wie die lose Kopplung, Sicherheit (z. B. nötige Zugriffsberechtigungen), Verfügbarkeit (Batchläufe, Wartungsfenster etc.), Transaktionalität, implizite Abhängigkeiten bestimmter Funktionen untereinander (keine Zustandslosigkeit, keine Kontextfreiheit), Performance, Bedeutungen von Fehlern und Fehler-Codes und vieles mehr. Außerdem besteht beim naiven Service Enablement von Legacy-Schnittstellen die Gefahr, dass ein Wildwuchs von Service-Schnittstellen entsteht. Eine strukturierte Vorgehensweise beim Service Enablement ist daher essentiell.

IBM bietet hierfür eine Full-Lifecycle-Methode für die Anwendungsentwicklung an – die SOMA (Service-Oriented Modeling and Architecture) Methode (siehe [SOMA1], [SOMA2]). SOMA beinhaltet Techniken, die dabei helfen basierend auf geschäftlichen Anforderungen (top-down), Geschäftszielen und der Analyse bestehender Systeme (bottom-up) (siehe Artikel „Die Wahrheit liegt im Code“, S. 11) die relevanten Prozesse, Systeme und Services zu bestimmen und z. B. die richtige Granularität für die Service Interfaces zu ermitteln (siehe Abb. 2). Zunächst werden aus Geschäftssicht sinnvolle Service-Kandidaten ermittelt. Gleichzeitig findet eine detaillierte Analyse der relevanten Systeme für die Realisierung inklusive der Dokumentation von Architektur und Design, Regeln, Datenmodellen und nicht-funktionaler Parameter statt, sowie das Mapping der Anwendungen auf Prozesse und der Funktionen auf Service-Kandidaten. Die hierbei gewonnenen Erkenntnisse helfen bei den Entscheidungen für die Realisierung. Für die Service-Spezifikation und Messagemodellierung geht man bei der Analyse noch weiter in die Tiefe, z. B. für das Mapping von Cobol Copybooks auf XML. Mittels geeignetem Tooling lassen sich für die Implementierung Code-Artefakte (WSDL, XSD, Java) aus den Service-Modellen generieren.

Um eine Verbindung zu einem Backendsystem zu erstellen bzw. diesem auch die Möglichkeit zu geben, eine Verbindung in die Außenwelt aufzubauen, gibt es zahlreiche Möglichkeiten, wobei ein zunehmender Trend in Richtung offener Standards zu beobachten ist. Stark vereinfacht lassen sich die unterschiedlichen Optionen in die Kategorien Standardarchitek-

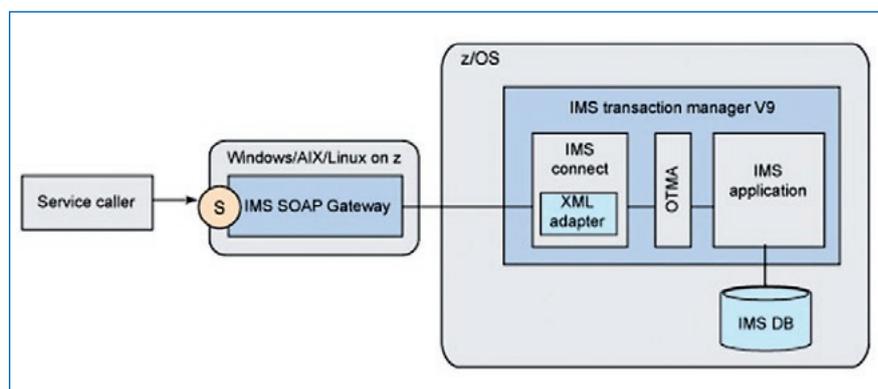


Abb. 1 Beispiel für das Enablement einer IMS-Anwendung via IMS SOAP Gateway (Quelle: <http://www.ibm.com/developerworks/library/ar-datapow/>)

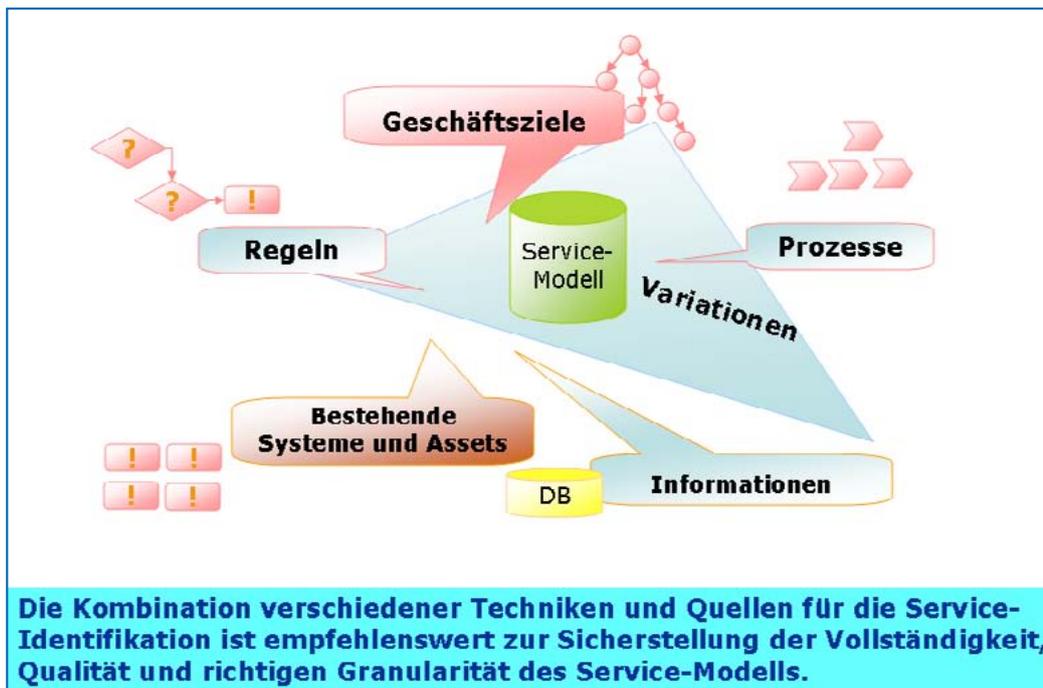


Abb. 2 Service-Identifikation

tur und Standardtransport unterteilen (siehe [CICS]). Die erste Kategorie stellt umfassende Entwicklungswerkzeuge und Laufzeitunterstützung bereit, während die zweite geeignet ist für Anwendungen, die größere Kontrolle über das Protokoll benötigen und nicht die Werkzeuge und Service-Qualitäten der Architekturen benötigen. Allerdings setzt dies auch eigene Verantwortlichkeit für Systemverwaltung, Sicherheit und Wiederherstellung voraus.

Im Transportbereich sind Protokolle wie HTTP, TCP/IP und die asynchrone Anbindung über Queues angesiedelt (unter Java als Implementierung der Java Messaging Architecture). Client und Server sind mittel bis eng gekoppelt, was Einfluss auf die Flexibilität haben kann und oft mit Code-Anpassungen einhergeht.

Im Gegensatz dazu bieten Standardarchitekturen mittlere bis lose Kopplung zwischen Client und Server, mit Web Service und Java Connector Architektur (JCA) als wichtigsten Vertretern.

Connectoren sind bereits seit einiger Zeit im Einsatz und haben im Fall eines Java Clients die Möglichkeit, standardisiert mit einem Resource Adapter zu kommunizieren, der anschließend die spezifische Verbindung mit dem Backend übernimmt. Da hierbei reine Nutzdaten übertragen werden, stuft man die Kopplung als mittel ein. Connectoren können ausschließlich in eine Richtung verwendet werden – das Backend kann sie nicht nutzen, um selbst mit der Außenwelt zu kommunizieren. Bekannte Vertreter, die diese Architektur implementieren, sind CICS Transaction Gateway, IMS Connect oder DB2 Connect.

Im Gegensatz dazu werden die Nutzdaten in einer Web Service Architektur mittels XML übertragen, was eine vollständige Entkopplung von Requester und Provider ermöglicht. Je nach Backend wird auch eine wachsende Anzahl der WS-\* Standards, die das Web Service Rahmenwerk darstellen, unterstützt, beispielsweise WS-Security, WS-AtomicTransactions, WS-Adressing.

IMS unterstützt den Web Service-Standard in einem zweistufigen Ansatz über das IMS SOAP Gateway, welches den Web Service Endpunkt auf einen J2EE Application Server vorlagert und von dort über einen Connector ins Backend weiterführt. Der CICS Web Service Support siedelt den Endpunkt direkt im CICS an. Beide Möglichkeiten erfordern die Generierung von Artefakten, die das Backend zur Laufzeit bei der Umsetzung zwischen XML und Sprachstrukturen wie Cobol oder PLI unterstützen. Hierzu existieren batch- oder eclipse-basierte Werkzeuge.

Welche der genannten Optionen im Einzelfall geeignet ist, erschließt

sich aus einem Vergleich der einzelnen Qualitäten und Einschränkungen mit den geforderten Leistungsmerkmalen einer Anbindung. Hauptkriterium für den Einsatz von Web Services ist meist, ob wirklich eine vollständige Entkopplung von Requester und Provider notwendig ist, um den Mehraufwand der XML-Konvertierung zu rechtfertigen.

Ein Beispiel hierfür ist die Realisierung bei der Deutschen Angestellten Krankenkasse (DAK). DAK unterhielt zwei Transaktionssysteme, IMS (200.000 Transaktionen pro Tag) und CICS (11 Millionen Transaktionen pro Tag) und entschied sich, sämtliche Transaktionslast im CICS zu vereinen. Allerdings musste den bisherigen Nutzern, die über Connectoren an das IMS angebunden waren, eine Zugriffsmöglichkeit ins CICS gegeben werden. Es galt, Visual Basic, C/C++ und stand-alone Java Clients mit dem CICS zu verbinden, wobei sich schnell eine lose gekoppelte Lösung herauskristallisierte. Heute beantwortet die Web-Service-Schnittstelle mehr als 200.000 Anfragen am Tag und nach diesem Erfolg wurden Web Services sogar als Kommunikationsmittel für CICS als Requester nach außen etabliert.

## Web Service Enablement mit Data Web Services (DWS)

Data Web Services (DWS) (siehe [DWS]) ist eine Entwicklungs- und Laufzeitumgebung, mit der Sie denkbar einfach einen Web Services-basierten Zugriff auf Ihre DB2-Datenbank realisieren können. Aus existierenden DML-Befehlen (SELECT, INSERT, UPDATE, DELETE und Aufruf von Stored Procedures) entsteht innerhalb kürzester Zeit und ohne Programmieraufwand eine deploymentfähige JEE-Anwendung.

DWS unterstützt derzeit sechs Protokolle (SOAP/HTTP, SOAP/JMS und vier REST-Varianten). Als Laufzeitumgebung unterstützt das Tool neben WebSphere Application Server (inklusive der kostenfreien Community Edition) und WebSphere Data Power auch Apache Tomcat.

## SQL WS-fähig machen

Das wohl simpelste Szenario für den Einsatz von DWS ist folgendes: Eine bestehende Tabelle als Web Service zu exponieren, indem man die typischen CRUD-Operationen (also INSERT, SELECT, UPDATE und DELETE) erstellt und dann daraus einen Web Service generiert.

Die Einsatzmöglichkeiten für dieses Szenario sind allerdings eher begrenzt. Es fordert nämlich fast dazu heraus, einen Kardinalfehler bei SOA-Implementierungen zu begehen, nämlich die Granularität der Services zu fein zu wählen. Dennoch gibt es einige sinnvolle Einsatzmöglichkeiten. Mögliche Beispiele sind zentrale DB2-Tabellen mit unternehmensweit gültigen Bewegungsdaten (z. B. Zinssätze, BLZ-Verzeichnis etc.). Diese können mit extrem geringem Aufwand als unternehmensweiter Service zur Verfügung gestellt werden.

## Stored Procedures als Web Services kapseln

Die Möglichkeit, SQL als Web Service exponieren zu können, ist zwar „nice to have“. Viel interessanter aus Sicht des Enterprise-Architekten ist aber die Möglichkeit, auch Stored Procedures als Service zur Verfügung zu stellen. Warum? Weil es ja gerade eine der Vorzüge von Stored Procedures ist, Anwendungslogik zu kapseln und möglichst nahe an der Datenhaltung zu haben. Durch Web Services Enablement wird dieser Gedanke lediglich noch einen Schritt weiter gedacht – die Dienstanwender sind nicht mehr auf Client-Software und SQL-Programmierung angewiesen. Besonders vorteilhaft ist dies, seit mit DB2 Version 9 die Möglichkeiten der Stored-Procedure-Entwicklung drastisch vereinfacht wurden (Stichwort „Native Stored Procedures“).

## Standardmodule via DB2 als Web Services exponieren

Wohl jeder Mainframe-Shop hat einen Satz von Standardmodulen, die von etlichen Anwendungen benutzt werden, da sie übergreifende Funktionalitäten anbieten. Beispiele aus dem Bankbereich sind Prüfsummenroutinen, Feiertagskalender und dergleichen. „Klassisch“ werden solche Standardmodule z. B. aus Cobol-Anwendungen über dynamische Calls eingebunden.

Was liegt näher, als derartige anwendungsübergreifende Services auch plattformübergreifend zur Verfügung zu stellen? In der Praxis scheiterte dies bisher daran, dass die verfügbaren Lösungen zu komplex oder zu schwergewichtig waren. Die DWS-Funktionalität von DB2 bietet hier eine hochinteressante, weil leichtgewichtige und risikoarme Alternative.

Die Standardmodule werden als DB2 Stored Procedures gekapselt. Typischerweise braucht man dazu nur einen simplen „Wrapper“ – die Module sind ja gerade dafür ausgelegt, möglichst universell einsetzbar zu sein und wenige Abhängigkeiten zur Laufzeitumgebung zu haben. Im nächsten Schritt wird die Stored Procedure dann über DWS-Tooling als Web Service zugänglich gemacht.

Die Vorteile einer solchen Lösung:

- Minimal invasiv: Bestehender Code wird nicht angefasst, nur gekapselt.
- Unkompliziert: Die Kapsel um den bestehenden Code führt allenfalls Umformatierungen und Parametervalidierungen aus.

- Kein Programmieraufwand: Das Web Service Enablement geht ohne jeden Programmieraufwand mit DWS-Tooling. Es entsteht eine .war-Datei, die dann wie gewohnt auf den Application Server bereitgestellt werden kann.

## Fazit

Mittels Service Enablement von Altanwendungen lässt sich bestehende Funktionalität auf elegante und nicht-invasive Weise in neuen Service-orientierten Applikationen nutzbar machen. Es existieren für verschiedenste Technologien stets diverse technische Optionen wie die Beispiele bezüglich IMS, CICS und DB2 belegen. Essentiell für die Identifikation relevanter Service-Kandidaten ist dabei eine methodische Vorgehensweise, die die sehr technische Bottom-up-Perspektive um einen geschäftsgetriebenen Top-down-Ansatz ergänzt. ■

## Literatur/Quellen

[1] Enabling z/OS Applications for SOA, <http://www.redbooks.ibm.com/abstracts/sg247669.html>

[2] Adapting legacy systems for SOA, <http://www.ibm.com/developerworks/webservices/library/ws-soa-adaptleg/>

[SOMA1] Ali Arsanjani et al. (2008). „SOMA: A method for developing service-oriented solutions“. IBM systems Journal Oct 2008, [http://www.cs.jyu.fi/el/tjtse54\\_09/Artikelit/ArsanjaniEtAlIBMSsJ.pdf](http://www.cs.jyu.fi/el/tjtse54_09/Artikelit/ArsanjaniEtAlIBMSsJ.pdf)

[SOMA2] Ali Arsanjani (2004). „Service-Oriented Modeling & Architecture“. IBM Online article, 09 Nov 2004, <http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/>

[CICS] Mark Cocker (2007). „Options for integrating CICS applications in an SOA“. IBM White paper, Sep 2007, [ftp://ftp.software.ibm.com/software/http/cics/tserver/v32/library/WSW11339-USEN-00\\_CICS\\_Web\\_svs\\_wp\\_V3.pdf](ftp://ftp.software.ibm.com/software/http/cics/tserver/v32/library/WSW11339-USEN-00_CICS_Web_svs_wp_V3.pdf)

[DWS] Data Web Services on DB2 for z/OS, <http://www.ibm.com/developerworks/data/library/techarticle/dm-0905db2zosstoredprocedures/>

### Isabel Arnold

ist als IT Specialist für CICS und moderne Anwendungsentwicklung im Technical Presales tätig. ([isabel.arnold@de.ibm.com](mailto:isabel.arnold@de.ibm.com))

### Bernhard Meyer-Willner

beschäftigt sich als IT Architect bei IBM Global Business Services mit Service-orientierter Modellierung, Architektur und Integration. ([bernhard.meyer-willner@de.ibm.com](mailto:bernhard.meyer-willner@de.ibm.com))

### Ulrich Seelbach

beschäftigt sich bei IBM Global Business Services seit über zehn Jahren mit Anwendungsmodernisierung, insbesondere mit Java-Technologie unter z/OS. ([ulrich.seelbach@de.ibm.com](mailto:ulrich.seelbach@de.ibm.com))

# HATS – zwei Funktionen unter einem Hut

Host-Anwendungsmodernisierung über die Bildschirmschnittstelle. Bedingt durch die derzeitige wirtschaftliche Lage stehen Neuentwicklungen komplexer IT-Anwendungen unter sehr großem Kosten- und Rechtfertigungsdruck. In dieser finanziellen Situation kann es sinnvoll sein, existierende unternehmenskritische Anwendungen so zu modernisieren, dass die meist host-basierten Anwendungen bestehen bleiben, der Zugriff darauf allerdings verändert bzw. erweitert wird. Sollten diese Anwendungen Bildschirm-basiert (3270 oder 5250, „Green-Screen“) sein, stellt IBM mit Host Access Transformation Services (HATS) ein Werkzeug zur Verfügung, welches den Wert der etablierten Anwendungen erhält, eine grafische Aufwertung erlaubt, aber auch die Integration in neue J2EE Umgebungen ermöglicht.

## Erstellung einer modernen Benutzer-Oberfläche mit intuitiven Web-Elementen

Die erste und wohl bekanntere Funktion von HATS ist die automatische Umsetzung von Telnet 3270/5250-Bildschirmen in HTML. Die Umsetzung erfolgt dynamisch aufgrund von Regelsätzen, welche mit dem HATS-Toolkit erstellt werden. Dieses interaktive Entwicklungswerkzeug ist ein Eclipse-Plugin für IBM Entwicklungsumgebungen wie RAD (Rational Application Developer), RDz (Rational Developer for System z), RDi (Rational Developer for i) etc. und kann von IBM Internetseiten heruntergeladen werden.

Neben einer quasi Eins-zu-Eins-Umsetzung sind auch umfangreiche erweiterte Anpassungen möglich. Hier bieten sich nahezu alle grafischen Möglichkeiten, die HTML, JAVA Skript und Cascading Stylesheets bieten. Von Buttons über Links und Pull-Down-Menüs zu Popup-Fenstern – mit dem Toolkit können Grafiken wie auch Makrofunktionen einfach eingebunden, erweitert und angepasst werden. Logische Funktionserweiterungen sowie der Zugriff auf andere Systeme, z. B. über JDBC-Calls, können mit Hilfe von einbindbarer JAVA-Logik vorgenommen werden. Das Toolkit bietet zur Erstellung geeignete „Wizards“ (Hilfswerkzeuge).

Als Laufzeitumgebung für die erstellten HATS-Anwendungen werden IBM WebSphere Application Server und IBM Portal Server unterstützt. Mit diesem Aufbau ist es nun möglich, mit einem Standard-Browser – aber auch mit einem Smartphone (Mobile Device) – auf bestehende Hostanwendungen zuzu-



Abb. 1 HATS Graphical User Interface

greifen. Der Zugriff von sogenannten „Rich Clients“, die die Eclipse RCP-Plattform als Basis für Client-Anwendungen verwendet, wird ebenfalls unterstützt. Hier lassen sich Integrationen mit typischen Client-Anwendungen wie Texteditoren- und Spreadsheet-Anwendungen besonders gut realisieren.

## Erstellung von Web Services

Die zweite, weniger bekannte HATS-Funktionalität ist die Möglichkeit, Web Services zu erstellen. Diese Web Services stehen als wieder verwendbare Komponenten in einer größeren integrierten Anwendung oder in einer Service-orientierten Architektur (SOA) zur Verfügung. Der Einsatz erfordert kein Reengineering der Anwendung oder keinen Zugriff auf den Quellcode. Die Anwendung kann in gewohnter Weise weiter verwendet werden. Zur Erstellung eines Web Services werden lediglich Verständnis für die Business-Anwendung sowie Kenntnisse im Umgang mit dem Werkzeug benötigt.

Das folgende Beispiel beschreibt die Vorgehensweise: Eine wiederkehrende Tätigkeit, beispielsweise das Anlegen einer Kundennummer in der Hostanwendung, soll als Web Service zur Verfügung gestellt werden. Hierzu meldet man sich über das HATS-Toolkit in der Anwendung an. Die für das Anlegen der Kundennummer notwendigen Schritte, wie das Auswählen von Bildschirmen, die Eingabe von Daten, aber auch das Auslesen von Informationen, werden in einem Makro aufgenommen. Dieses Makro kann man anschließend mit Hilfe eines integrierten graphischen Editors testen und beispielsweise um Fehlerbehandlungen erweitern. Nach erfolgreichem Test bietet das Toolkit die Möglichkeit, aus diesem Makro ein Integrationsobjekt und daraus einen Web Service zu erstellen. Die generierte WSDL-Datei kann man daraufhin mit einem integrierten Testwerkzeug überprüfen.

So ist es möglich, innerhalb kurzer Zeit Web Services auf Basis von Hostanwendungen zu generieren, ohne deren Code anzufassen; eine speziell für recht alte Anwendungen sehr einfache und schnelle Methode.

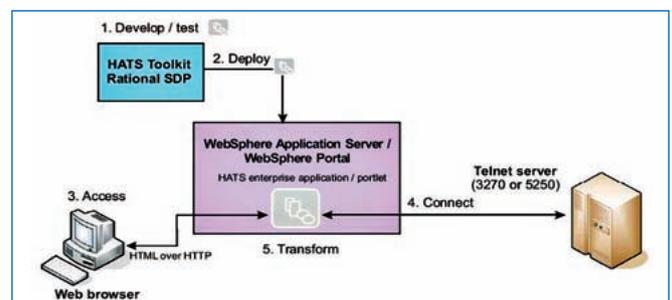


Abb. 2 HATS Architecture

## Fazit

Zusammenfassend kann man sagen: HATS ist für System z und IBM i-Kunden interessant, die ihre Host-Anwendungen über Terminal-Emulatoren betreiben und die planen, eine modernisierte Version dieser „Green-Screen“-Anwendungen im „Web“, im Portal, auf einem „Rich Client“ (RCP) oder einem „Mobile Device“ verfügbar zu machen. Die Fähigkeit von HATS, Daten von verschiedenen „Backend“-Anwendungen und Systemen zu kombinieren sowie Teile einer Terminal-Anwendung als Web Services zu verwenden, eröffnet weiteres Modernisierungspotential.

**Ralf Pflug**

ist Master Certified IT Specialist  
(ralf.pflug@de.ibm.com)

# Genauso wie früher, nur besser

„Genauso wie früher, nur besser“ – Mit diesen Worten beginnt oft die Anforderungsspezifikation eines Kunden, wenn es um die Ablösung einer Altanwendung geht. In der Regel versteckt sich dahinter der Wunsch, eine in die Jahre gekommene, aber funktional korrekte Anwendung zu modernisieren, d. h. flexibler und besser wartbar zu machen. Eine Modernisierung kann wie beim Hausbau nur kleinere Reparaturen bedeuten oder einen kompletten Neubau.

Den Kern der IT-Landschaft vieler Unternehmen bilden oft geschäftskritische Anwendungen, die über Jahrzehnte gewachsen sind und mannigfaltige Umorganisationen überlebt haben. Das so genannte „Wachstum“ der Anwendung hat in vielen Fällen zu einer schwer beherrschbaren Komplexität und kaum mehr vorhandenen Erweiterungsmöglichkeiten geführt. In vielen Fällen sind keine fachlichen Spezifikationen vorhanden, mit Ausnahme der im Source Code vorhandenen Kommentare. Die vor Jahrzehnten verfügbaren Programmiertechnologien wie Cobol oder C oder systemgetriebene Restriktionen machen es schwierig, den Code zu verstehen oder sogar zu erweitern. Hinzu kommt, dass die Wissensträger für die Anwendung oft „im Rentenalter“ sind und die jüngeren Kollegen wenig Interesse an Cobol- oder C-Programmierung zeigen. Alle genannten Problemstellungen sind in der Regel Ausgangspunkt für Anwendungsmodernisierungsprojekte. Randbedingung der Projekte ist mehrheitlich, dass kein Aufwand auf Fachbereichsseite und auch möglichst wenig Transformationsaufwand in der Entwicklung entstehen darf.

## Transformationsszenarien

Welcher Weg bei der Modernisierung der Beste ist, ergibt sich oft erst nach der detaillierten Analyse der Altanwendung (siehe Artikel „Die Wahrheit liegt im Code“, S. 11) und der Festlegung der Zielarchitektur. Das Transformationsszenario hängt von der Tiefe des Modernisierungsvorhabens ab. Folgende drei Transformationsszenarien stehen exemplarisch für die Herausforderungen und bereits erprobte Lösungsansätze zur Unterstützung des Transformationsprozesses (siehe Abb. 1).

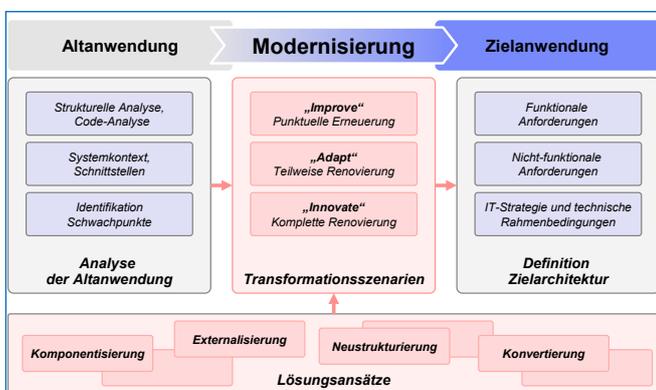


Abb. 1 Anwendungs-Modernisierung und Transformationsszenarien

## „Improve“ - Punktuelle Erneuerung

Abgeschlossene Module einer Anwendung sind auf eine neue technologische Basis umzustellen und zu erweitern. Im Haus-

bau wäre ein Beispiel der Ersatz einer Heizungsanlage. Anforderung ist dabei, möglichst minimal-invasiv vorzugehen. Fraglich ist dabei, ob die relevanten Module wirklich sauber abzugrenzen sind und die technologische Basis die Integration neuer Technologien erlaubt.

## „Innovate“ - Komplettsanierung oder Neubau

Die Entscheidung für eine komplette Renovierung wird in vielen Fällen aufgrund von technischen Kriterien oder Ereignissen gefällt, beispielsweise durch die Abkündigung von Programmiersprachen. Ziel ist es nicht, den bestehenden Code 1:1 in eine neue Programmiersprache zu übertragen, sondern durch eine strukturelle Umorganisation eine zukunftsfähige Anwendung mit größerer Flexibilität und besserer Wartbarkeit zu erhalten.

## „Adapt“ - Teilweise Renovierung

Als Misch-Szenario ist das Szenario einer Teilrenovierung zu nennen. Im Gegensatz zur punktuellen Renovierung sind größere Teile der Anwendung zu überarbeiten, jedoch ist keine vollständige Restrukturierung erwünscht bzw. leistbar. Herausforderungen liegen in der technischen Integration und den Abhängigkeiten der Module der Anwendung. Ein Risiko ist zudem, dass aus der punktuellen Erneuerung eine ausgewachsene Komplettsanierung wird.

## Lösungsansätze

Für Modernisierungsmaßnahmen ist eine Vielzahl von Verfahren und Technologien im Markt verfügbar. Im Blick über verschiedenartige Anwendungsmodernisierungsprojekte wiederholen sich jedoch einige Erfahrungen und kristallisieren sich Lösungsansätze heraus, die relativ unabhängig von den Grundtechnologien sind. Im Folgenden beschreiben wir beispielhafte Lösungsansätze, um möglichst schnell einen tragfähigen Entwurf der Renovierungsmaßnahme mit einer guten technischen Umsetzung zu finden.

## Lösungsansatz 1: Neustrukturierung

Im Szenario „Innovate“ wird ein bestehendes System unter Nutzung einer neuen Technologie vollständig abgelöst. In vielen Fällen sind solche Projekte technisch motiviert, so dass der Fachbereich die Vorgabe „fachlich wie das Altsystem“ als durchaus ausreichende Fachspezifikation empfindet. Erfahrungsgemäß besteht jedoch gerade hier eine Notwendigkeit zur fachlichen Mitarbeit, da oftmals die vermeintliche Fachlogik im Wust von jahrzehntelanger, prozeduraler Programmierung und aufgrund der technischen Restriktionen alter Programmiersprachen schwer erkennbar ist. Durch den Einsatz moderner Programmiersprachen sind viele Logiken wesentlich einfacher umsetzbar, wie zum Beispiel der wahlfreie Zugriff auf Datensätze in Tabellen- oder Listenstrukturen in Java im Vergleich zu Speicherstrukturen in C. Eine 1:1 Umsetzung der Logik des Altsystems ist aus diesen Gründen mit einer fachlichen Spezifikation zu verbinden und sollte nicht durch Kopieren der Konditionallogik aus dem Altsystem passieren.

Neben der fachlichen Konzeption ist der Entwurf einer tragfähigen Anwendungsarchitektur der Schlüssel für eine zukunfts-

fähige und flexible Anwendung. Diese leitet sich aus den technologischen Rahmenbedingungen und den funktionalen wie nicht-funktionalen Anforderungen ab. Es sei hier auf einige, aus unserer Erfahrung wichtige Grundprinzipien verwiesen:

- Anwendung Service-orientierter Paradigmen (siehe Artikel „Save Our Assets – Alte Systeme im neuen Gewand“, S. 15) zur Erreichung einer guten Wiederverwendbarkeit von Services
- Auswahl von standardisierten Technologien, marktgängigen Produkten und Frameworks
- Auslagerung von querschnittlichen, systemnahen Diensten aus der Fachlogik
- Definition eines sauberen Grundmusters für die Entwicklung (u. a. Struktur, Stereotypen, Layering)
- Modellgetriebene Verfahren zur Code-Generierung

Zusammenfassend ist die Kombination von fachlicher Konzeption und technischen Verfahren ausschlaggebend für die Zukunftsfähigkeit der Anwendung.

## Lösungsansatz 2: Evolutionäre Weiterentwicklung durch Komponentisierung

Viele Unternehmen ziehen einen evolutionären Ansatz einer Komplett-Renovierung vor. Kernproblem ist hierbei die starke Verflechtung in den Systemen: Obwohl nur eine Altkomponente transformiert werden soll, besteht die Gefahr, dass größere Anteile der Anwendung ebenfalls umzustellen sind. Um dieses Problem zu adressieren, sollten für den evolutionären Ansatz wichtige Grundlagen gelegt werden. Das System sollte zum einen in Komponenten, also funktional zusammengehörige Blöcke mit definierten Schnittstellen, strukturiert werden. Grundlage für die Kommunikation zwischen den Komponenten sollte ein Integrationsverfahren sein, das bidirektional in alter als auch in neuer Technologie nutzbar ist. Ein produktgestütztes Beispiel ist die Channel/ Container Kommunikation in einer CICS TS® Laufzeitumgebung. Bei gleichbleibender Schnittstelle kann eine Cobol-Komponente in Java-Technologie überführt werden. Angrenzende Cobol-Module können die renovierte Komponente transparent nutzen, gleichzeitig kann die Java-Komponente andere Cobol-Komponenten nutzen. Durch das bidirektionale Integrationsverfahren wird es nicht notwendig, die von der Java-Komponente genutzten Module ebenfalls in Java zu transformieren (siehe Abb. 2).

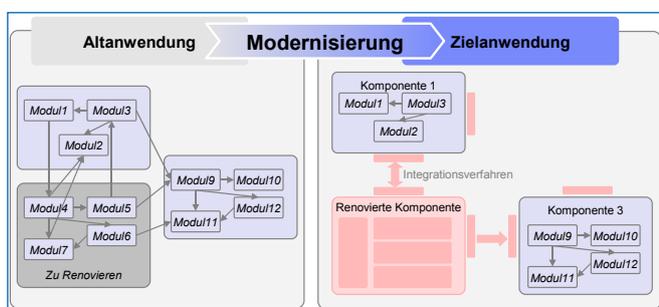


Abb. 2 Weiterentwicklung durch Komponentisierung

## Lösungsansatz 3: Extraktion und Externalisierung

Das primäre Ziel der Extraktion und Externalisierung ist das Herauslösen und Gruppieren von bestimmten Funktionen aus der Altanwendung, die eine hohe Analogie aufweisen und quer im

Source Code verteilt sind. Dies können z. B. Geschäftsregeln, Transformationsfunktionen oder Funktionen zur Prozesssteuerung sein. Die Gruppierung kann durch Ansiedelung dieser Funktionen in einer neuen Anwendungsschicht (z. B. Integrationsschicht), in einem Subsystem oder über Einsatz von dedizierten Laufzeitumgebungen realisiert werden. Letzteres soll hier exemplarisch anhand von Geschäftsregeln demonstriert werden.

Geschäftsregeln stellen wichtige Steuerungsinstrumente in der Ablauflogik einer Anwendung dar. In vielen Altanwendungen sind diese Geschäftsregeln stark verstreut und zum Teil redundant implementiert. In der Regel ist die Änderungsfrequenz von Geschäftsregeln hoch und zieht somit häufige Code-Anpassungen nach sich. Die Idee besteht nun darin, diese Geschäftsregeln aus der Altanwendung im Zuge einer Transformation zu extrahieren und zentral zu halten. Diese Zentralisierung reduziert den zukünftigen Wartungsaufwand in der neuen Anwendung und erhöht zugleich die Wiederverwendbarkeit der Geschäftsregeln. In der Luxusausführung werden Geschäftsregeln in einer softwareproduktspezifischen Sprache definiert und in einer Business Rule Engine als Laufzeitumgebung ausgeführt (z. B. iLog JRules oder Drools). Erfahrungsgemäß sind bei einer Extraktion und Externalisierung mehrere Module und Subsysteme der Altanwendung betroffen. Daher eignet sich dieser Lösungsansatz für die Transformationsszenarien „Adapt“ und „Innovate“.

## Lösungsansatz 4: Tool-unterstützte Konvertierung

Die Tool-unterstützte Konvertierung von Source Code kann den Aufwand in einem Transformationvorhaben erheblich reduzieren. Das Prinzip lässt sich vereinfacht in folgenden Schritten beschreiben. Im ersten Schritt werden Source Code (z. B. RPG) und Datenobjekte einer Altanwendung mit Hilfe eines Tools analysiert und in eine Metasprache überführt, etwa Enterprise Generation Language (EGL). Implizit kann hierbei auch eine automatisierte Dokumentation einhergehen. Liegt die Anwendung einmal in EGL vor, kann von dort aus leicht in moderne Sprachen wie Java automatisiert konvertiert werden. Zudem lässt sich die Tool-unterstützte Konvertierung sehr gut mit der Extraktion und Externalisierung von Funktionen kombinieren. Der Automatisierungsgrad hängt hierbei sehr stark von der Strukturierungsgüte der Altanwendung ab. Hier sind Raten bis zu 95 % durchaus machbar. Eine Tool-unterstützte Konvertierung eignet sich aufgrund der Eingriffstiefe für das Transformationsszenario „Innovate“.

## Fazit

Der vorliegende Artikel zeigt auf, welche Lösungsansätze und Werkzeuge in den drei beschriebenen Transformationsszenarien hilfreich sind. Diese sollen lediglich bei der Auswahl der Technologien, Verfahren und Werkzeuge helfen. Am Ende ist es jedoch wie im Hausbau: Mit guten Architekten, Statikern und Handwerkern wird die Modernisierung ein Erfolg. ■

**Cemal Cömert**

Managing Consultant / Senior IT Architect  
(cemal.coemert@de.ibm.com)

**Thomas Matal**

Senior IT Architect  
(thomas.matal@de.ibm.com)



- Standardfunktionen wie Build oder Checkin finden Einzug in die Kontextmenüs und binden bestehende Buildmechanismen und Config Management Systeme an.
- Die Fehlersuche wird durch Error-Feedback erleichtert und beschleunigt. Während man normalerweise im traditionellen Bereich nach einer Syntaxüberprüfung die Fehler im Listing herausucht und anschließend im Quellcode wieder lokalisieren und beheben muss, kann mit einer grafischen Oberfläche das Listing geparkt und direkt in die Quelle als Fehlermarkierung zurückgegeben werden, ähnlich der Standard-Fehlerbehandlung von Eclipse für Java. Diese Funktionen sind auch für traditionelle Entwickler von Vorteil.

Neben allen bereits genannten Vorteilen profitiert der Modernisierer vor allem von folgenden Funktionen:

- Um die Backends über neue, standardisierte Schnittstellen zu öffnen, ist meist die Generierung bestimmter Dateien notwendig, die beispielsweise die Konvertierung zwischen XML und Cobol steuern. Hierzu bietet die Oberfläche umfangreiche Wizards, u. a. um ein Binding zu erzeugen, das im CICS oder IMS beschreibt, wie eingehendes XML in Cobol, PLI oder C/C++ umgewandelt werden kann. Diese Funktion ist für die Erstellung von Web-Service-Schnittstellen hilfreich. Oder man kann Java Wrapper für traditionelle Sprachen erstellen, die für den Datenaustausch mittels Resource Adapter der Java Connector Architektur zum Einsatz kommen können. Für einige Modernisierungsoptionen wie die Einführung der Service Component Architecture im CICS existieren nicht einmal mehr konsolen-basierte Werkzeuge sondern ausschließlich Plugins für Eclipse.

- Noch vor den anderen beiden Typen profitiert der Modernisierer am stärksten von der Integration bestehender Werkzeuge im Backend wie z. B. das IBM Debug Tool. Dadurch ist nahtloses Debugging verteilter Anwendungen (wie J2EE Web Service Requester und CICS bzw. RPG basierende Web Service Provider) über Sprach-, Laufzeitumgebungs- und Systemgrenzen hinweg möglich.

Diesem Trend folgend werden über das Debug Tool hinaus auch andere konsolen-basierte Werkzeuge als Plugins zur Verfügung gestellt. Hierzu zählen unter anderem File Manager, Fault Analyzer und Application Performance Analyzer sowie einige CICS Tools wie CICS Performance Analyzer, CICS Interdependency Analyzer und CICS Configuration Manager. Analog dazu gibt es mit der Unterstützung durch RDi erheblich erweiterte Möglichkeiten zum Debugging von Anwendungen, die unter IBM i laufen.

Ausgestattet mit diesen integrierten Entwicklungsumgebungen sind alle Arten von Entwicklern gewappnet für den täglichen Alltag und die Zukunft als Programmierer für z/OS oder IBM i. Mit Eclipse als Basis vereinen RDi und RDz die Vorteile der grafik-orientierten und die der terminal-basierten Welt und ebnen den Weg, neue Schnittstellen und Technologien nutzen sowie Effizienz und Produktivität steigern zu können. ■

### Isabel Arnold

ist als IT Specialist für CICS und moderne Anwendungsentwicklung im Technical Presales der IBM Deutschland tätig. (isabel.arnold@de.ibm.com)

### Advertorial



projects | people | products  
profession

Q Group und Q four sind die führenden IBM Businesspartner im Bereich Enterprise Modernisation und spezialisiert auf das gesamte Rational Portfolio.

Eines der Schwerpunktthemen war in den letzten Jahren die Einführung und Integration des Rational Developer for System z (RDz).



projects | people | products  
profession

Das Hauptkriterium für eine erfolgreiche Modernisierung der Anwendungsentwicklung ist die Integration der Produkte in den AD Lifecycle. Auf Basis von individuellen und standardisierten IBM Produkterweiterungen können alle zentralen und dezentralen Prozesse und Tools auf der Eclipse-Plattform konsolidiert werden.

Daraus resultieren signifikante Kostenersparnisse durch Steigerungen der Produktivität bei gleichzeitiger Entlastung der CPU-Zeiten. In etlichen europaweiten Referenzprojekten haben Q Group und Q four Lösungen für VSE, die Integration von CCM-Systemen wie Endevor und Changeman, der Einbindung von 4 GL Sprachen und diversen anderen kundenspezifischen Anforderungen entwickelt.

Besuchen Sie uns unter [www.Qgrp.com](http://www.Qgrp.com) oder schreiben Sie uns an [Q-info@Qgrp.com](mailto:Q-info@Qgrp.com).

#### Q four GmbH

Friedrich-Ebert-Anlage 28  
D - 69117 Heidelberg  
T: +49 (0)6221 58 999 10  
F: +49 (0)6221 58 999 11  
E: [q-info@qfour.de](mailto:q-info@qfour.de)  
[www.Qfour.de](http://www.Qfour.de)

#### Q Group AG

Obere Bahnhofstrasse 13  
CH - 5407 Mellingen  
T: +41 (0)56 483 41 91  
F: +41 (0)56 483 08 80  
E: [q-info@qgrp.com](mailto:q-info@qgrp.com)  
[www.qgrp.com](http://www.qgrp.com)

# Nachhaltig modern

Enterprise Generation Language erleichtert Programmierung von Geschäftsanwendungen. Anwendungen, die stabil sein müssen, laufen häufig auf IBM System z oder IBM i und sind in Cobol oder RPG programmiert. Dagegen nutzen Geschäftsprozesse zunehmend Umgebungen, die durch SOA-Architekturen und Web 2.0-Technologien bestimmt sind. B2B Kommunikation erfolgt immer mehr in Form von (Web) Services. Die dafür erforderlichen komplexen Technologien wie XML, JSON, Java, J2EE, HTML, SQL, Ajax und JavaScript setzen zeit- und kostenaufwändige Schulungen voraus.

Eine nachhaltige Modernisierungsstrategie muss in diesem Umfeld Antworten auf vier wesentliche Fragen finden. Wie können bestehende Geschäftsdaten und Geschäftsprozesse schnell und sicher mit modernen Oberflächen verknüpft werden? Wie können vorhandene Anwendungen in eine SOA-Architektur integriert werden? Wie kann das Geschäftsverständnis der Entwickler mit Fertigkeiten bezüglich neuer Technologien verbunden werden? Welche Lösung bietet Flexibilität hinsichtlich Plattformen und Wartbarkeit bei gleichzeitiger rascher Umsetzung neuer Geschäftsanforderungen?

## Standardisierte Programmiersprache

Als essentiellen Bestandteil der Antwort darauf hat IBM die Enterprise Generation Language (EGL) entwickelt. In Fortführung der mit CSP und VaGEN begonnenen Historie ist EGL die Standard-Programmiersprache für Geschäftsanwendungen. Ähnlich wie in den 60er Jahren Cobol im Vergleich zu Assembler der einfachere und produktivere Ansatz zur Entwicklung von Geschäftsanwendungen war, bietet heute EGL den einfachen und produktiven Weg zu Geschäftsanwendungen für beliebige Plattformen und verteilte Netzwerke unter Einbindung vorhandene Anwendungen.

EGL abstrahiert technische Anforderungen und erlaubt die Entwicklung von modernen Oberflächen, Services und Geschäftslogik, ohne sich im Detail mit den für die Realisierung notwendigen Technologien auskennen zu müssen. Dabei verwendet EGL keine Virtual Machine oder Interpreter, sondern

kompiliert EGL Programmspezifikationen in die Sprachen, die die jeweiligen Plattformen unterstützen. Innerhalb der auf Eclipse basierenden EGL Entwicklungsoberfläche IBM Rational Business Developers wird die Generierung von Java, Cobol und für Web 2.0-Anwendungen auch von JavaScript und HTML unterstützt. Die Entwicklung und das Debugging komplexer Anwendungen erfolgt dabei komplett in EGL innerhalb des Rational Business Developers.

Wie sich gezeigt hat, lässt sich EGL sowohl mit prozeduralem als auch mit objektorientiertem Programmierhintergrund einfach erlernen. Im Gegensatz zu Monaten für Java oder .NET verläuft die Lernkurve für EGL in zwei bis vier Wochen bis zum produktiven Einsatz. Untersuchungen zeigen eine wesentlich höhere Produktivität für die Erstellung von Geschäftsanwendungen mit EGL (<http://www-949.ibm.com/software/rational/cafe/docs/DOC-1203>).

## Einfache Integration

EGL ist außerdem die Zielsprache für die im Rahmen der Enterprise Modernisierung angebotenen Migrationswerkzeuge wie IBM Rational Migration Extension for IBM i (RPG), Cobol, VAGen, Natural/ADABAS, PL1 oder andere. Vorhandene Geschäftslogik, Anwendungen und Daten zum Beispiel auf IBM System z und IBM i lassen sich in EGL einfach einbinden. Wizards im Rational Business Developer erlauben es, aus Datenquellen die Tabellen oder Strukturen als Data Types in Form abstrakter Records automatisch zu extrahieren. Diese können über alle EGL Programme hinweg als komplexe Variablentypen verwendet werden, in denen Validierung und Formate bereits hinterlegt sind. Die Manipulation der Daten in EGL ist mit einfachen Befehlen wie *get* oder *update* möglich und führt zur Laufzeit auf einen Zugriff auf die tatsächlichen Datenquellen. Mittels der Schlüsselwörter *service* und *call* lässt sich vorhandener Cobol- oder RPG-Code einfach einbinden und darüber hinaus in einen Service oder Web Service verwandeln. Ebenso einfach können vorhandene Javaklassen und Bibliotheken in EGL benutzt werden.

EGL basiert auf offenen Standards und kann auf jeweils aktuelle technische Standards eingehen und diese einbinden. Neben der Wahl von Eclipse als Grundlage für die Entwicklung des Rational Business Developers gehört dazu auch die ebenfalls Eclipse basierende kostenfreie EGL Community Edition, mit der Rich Internet Applications (RIA) komplett in EGL erstellt werden können. Die EGL Community Edition ist Teil der offenen Strategie der IBM. EGL selbst ist bereits zur Standardisierung eingereicht. Die offene Gemeinschaft rund um EGL trifft sich im EGL Cafe (<http://www-949.ibm.com/software/rational/cafe/community/egl>). Dort findet man neben Dokumentation, Downloads und Code-Beispielen auch kostenfreie online Trainings, Diskussionen und Erfolgsgeschichten. ■

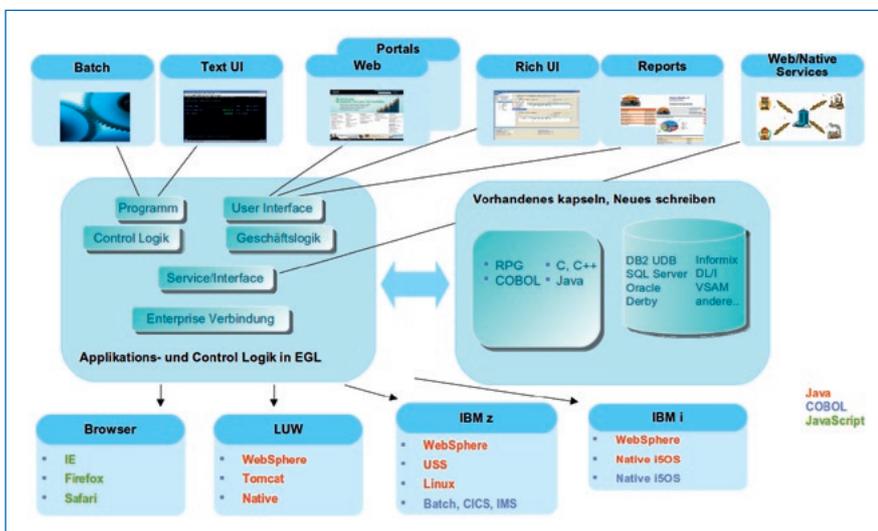


Abb. Alles aus einem Guss – EGL erzeugt Anwendungen auf beliebigen Plattformen und bindet vorhandene Daten und Geschäftslogik ein.

### Arne Bister

IBM IT Specialist  
(arne.bister@de.ibm.com)

### Mathias Tempel

IT Specialist  
(mathias.tempel@de.ibm.com)

# Mit Anwendungsmodernisierung zu mehr Datenqualität

*Ein wichtiger Aspekt bei der Modernisierung von Anwendungen sind die durch sie verwalteten Daten. Speziell die Personen-Stammdaten spielen dabei eine zentrale Rolle, weil sie von vielen anderen Anwendungen verwendet, referenziert und angereichert werden. Der folgende Artikel gibt einen Überblick, wie mit Hilfe eines Master Data Management (MDM) Systems die Qualität dieser Daten nachhaltig gesichert werden kann.*

Die zentralen Personen-Stammdaten werden typischerweise mit unterschiedlichen Frontends, durch Mitarbeiter verschiedener Abteilungen gepflegt. Dadurch unterliegen diese Daten in der Regel einer kontinuierlichen Qualitätsverschlechterung. Das hat mehrere Ursachen:

- Eine fehlende Standardisierung einzelner Felder („+49 69 1234“ bzw. „00 49 (0)69 1234“ usw.) führt zu vielfältigen Schreibweisen identischer Informationen.
- Wertebereiche sind nicht zentral definiert (z. B. „Titel“ = [„Dr.“, „Prof.“]).
- Auf Speicherverbrauch optimierte Feldlängen genügen modernen Anforderungen nicht mehr.
- Anstatt neu benötigte Felder einzuführen, werden bereits bestehende „missbraucht“.
- Freitextfelder ohne Typisierung enthalten Steuercodes zur Kompensation.
- Zentrale Vorgaben zur Nutzung bzw. Befüllung von Feldern wurden über Jahre hinweg missachtet.
- Bei Merger und Akquisitionen werden fehlende Daten nicht nachträglich erfasst sondern entweder leer gelassen oder mit Default-Werten belegt.
- Die mit der Zeit entstandenen Daten-Dubletten werden nicht bereinigt.

Wenn mehrere dieser Aspekte aufeinander treffen, führen sie zu einer kontinuierlichen Qualitätsverschlechterung. Es ist deshalb eine zentrale Aufgabe der Anwendungsmodernisierung, die Datenqualität wieder zu verbessern und Maßnahmen zu ergreifen, sie nachhaltig zu sichern.

Für den ersten Schritt bietet der Markt eine Reihe von Werkzeugen, wie z. B. den IBM InfoSphere Information Server (IIS). Auf Basis von Metadaten bietet der InfoSphere Information Server ein hohes Maß an Flexibilität bei der Integration und Aufbereitung von Informationen. Er verfügt über Datenqualitätsfunktionen, die die konsistente und exakte Übermittlung von Informationen sicherstellen, damit gesetzliche Anforderungen zuverlässiger eingehalten werden können. Zum IIS gehört IBM InfoSphere DataStage, eine skalierbare ETL-Lösung, welche Daten aus

einer Vielzahl von Quellsystemen (Text, XML, ERP-Systeme, Web Services etc.) sowohl im Batch als auch realtime verarbeitet. Diese Daten können dann wiederum über Services im Unternehmen verteilt werden. Dem voraus geht der Einsatz von IBM InfoSphere QualityStage, mit dessen Hilfe ein Profiling der bestehenden Daten erfolgen kann. Auf Basis dieser Ergebnisse werden die Daten dann entweder manuell in den Quellsystemen bereinigt oder mit Hilfe des IIS DataStage regelbasiert angepasst. IIS QualityStage bietet ausgefeilte probabilistische Verfahren zur Datenbereinigung. Ebenso gehören zum Funktionsumfang Standardisierungs-, Überprüfungs- und Anreicherungsmechanismen für Adressen weltweit.

Der Einsatz von ETL-Tools und Werkzeugen zur Verbesserung der Datenqualität erfolgt oft im Kontext einer Migration auf ein neues MDM System, welches durch die Implementierung von fachlichen Regeln auch den zweiten Schritt, die Erhaltung der Datenqualität, unterstützen kann.

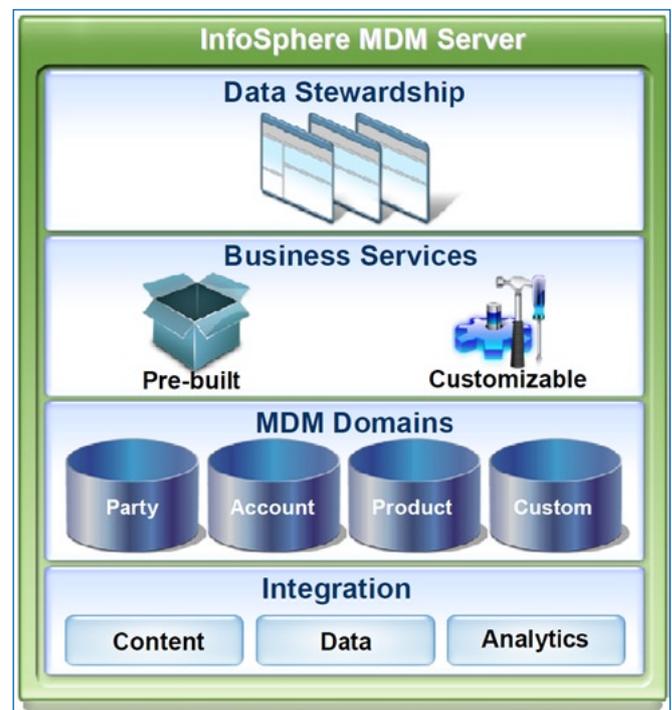


Abb. 1 MDM Server

Master Data Management (MDM) Systeme adressieren viele dieser Herausforderungen. Grundlage ist ein umfassendes normalisiertes Datenmodell. Die Datenmodelle leistungsfähiger MDM Systeme repräsentieren die jahrzehntelange Erfahrung in der Nutzung von Stammdaten. Hierbei unterscheidet man zwischen kollaborativem (Product Information Management - PIM) und operativem Master Data Management (Customer Data Integration - CDI). Darüber hinaus nutzen analytische MDM Systeme PIM- und CDI-Implementierungen, um weitergehende Kenntnisse aus diesen zu gewinnen und zur Optimierung des Geschäfts beizutragen. Datenbanken heutiger MDM Systeme implementieren vier Datendomänen: Party (natürliche und juristische Personen), Account (Verträge), Product (Produktspezifikationen) und Location (Adressen sowie Telefon, E-Mail etc.) sowie die Beziehungen innerhalb und zwischen diesen Domänen.

Der IBM InfoSphere Master Data Management Server 9.0 ist ein operationales MDM System, basierend auf einer Service-orientierten Architektur (SOA). Kernbestandteile sind ein umfassendes und erweiterbares Datenmodell sowie hunderte von Services unterschiedlicher Granularität. Die grafische Benutzeroberfläche (Data Stewardship User Interface) unterstützt die zentrale Ausführung der Data Governance Prozesse, bspw. die Erkennung und Bereinigung von Datendoubletten, die Administration der Zugriffsrechte (Rules of Visibility) sowie die Pflege der Schlüsselwerte.

Ein Batch-Framework ermöglicht die Verarbeitung von Massendaten sowie das erstmalige Befüllen des Systems (Initial Load). Zu Revisionszwecken können sämtliche Transaktionen protokolliert werden. Die Historienfunktion ermöglicht jederzeit die Sicht auf einen Datenbestand zu einem beliebigen Zeitpunkt. Der Event Manager benachrichtigt beim Eintreten definierter Ereignisse externe Systeme. Neben individuellen Validierungs- und Plausibilitätsregeln können komplexe Regeln auch auf Basis einer Rules Engine (iLog) definiert werden.

Ein (neues) MDM System mit adäquatem Datenmodell ist ein für das Geschäft zentrales System mit vielen Schnittstellen und stellt damit ein hohes Risiko dar. Um dieses Risiko zu minimieren, hat sich ein Vorgehen bewährt, welches die Einführung in drei Phasen unterteilt:

1. Die Etablierung des MDM ohne Primär-Verantwortung ggf. bei eingeschränktem Attribut-Set mit dem Ziel: Einführung und Stabilisierung des Systems für eine überschaubare Aufgabe ohne Risiko für das Gesamt-Unternehmen
2. Primärverantwortung für die verwalteten Daten, wobei die schreibenden Systeme zur Pflege der Daten (z. B. Frontends) angebunden werden und das Altsystem zur indirekten Anbindung der lesenden Systeme rückversorgt wird, mit dem Ziel: Stabilisierung des Betriebs sowie Verbesserung und Sicherung der Datenqualität
3. Die direkte Anbindung aller lesenden Systeme an das neue MDM und Rückbau des Altsystems

Diese Phasen können bei komplexen Unternehmensarchitekturen durchaus in mehrere Releases unterteilt werden. So hat sich speziell in der zweiten Phase ein zeitweiliger Mix von Daten mit Primär- sowie Sekundärverantwortung bewährt. Dabei ist jedoch strengstens zu beachten, dass diese Daten komplett voneinander isoliert sind, wie das z. B. üblicherweise bei verschiedenen Mandanten der Fall ist.

Begleitend zu den ersten beiden Phasen sollte eine Data Governance etabliert werden, um auch organisatorisch zur kontinuierlichen Sicherung der Datenqualität beizutragen und für die Zukunft zu vermeiden, dass z. B. wieder vorhandene Felder für andere Aufgaben missbraucht werden (siehe Abb. 2).

Die Verbesserung der Qualität von Stammdaten ist einer der Hauptgründe zur Modernisierung der Anwendungsland-

Eine Vielzahl sogenannter Code-Tabellen unterstützt die typisierte Speicherung von Attributwerten, beispielsweise eine vollständige Liste gültiger Währungs- oder Länderschlüssel, und die weitgehende Vermeidung von Freitextfeldern. Ihre Pflege obliegt dem Datenqualitätsmanager, der auch für die Aktualität und Korrektheit der möglichen Typwerte verantwortlich ist. Hierzu steht ihm eine grafische

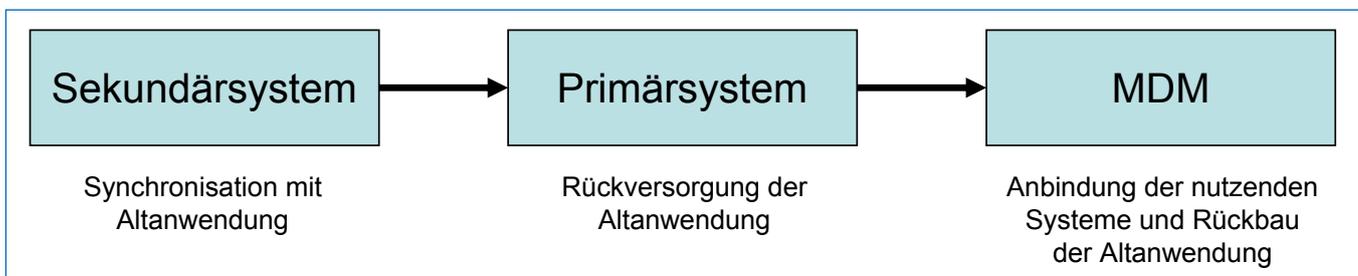


Abb. 2 Integrationsschritte

Benutzeroberfläche zur Verfügung, die ihm eine vollständige Unterstützung aller datenqualitätsbezogenen Aktivitäten bietet. Diese umfasst auch Funktionen, um die erfassten Datendoubletten zu erkennen und zu bearbeiten.

Wichtig ist, dass das Datenmodell offen und erweiterbar ist, damit man kundenspezifische Modellerweiterungen einfach aber kontrolliert vornehmen kann. Dies reduziert die Motivation, bestehende Felder für andere Daten zu „missbrauchen“. Ebenso wirken der Speicherung in semantisch fremden Feldern individuelle Validierungsregeln entgegen. Moderne MDM Systeme bieten eine Vielzahl von Services um Daten speichern, lesen und pflegen zu können (sog. CRUD Services: create, read, update, delete). Diese sind Voraussetzung, um einerseits weitere Systeme (wie z. B. im Zuge von Mergers & Acquisitions) effizient integrieren zu können. Andererseits bilden sie die Grundlage für die Einhaltung der Zugriffsrechte einschließlich Mandantenfähigkeit.

schaft eines Unternehmens. Die stufenweise Einführung eines Master Data Management Systems unterstützt die Einführung von Datenqualitätsprozessen und ermöglicht die sichere Ablösung der Altsysteme. ■

**Literatur/Quellen**

- [1] [www.ibm.com/software/de/iod/trustedinformation/](http://www.ibm.com/software/de/iod/trustedinformation/)
- [2] [www.gartner.com/DisplayDocument?doc\\_cd=167733](http://www.gartner.com/DisplayDocument?doc_cd=167733)

**Michael Ehrmantraut**

ist Senior IT Architect bei IBM SWG Services und ist technisch verantwortlich für die Einführung von MDM Systemen. (ehrman@de.ibm.com)

**Michael Sebald**

berät als Product Consultant bei IBM Global Business Services Unternehmen bei der Einführung von MDM Systemen. (michael.sebald@de.ibm.com)

# Mit Jazz und Rational Team Concert zum Dream Team werden

Werkzeuge in der Softwareentwicklung legen den Schwerpunkt bisher meist sehr stark auf einen bestimmten Teilbereich des Gesamtprozesses. Vernachlässigt wird dabei die Integration, die dann in einem aufwändigen Punkt-zu-Punkt-Verfahren erfolgen muss. Obwohl es Ansätze zur Vereinheitlichung auf Eclipse-Basis gibt, liefern Tools wie CVS (Concurrent Versions System), Apache ANT (Another Neat Tool), Bugzilla und Jira lediglich Integrationsmöglichkeiten über die Oberfläche. Vernetzte Informationen können dabei allerdings nicht verarbeitet werden. So kann CVS beispielsweise keine Angaben darüber machen, welche der Änderungen im Code für welchen von Bugzilla entdeckten Fehler gemacht wurden. Und Bugzilla gibt keine Auskunft darüber, welche Anforderungen mit den von Jira verwalteten Ressourcen geliefert werden. Jazz und Rational Team Concert von IBM stellen dagegen die Integration und die Zusammenarbeit im Entwicklungsteam in den Mittelpunkt und eröffnen so eine völlig neue Dimension der Softwareentwicklungstools.

## Jazz-Plattform integriert Entwicklungswerkzeuge

Jazz ist eine im Jahr 2004 gestartete Initiative von IBM, welche die Softwareentwicklung in Teams signifikant verbessern soll. Jazz besteht aus drei Elementen:

1. einer Architektur, um den Software-Lebenszyklus zu integrieren,
2. einem Portfolio von Produkten mit starker Teamorientierung und
3. einer Gemeinschaft von Menschen, die sich aktiv an der Verbesserung beteiligen.

Jazz erweitert die Eclipse-Technologie und sorgt für eine bessere Integration von Entwicklungswerkzeugen untereinander durch die so genannte Jazz Integration Architecture (JIA). Diese definiert eine Menge an Jazz Foundation Services, die von jedem Jazz-basierten Tool verwendet werden kann, und beschreibt, wie diese Services verwendet werden. Für die Umsetzung der flexibleren Integration kommt ein offener Standard, die Open Services for Lifecycle Collaboration, zum Zuge.

## Rational Team Concert orchestriert Zusammenarbeit der Entwickler

Eine zentrale Rolle spielt dabei Rational Team Concert, das als Werkzeug auf Jazz-Basis die Zusammenarbeit und Integration über einen weiten Entwicklungszyklus hinweg ermöglicht. Herzstück von RTC ist der Jazz Server. Er liefert die nötige Repository-Technologie und zentrale Dienste, wie beispielsweise die Verwaltung von Benutzern, Team, Projekten, Prozessen und der Kollaboration. Der Jazz Server kann auf zahlreichen unterschiedlichen Plattformen betrieben werden, darunter Linux, Windows, Unix, zLinux oder z/OS.

Auf dem Jazz Server setzen drei zentrale Komponenten von Rational Team Concert auf:

- Software Configuration Management (SCM)
- Anforderungen (Work Items)
- Build (automatisiertes Bauen von Anwendungen)

Kernfunktionalitäten des Software Configuration Management (SCM) sind die Organisation der Entwicklungsebenen in Form von Streams, die Untergruppierung in Komponenten, sowie eine Versionierung. Mehrere Konstrukte im SCM unterstützen die Agile Entwicklung, zum einen, dadurch dass mit optimistischen Locking parallele Entwicklung möglich ist, zum anderen weil ein anpassbarer Entwicklungsprozess wie z. B. der SCRUM Prozess aktiv berücksichtigt wird. Für die Entwickler ist das SCM deutlich komfortabler zu bedienen als vergleichbare Lösungen.

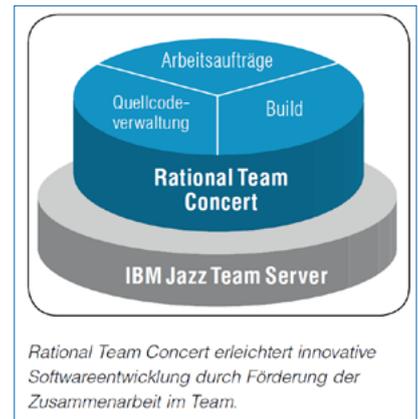


Abb. RTC Übersicht

Work Items sind als Arbeitsaufträge verschiedener Arten zu verstehen, zum Beispiel Defekte, Verbesserungen und Erweiterungswünsche. Über die Komponente können einzelne Aufträge den jeweiligen Entwicklern zugewiesen werden. Die Entwickler melden sich am Jazz Team Server an und bekommen Informationen über neue Aufträge, die ihnen zugewiesen wurden. Außerdem ist es möglich, über einen Query-Mechanismus Aufträge gezielt anzuzeigen. Wenn Änderungen im Code beim Einchecken in das SCM mit einem Auftrag verknüpft werden, ist im Nachhinein klar ersichtlich, welche Code-Änderungen durchgeführt wurden. Diese Verknüpfung kann im Prozess definiert werden und wird dann von RTC beim Check-in aktiv eingefordert.

Der Build schließlich lässt sich lokal oder auch auf einem entfernten System anstoßen. Wichtig ist hier die Nachvollziehbarkeit. Diese geht so weit, dass aus einem Build klar ersichtlich wird, welche Aufträge und Change Sets eingeflossen sind. Arbeitsaufträge kann man sehr komfortabel von der Eclipse-Oberfläche in das integrierte Instant Messaging System ziehen und an andere Entwickler im Team verschicken. RTC ermöglicht eine Interaktion im Kontext mit vernetzter Intelligenz. Jeder Baustein von RTC kann daher auf gemeinsam verwaltete Informationen zugreifen.

## Das Ganze ist mehr als die Summe seiner Teile

Die Entwicklungsumgebung Rational Team Concert liefert Entwicklungsteams eine stark vernetzte Funktionalität. Das Software Configuration Management, die Auftragsverwaltung und der Build, aber auch die Planung von Iterationen, die Projekttransparenz und die Unterstützung vorgegebener Entwicklungsprozesse tragen dazu bei, dass Teams effizient zusammenarbeiten können und eine Gesamtlösung entwickeln, die optimal zusammenspielt. Dies verbessert und beschleunigt den Entwicklungsprozess erheblich. ■

## Literatur/Quellen

<http://jazz.net>

### Benjamin Storz

ist akkreditierter IT Specialist bei IBM und beschäftigt sich mit dem Thema Anwendungsentwicklung und -modernisierung für Mainframe Systeme (benjamin.storz@de.ibm.com)

# Application Modernization by Migration

Der Begriff „Migration“ wird im Sprachgebrauch, auch im IT-Umfeld, vielfältig verwendet, so dass zur Einleitung zunächst eine grobe Definition erfolgen soll. Unter Migration verstehen wir im Folgenden die Portierung von Software Applikationen von einer Betriebssystem-, Sprach- oder Datenbank-Umgebung in eine neue Zielumgebung, ohne dass signifikante Änderungen an Funktionen oder fachlichem Design vorgenommen werden. In Erweiterung dazu lassen sich auch Aufgaben unter dem Oberbegriff Migration einordnen, bei denen zwar die Plattform nicht gewechselt wird, aber Transformationen rein formal und ohne Kenntnis der Funktionalität beschrieben werden können, wie z. B. die Erweiterung von Schlüssel- und Datenfeldformaten oder die Konvertierung von Codepages innerhalb einer Applikation.

Neben extern getriebenen Motiven, wie z. B. regulatorischen Vorgaben oder Mergers & Acquisitions, treten überwiegend Aspekte der Anwendungsmodernisierung als Auslöser für Migrationsvorhaben auf. Die wichtigsten Ziele dabei sind:

- Durchsetzung der Firmen-IT-Strategie: Vereinheitlichung der Plattformen
- Senkung der Betriebs- und Software-Wartungskosten
- Unabhängigkeit von auslaufendem, damit risikobehaftetem und schwer zu ersetzendem Know-how
- Schaffung einer technologischen Basis für weitere Schritte der Anwendungsmodernisierung

Die konkreten Migrationsprojekte können sowohl von ihren jeweiligen Rahmenbedingungen als auch von ihrem technischen Inhalt her sehr vielfältig sein. Folgende Grafik gibt einen kurzen Überblick und zeigt einige Beispiele auf:

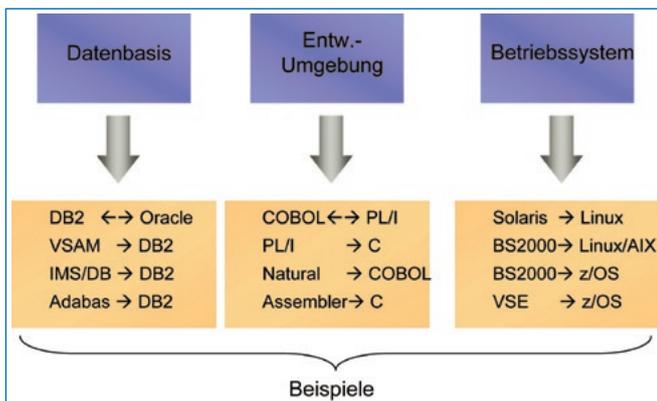


Abb. 1 Typen und Inhalte von Migrationsprojekten

Unabhängig von der Verschiedenartigkeit der Migrationaufgaben, welche übrigens häufig auch in Kombinationen auftreten, werden erfolgreiche Migrationsprojekte nach gleichen Methoden und Vorgehensmodellen umgesetzt. An erster Stelle ist hier das bewährte Phasenmodell zu nennen, welches nachfolgend komprimiert dargestellt ist (siehe Abb. 2).

Von besonderer Bedeutung ist die Vorstudie, die eine gründliche Analyse der Applikation(en) und der Rahmenbedingungen, die Konzeption der Lösung und darauf aufbauend die Planung des Projektes zum Inhalt hat. Eine mangelhaft durchgeführte oder gar fehlende Voranalyse ist die häufigste Ursache für scheiternde oder zu hoch kalkulierte Migrationsprojekte.

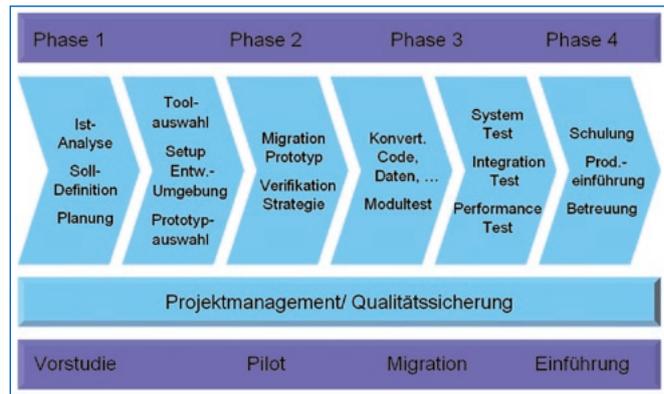


Abb. 2 Phasenmodell

Da man es in Migrationsvorhaben typischerweise mit einer großen Anzahl von Objekten und sich wiederholenden, regelbasierten Aktivitäten zu tun hat, ist der Einsatz von Software-Tools in fast allen Fällen obligatorisch. Abhängig von den konkreten Aufgaben und den Ergebnissen der Vorstudie werden in der Praxis bewährte Tools eingesetzt, die entweder aus dem Produktportfolio der IBM oder aber von, auf bestimmte Teilaufgaben spezialisierten, Tool-Partnern stammen. Manuelle Umstellungsarbeiten kommen genau dann zum Tragen, wenn die Migrationsschritte sehr komplex sind und der Wiederholungsgrad gering ist. Entscheidend für die Auswahl des richtigen Tools, ggf. in Kombination mit manuellen Aktivitäten, ist ein Kriterienkatalog.

Neben dem Vorgehensmodell und der Tool-Auswahl ist die sorgfältige Abstimmung der Rollenverteilung zwischen dem Entwickler und Betreiber der zu migrierenden Applikation(en) und dem Migrationsdienstleister ein wichtiger Erfolgsfaktor. Erfahrungsgemäß stecken 40-50% des Gesamtaufwandes für ein Migrationsvorhaben im Test und dessen Vorbereitung. Da der Eigentümer der Applikation über das für die wichtigsten Teststufen benötigte Fachwissen verfügt, übernimmt er typischerweise diese Aufgabe, und das mit hoher Effizienz. Gleichwohl sind beim Thema Test auch andere Szenarien denkbar und erfolgreich praktiziert worden.

Abschließend soll an einem praktischen Fallbeispiel der Rationalisierungseffekt von Migrationsprojekten gezeigt werden.

Die Aufgabe bestand in der Migration einer Versicherungsanwendung (Plattform z/OS) von Cobol, Assembler, IMS/DB nach PL/I und DB2. Das Motiv für die Umstellung war eine Vereinheitlichung der IT-Plattform und damit verbunden eine Kostenersparnis. Wegen des vergleichsweise geringen Umfanges der Applikation wurde in der Vorstudie das Konzept einer manuellen Umstellung unter Nutzung selbstentwickelter unterstützender Assets gewählt. Weiterhin wurde die Migration mit einer rein technischen Restrukturierung und Modularisierung kombiniert. Die zeit- und aufwandsgerichteten Ergebnisse mündeten in einer Reduktion der Programmanzahl auf ca. 20%, der Anzahl der Copystreifen auf ca. 15% und dem Ersatz von 24 IMS-Segmenten durch 5 DB2-Tabellen. Die Aufwendungen in der Betreuung und Software-Pflege der Applikation konnten dauerhaft um 80% gesenkt werden, die Benutzerfreundlichkeit und die Stabilität der Anwendung wurden deutlich verbessert. ■

### Karl-Heinz Homilius

Leiter Geschäftseinheit Migration Services  
(kh.homilius@de.ibm.com)

### Sebastian Junge

IT Architect  
(sebastian.junge@de.ibm.com)

# Dynamische Software Infrastrukturen auf Basis agiler IBM Mainframe Lösungen

*Für den Erfolg in einer Welt, die zunehmend digitalisiert, intelligent und vernetzt ist, wird eine intelligente Art von Infrastruktur benötigt. Eine Infrastruktur, mit der sie den Service optimieren, die Kosten senken und die Risiken unter Kontrolle halten, während sie gleichzeitig das Wachstum ihres Unternehmens beschleunigen.*

Eine dynamische Infrastruktur stimmt die Ressourcen des Geschäfts- und des IT-Bereichs aufeinander ab und unterstützt die Gesamtziele des Unternehmens. Dabei wird ein intelligenter, neuer und optimierter Ansatz verfolgt. Folgende Herausforderungen sind zu bewältigen:

## Anforderungen an den Service waren nie höher

Interne Service Level Agreements sind nicht mehr ausreichend. Insbesondere digitale Kunden erwarten innovative, 24 x 7, High-Quality-Dienstleistungen.

33 % der Konsumenten mit Breitbandverbindungen warten nicht länger als vier Sekunden auf das Öffnen einer Website.<sup>1</sup> 69 Mrd. \$ Umsatz im Handel gehen im Jahr verloren, weil nicht die richtigen Produkte im Lager sind, die dem Kundenbedarf entsprechen.<sup>2</sup>

## Verbesserung des Return on Investment (ROI) auf die Infrastruktur

Kosteneffizienz ist heutzutage ein Top-Thema für den Großteil der C-Level-Verantwortlichen. Es geht aber nicht nur um Kosteneinsparung, sondern auch darum, wie das verfügbare Budget besser eingesetzt wird. In verteilten Umgebungen sind zum Beispiel 85 % der Computer-Kapazitäten nicht ausgelastet.<sup>3</sup> Im Durchschnitt werden von 100 Energieeinheiten im Rechenzentrum nur drei Energieeinheiten für das eigentliche Computing genutzt.<sup>4</sup> 70 % des durchschnittlichen IT-Budgets wird für den Betrieb der bestehenden IT-Infrastruktur eingesetzt und steht nicht für neue Projekte, Prozesse und Innovationen zur Verfügung.

## Risiko ist Realität im täglichen Business

Organisationen fokussieren sich auf Risikoreduzierung und Risikomanagement bei gleichzeitiger Unterstützung der Geschäftsziele. Dabei müssen gesetzliche, regulatorische, industriespezifische und organisatorische Vorschriften eingehalten

werden. 71 % der befragten CIOs identifizierten Risk-Management als wichtigen Teil ihrer Wettbewerbsstrategie. 33 % der Konsumenten beenden die Geschäftsbeziehung mit dem verantwortlichen Unternehmen nach Bekanntwerden von Sicherheitslücken.<sup>5</sup>



Abb. 1 IBM System z10 Mainframe

Mit einer dynamischen Infrastruktur lassen sich der Service verbessern, die Kosten reduzieren und das Risiko managen. Die IBM Strategie hilft, eine gemeinsam genutzte, integrierte und hochverfügbare Infrastruktur zu liefern, um den Herausforderungen und Geschäftschancen zu begegnen.

Was macht den IBM Ansatz für eine dynamische Infrastruktur so einzigartig?

- Integration von Transparenz, Kontrolle und Automatisierung für alle Geschäfts- und IT-Assets
- Effiziente, umweltschonende Infrastrukturen mit reduziertem Energieverbrauch, um wachsende Aufgaben mit weniger Mitteln zu bewältigen
- Bewältigung der informationsbezogenen Herausforderungen mit einer Datenstruktur für erhöhte Service-Level-, Compliance-Anforderungen und Reduzierung der Gesamtkosten
- Kontrolle und Begrenzung von Risiken
- Flexible Bereitstellungsoptionen von virtualisierten Ressourcen und Cloud Computing Umgebungen

Das Management von (digitalen) Geschäftsprozessen und (physischen) Infrastrukturkomponenten und die Erbringung neuer Services bezeichnet die daraus resultierende Aufgabenteilung als ‚Service-Management‘. In diesem Umfeld ist das Business Process Management ein Fokusthema. Im Januar 2009 akquirierte IBM nach 12-jähriger Partnerschaft den Softwarehersteller ILOG. Das bislang eigenständige Unternehmen betreute als einer der führenden Anbieter im Bereich Businesslösungen über 2500 Kunden wie beispielsweise Commerzbank,

Airbus und Siemens. Mit der Übernahme von ILOG erweitert IBM das WebSphere Produktportfolio um Supply-Chain-Management-Systeme, Visualisierungs-Software, Optimization-Decision-Management-Systeme wie auch Business-Rules-Management-Systeme.

Der Vorteil von WebSphere ILOG liegt darin, dass diese Software einem Unternehmen hilft, besser Entscheidungen zu treffen und daraus folgende Änderungen leichter durchzuführen, da sie die direkte Kontrolle über geschäftskritische Anwendungen und Prozesse erlaubt. Gerade im Bereich des Business Process Management ergänzt ILOG die Lösungen der IBM, da ILOG es ermöglicht, die Regellogik unabhängig vom Prozess zentral zu verwalten. Die Regeln können im Business-Rules-Management-System (kurz BRMS) über eine einfach zu bedienende und graphische Oberfläche erstellt und verwaltet werden.

Es werden auf dem IBM Mainframe zwei BRMS-Versionen unterschieden. Mit „JRules“ können Java Applikationen ihre extern verwalteten Regeln über eine durch das BRMS verwaltete Regeldatenbank eigenständig abgleichen. Die Änderung einer solchen Regel ist im BRMS ohne große technische Kenntnisse möglich und kann ohne neues Deployment im WebSphere Application Server erfolgen. Für bestehende Mainframe-Anwendungen in Cobol bietet das BRMS mit „Rules-for-Cobol“ ebenfalls eine entsprechende Modernisierungslösung zur Generierung von Regelsätzen.

Über alle Branchen hinweg gibt es Regeln, die sich aufgrund neuer Gesetzaufgaben oder interner Richtlinien ständig verändern. Hier hat IBM nun mit dem BRMS eine Lösung, die nicht nur die Reaktionszeit auf mögliche Änderungen ver-

kürzt, sondern durch einen schnellen Return on Investment zusätzlich Einsparungen bietet.

Neben dem Business-Rules-Management-System bietet WebSphere ILOG auch Optimierungssoftware an. Mit dieser Software können Unternehmen in der Fertigungs- und Transportbranche ihre Planungen optimieren und somit effizienter und rentabler handeln. Neben diesen Branchen hat auch die Bundesliga diese Optimierungssoftware für sich entdeckt: Um die Spielpläne unter Einbezug mathematischer Algorithmen zu erstellen, wird auch hier ILOG seit 2005/2006 erfolgreich eingesetzt.

Der IBM WebSphere Process Server ist eine weitere moderne Plattform für Geschäftsprozessmanagement, die den stetigen Wandel in der dynamischen, bedarfsorientierten Welt unterstützt. Hiermit werden die kritischen Geschäftsprozesse sicher, konsistent und unter Wahrung der Transaktionsintegrität verwaltet und ausgeführt. Diese SOA-Plattform kann die verschiedenen Anwendungen, Services und Geschäftsressourcen so koordinieren, dass sie optimierte und effektive Prozesse zur Erreichung der Geschäftsziele abbilden. Die benutzerfreundliche Entwicklungsumgebung mit dem WebSphere Integration Developer ist perfekt abgestimmt. Die Integration in eine SOA gelingt, indem vorhandene IT-Ressourcen als wiederverwendbare und effiziente Service-Komponenten dargestellt werden. Bei diesen Komponenten kann es sich um Prozesse, Mediationen, Services, Benutzer-Tasks, Geschäftsregeln, Adapter oder Code handeln. Prozess- und Integrationslösungen können mithilfe einer Drag-and-drop-Oberfläche visuell definiert werden. Dabei können sie höchst flexibel festlegen, welche Services dy-

## Advertorial

# Nutzen Sie Ihre Einsparpotenziale bei der Entwicklung von Großrechner-Anwendungen

Mit der Einführung der Internettechnologie, der Programmiersprache Java und der darauf aufsetzenden Konzepte kamen in den letzten zehn Jahren auch neue, moderne Entwicklungswerkzeuge in den Unternehmen zum Einsatz. Für die Wartung und Weiterentwicklung der Großrechner-Anwendungen wurde hingegen wenig am existierenden Entwicklungsumfeld geändert. Wohl hofften die IT-Verantwortlichen in den 90er Jahren darauf, dass die bestehenden Anwendungen langfristig durch Neuentwicklungen vollständig ersetzt würden. Somit wurden Modernisierungsvorhaben in diesem Umfeld immer wieder zurückgestellt.

Inzwischen ist jedoch ein deutliches Umdenken zu registrieren. Da für kaum ein Unternehmen die Neuentwicklung aller Großrechner-Anwendungen eine ernsthafte Option darstellt, besteht nun ein nachhaltiges Interesse darin, den Entwicklungsprozess und damit auch die Entwicklungswerkzeuge zu modernisieren. Eine Vielzahl von Gründen macht dies auch dringend erforderlich. Hier nur die wichtigsten Argumente:

- Die modernen Werkzeuge zur Entwicklung von Großrechner-Anwendungen versprechen eine erhebliche **Produktivitätssteigerung**.
- Die **Einarbeitung junger Mitarbeiter** in die veralteten Verfahren und Werkzeuge erweist sich als sehr aufwändig, kostspielig und demotivierend.
- Die unterschiedlichen Entwicklungsumgebungen führen zu erheblichen **Kommunikationsproblemen** innerhalb der IT.

Speziell mit dem „**Rational Developer for System z (RDz)**“ stellt IBM die moderne Schlüsselanwendung zur Entwicklung von Großrechner-Anwendungen zur Verfügung, die auf dem Open Source Standard Eclipse basiert.

Die offene Architektur von RDz erlaubt einen sukzessiven Umstieg in eine deutlich produktivere Entwicklungsumgebung. Einige Unternehmen haben diesen Prozess bereits abgeschlossen, die meisten sind mitten dabei.

Die **SoforTe GmbH** als IBM Advanced Business Partner und Spezialist zur Modernisierung des Anwendungsentwicklungsprozesses begleitet Sie dabei in allen Projektphasen, von der ROI-Betrachtung, Konzeption und Analyse bis hin zur Entwicklung funktionaler Erweiterungen, der technischen Integration unternehmensspezifischer Werkzeuge sowie der Produktionseinführung und Schulung der Anwender.



Ansprechpartner:  
**SoforTe GmbH**  
**Udo Partsch**  
 Donaustraße 16,  
 63452 Hanau  
 Tel.: 06181 1894 770  
 Email: service@sofor-te.de

namisch von IBM WebSphere Service Registry und Repository aufgerufen werden sollen, abhängig von den äußeren Faktoren, die sich zu dem Zeitpunkt auf die Prozessinstanz auswirken. Zusammen mit dem IBM WebSphere Business Monitor, mit dem Sie die Leistungswerte der Geschäftsprozesse dokumentieren können, bildet WebSphere Process Server eine umfassende und moderne Infrastrukturkomponente für den gesamten Lebenszyklus von Geschäftsprozessen.

### Warum es auf die Plattform ankommt

Moderne dynamische Infrastrukturen mit IBM WebSphere Process Server und ILOG werten die unbestrittenen Qualitäten des IBM Mainframes zusätzlich auf und profitieren von den Vorzügen moderner System z Server, die kosteneffizient, zuverlässig, umweltschonend und verwaltungsfreundlich sind. Die Lösungen bieten ein äußerst hohes Niveau an Verfügbarkeit und Nähe für Anwendungen und Daten. Das stellt nicht nur einen reibungslosen Ablauf der darauf betriebenen Geschäftsanwendungen sicher, sondern sorgt auch für überragende Sicherheit und Datenschutz. Angesichts der hohen Kosten von Unterbrechungen, ob durch Systemausfälle oder durch Sicherheitsverstöße, kommt dem reibungslosen Betrieb eine entscheidende Bedeutung zu. Cloud Computing bietet hierfür eine weitere Dimension, indem Rechenleistung und Programme im Internet bzw. Intranet genutzt werden können.

Eine neue Generation informierter Kunden, Mitarbeiter, Interessengruppen und Partner legt immer mehr Wert auf soziale

Verantwortung der Unternehmen. Mit dem Modell der modernen dynamischen Infrastruktur wird mehr Service geliefert, die IT umweltfreundlicher gestaltet und es können erhebliche Energieeinsparungen erzielt werden. ■

### Fußnoten:

- <sup>1</sup> Jupiter Media, Akamai 2008
- <sup>2</sup> Harvard Business School / Motorola Studie
- <sup>3</sup> Journal of Object Technology 2009
- <sup>4</sup> U.S. Department of Energy, Mai 2007
- <sup>5</sup> Ponemon Institute, 2007

### Olaf Hendrik Kappel

Sales Leader WebSphere Software auf  
IBM System z Nordosteuropa,  
IBM Deutschland GmbH  
(kappel@de.ibm.com)



Abb. 2 Wachstum ermöglichen, Risiken managen, Virtualisierung, Reduktion von Energiekosten mit IBM System 10

# Migrationen ohne Risiko für das Business

*Migrationen sind seit jeher etwas, das man als IT-Verantwortlicher mit einem gewissen Unbehagen betrachtet. Praktische Erfahrungen aus vielen Projekten zeigen, dass keine Migration der anderen gleicht. Es gibt keine Migrationen von der Stange. Es gibt aber sehr wohl etablierte Prozesse, Werkzeuge und Experten, die dafür sorgen, dass das Rad nicht jedes Mal wieder neu erfunden werden muss. Eine Vielzahl von Migrationsprojekten beweist, dass sich diese sehr erfolgreich und zur Zufriedenheit der IT-Verantwortlichen und auch deren Kunden durchführen lassen.*

## Gründe für eine Migration

Ein System, das zum Installationszeitpunkt State of the Art war, kann aus einer Vielzahl von Gründen veralten. Sei es, weil sich gesetzliche Bestimmungen geändert haben oder weil der Technologiewandel eine neue Situation vor allem auf der Kostenseite schafft. Man kann hier exemplarisch nur am Rande die Themen Energieeffizienz, Virtualisierung, Wartungskosten, Lizenzgebühren oder End-of-Life von Architekturen erwähnen. Meist sind Systeme an ihre Auslastungsgrenzen gestoßen, was die Behebung von Engpässen in CPU-, Platten- oder Hauptspeicherausrüstung erfordert.

## Der Blick auf aktuelle Anforderungen

IT Abteilungen unterliegen einem permanent steigenden Kostendruck. Behält man die zugrunde liegende Architektur bei, so kann eine Erneuerung der Infrastruktur bzw. der verwendeten Server zu Kosteneinsparungen führen, wenn durch den Einsatz leistungsfähigerer Systeme die Gesamtzahl der Server reduziert wird. Leistungsfähigere Systeme lassen sich in der Regel auch stärker auslasten und sorgen unter Verwendung von Virtualisierung für eine Konsolidierung der IT Umgebung.

Ständig muss hinterfragt und geprüft werden, ob es nicht architekturelle und technologische Verbesserungen gibt. Ein Beispiel im Software-Bereich ist die Ablösung einer eigenentwickelten Software durch ein Standardprodukt. Dies führt in der Regel zu erheblichen Einsparungen. Auch eine unklare Roadmap von Anbietern im Hard- oder Software-Bereich spielt hier eine essentielle Rolle für eine Risikoabschätzung. Die Erfahrung zeigt: Sehr viele Migrationen sind durch ein Architektur-Ende im Hardwarebereich getrieben. Das Fortführen von nicht mehr weiterentwickelten Insellösungen lässt sich oft nur durch extrem hohe Wartungskosten teuer erkaufen.

Die Migration berücksichtigt dabei die gesamte Anwendungs-umgebung: Server Plattform und Betriebssystem, Datenbanken, installierte Fremd- und eigenentwickelte Software sowie zugehörige Skripte. Generell gilt: Jede Migration birgt Risiken. Ziel aller Konzepte ist es, diese Risiken abzuschätzen, zu bewerten und zu minimieren. Dazu bieten sich methodische Verfahren an, die standardisierte Elemente wie z. B. Fragebögen, Analyseskripte, Checklisten umfassen. Am Anfang stehen Assessments, in denen alle notwendigen Daten und die Architektur des Quellsystems erfasst werden. Danach können passende Zielarchitektur, Software-Versionen und Migrationstools für das System festgelegt werden. Nach dem Aufsetzen des Projektteams (zertifizierte Migrationsexperten, Infrastruktur-spezialisten, Projektleiter, Software-Entwickler) kommt es in der Regel zu Proof of Concepts, bei dem die ausgewählte Mi-

grationsmethode und die Hardware auf Funktionalität, Performance und Verfügbarkeit sowie die Applikation auf Funktionalität und Konsistenz geprüft werden. Fallback-Lösungen werden evaluiert und in den Projektplan eingearbeitet.

Je nach Phase existieren auch für den Kunden unterschiedliche Mitwirkungspflichten, die für ein erfolgreiches Migrationsprojekt unerlässlich sind. Eine Migration lässt sich schwerlich als reines Service-Projekt extern vergeben, sondern ist immer auf eine enge Partnerschaft zwischen Auftraggeber und durchführendem Projekt-Team angewiesen.

## Migrationsberater, Kernteam und Erweitertes Team

In der Praxis bewährt hat sich der Einsatz von Migrationsberatern. Diese setzen sich aus IT-Architekten und erfahrenen Systemberatern aller Bereiche (Mainframe, Midrange, UNIX, x86 und Speichersysteme) zusammen, die auch über gute Kenntnisse der Architekturen verschiedener Anbieter verfügen. Diese Migrationsberater werden nach Skill Level und regionalen Gesichtspunkten den einzelnen Projekten zugeordnet. Ihr Einsatz ist am Anfang (beginnend mit der Assessmentphase) eines Migrationsprojektes hoch, im Fortgang des Projektes übernehmen dann die Projektleiter und führen die Migration bis zum Ende weiter. Der Migrationsberater begleitet aber bis zum Ende, und kann so alle gewonnenen Erfahrungen für die nächsten Projekte verwenden.

Als flankierende Maßnahme steht die permanente Fortbildung der Experten zu Migrationstools und -methoden (durch Konferenzen, Seminare, regelmäßigen Erfahrungsaustausch etc.) auf der Tagesordnung. Dies ermöglicht die richtige Einschätzung der geeigneten Methoden und Tools zu einer sehr frühen Phase. Zu ausgefallenen Themengebieten werden definierte Erweiterte Teammitglieder hinzugezogen, die Spezialgebiete abdecken.

Wenn sich ein gestiegener Bedarf für ein bestimmtes Migrations-Tool herauskristallisiert, werden vom Team auch Schulungen und Wissenstransfer für die Service-Bereiche aufgesetzt. Das Bild runden spezielle Handbücher zu Migrationsthemen ab, die durch gezielte Umsteigerkurse noch ergänzt werden können. All dies dient dem Ziel, Migrationen mit möglichst wenig Risiko und mit möglichst wenig oder ohne Betriebsunterbrechungen und Wochenendeinsätze der Administratoren und Applikationsabteilungen durchzuführen.

## Fallbeispiel 1: Migration eines Datenbanksystems

In einem deutschen Maschinenbau-Unternehmen wurden alle Konstruktions- und Fertigungsdaten in einer Oracle-Datenbank auf einem Sun Server abgelegt. Über die Jahre wuchs das gesamte Datenvolumen. Da diese Daten revisions-sicher und jederzeit zugreifbar abgelegt sein müssen, war ein zusätzlicher Server und Datenbank keine Lösung. Die Datenbank musste auf ein leistungsfähigeres skalierbares System umgezogen und auf eine neue Version gebracht werden.

Der Kunde versuchte, die 1,2 TB große Datenbank mit Standard Oracle Bordmitteln zu migrieren. Sein Wartungsfenster betrug 36 Stunden. Nach 40 Stunden brach er den Datenexport ab und wandte sich an IBM. Es wurde ein Team gebildet, das aus einem bereits beim Kunden engagierten Business Partner, einem erfahrenen Projektleiter und einem Migrations-experten bestand. Diese analysierten die vorhandene Installation, schlugen ein skalierbares IBM POWER System als Ziel-

plattform vor und erstellten einen detaillierten Projektplan. Bei der Analyse stellte sich heraus, dass die Daten fast gänzlich in einer einzigen Tabelle abgelegt waren.

Zur Durchführung des Projekts wurde das Datenbank Migrationstool XenoBridge der IBM Migration Factory eingesetzt und die Tabelle in eine größere Anzahl von Tabellen aufgespalten. Das Tool ist in der Lage, Tabelleninhalte parallel vom Quellsystem auf das Zielsystem umzuziehen, so dass die Migrationszeit im Wesentlichen nur von der größten Tabelle in der Datenbank abhängig ist.

Aufbau und Installation der Zielplattform erfolgten durch den Business Partner, der für die gesamte Infrastruktur des Kunden verantwortlich war. Danach wurde die Verbindung zwischen Quell- und Zielsystem getestet und optimiert, um sicherzustellen, dass die notwendige Netzwerkbandbreite gegeben ist. In einem ersten Test wurde eine Kopie der Datenbank umgezogen, um sicherzustellen, dass die Migration innerhalb des Wartungsfensters des Kunden durchgeführt werden konnte. So vorbereitet und mit der Fallback-Lösung, das bisherige System weiternutzen zu können, wurde ein Risiko für den laufenden Betrieb ausgeschlossen.

Der finale Umzug der Produktionsdatenbank wurde wie geplant an einem Freitag Abend gestartet und war nach 5,5 Stunden abgeschlossen, vom Kunden getestet und zur Produktion freigegeben worden. Die zugrunde liegende Datenbank wurde dabei von Version 9i auf 10g umgezogen. Dieser Upgrade erhöhte den Aufwand nicht. Die gesamte Projektdauer betrug 6 Wochen.

## Fallbeispiel 2: Migration einer komplexen Anwendungsumgebung

Zum Jahresende 2008 entschied sich die Unternehmensgruppe Tengelmann zur Migration der meisten HP Server auf IBM POWER Systems. Bei den abgelösten Servern handelt es sich um Datenbankserver für eine der wichtigsten Anwendungen: Die zentrale Warenwirtschaft und Logistik für Kaiser's Tengelmann. Neben dieser Produktionsumgebung wurden auch weitere Server für Disaster Recovery, Test, Entwicklung und Schulung ersetzt. Besonders bemerkenswert hierbei: Die abgelösten HP Server waren zum Teil erst ein bis zwei Jahre alt, die Abschreibungszeit von fünf Jahren also noch lange nicht erreicht.

Das Warenwirtschafts- und Logistiksystem war auf einem Midrange HP Itanium-basierenden Oracle RAC Cluster unter Linux installiert. Durch die Migration auf zwei AIX basierte IBM POWER Systeme 595 und ein IBM POWER System 570 unter Verwendung von Tivoli Systems Automation (TSA) ohne Oracle RAC wurde die für Kaiser's Tengelmann notwendige Skalierbarkeit und Ausfallsicherheit erreicht.

16 Oracle Datenbanken mit einem Volumen zwischen 100GB und 2.5TB wurden durch das IBM MigrateNow! Team innerhalb von 4 Wochen mit Einsatz des bewährten Tools XenoBridge umgezogen. Die Umstellung verlief problemlos und zur vollsten Zufriedenheit des Kunden.

### Nikolai von Dehn

Teamleader MigrateNow!  
(nvdehn@de.ibm.com)

### Dr. Oliver Rettig

Technical Solution Manager Migration Factory  
(rettig@de.ibm.com)

### Kontaktadresse

#### für die Beiträge S. 5 bis 34:

IBM Deutschland GmbH  
IBM-Allee 1  
71139 Ehningen  
URL: [www.ibm.com/software/de/rational/solutions/em](http://www.ibm.com/software/de/rational/solutions/em)

### Herausgeber/Verlag

SIGS DATACOM GmbH,  
Lindlastr. 2c, D-53842 Troisdorf  
Tel.: +49 (0) 22 41/23 41-1 00,  
Fax: +49 (0) 22 41/23 41-1 99  
URL: [www.sigs-datacom.de](http://www.sigs-datacom.de)  
E-Mail: [info@sigs-datacom.de](mailto:info@sigs-datacom.de)

### Verlagsleitung

Günter Fuhrmeister

### Vertriebs- und Marketingleitung

Emanuel Rosenauer  
Tel.: +49 (0) 22 41/23 41-5 60,  
E-Mail: [Emanuel.Rosenauer@sigs-datacom.de](mailto:Emanuel.Rosenauer@sigs-datacom.de)

### Redaktions- und Herstellungsleitung

Susanne Herl  
Tel.: +49 (0) 22 41/23 41-5 50,  
E-Mail: [Susanne.Herl@sigs-datacom.de](mailto:Susanne.Herl@sigs-datacom.de)

### Schlussredaktion

Heike Weidner

### Druck

Erdl Druck Medienhaus GmbH,  
Gabelsbergerstr. 4-6, 83308 Trostberg

### Abonnenten-Service

IPS Datenservice GmbH, Postfach 13 31,  
D-53335 Meckenheim,  
Tel.: +49 (0) 22 25/70 85-3 74  
Fax: +49 (0) 22 25/70 85-3 76,  
Patrick König, Markus Preis  
E-Mail: [aboservice@sigs-datacom.de](mailto:aboservice@sigs-datacom.de)

**Erscheinungsweise OBJEKTSpektrum** zweimonatlich

### Bezugspreis Sonderheft

Deutschland € 3,80, Europa € 4,50

### Bezugspreise OBJEKTSpektrum

Einzelverkaufspreis: D: € 8,50, A: € 9,50, CH: sfr 15,60  
Jahresabonnement Deutschland: € 48,00 inkl. Versand  
Jahresabonnement Europa: € 55,20 inkl. Versand  
Studentenabo: € 43,20 inkl. Versandkosten

### Lieferung an Handel

Verlagsunion KG, Postfach 57 07,  
65047 Wiesbaden, Tel.: +49 (0) 61 23/6 20-0

© 2010 SIGS DATACOM GmbH

# Software-Evolution in Forschung und Praxis: Fachgruppe Software- Reengineering

*In Deutschland veranstaltet die Gesellschaft für Informatik (GI) seit 1999 jährlich den Workshop Software-Reengineering (WSR). Der Low-Cost-Workshop wurde von Jürgen Ebert (Universität Koblenz) und Franz Lehner (Universität Passau) als regelmäßiger Treffpunkt für Praktiker und Wissenschaftler aus Software-Technik und Wirtschaftsinformatik geschaffen, die sich mit dem Umgang und der Evolution von vorhandenen Software-Systemen befassen. Der WSR hat sich inzwischen als zentrale deutschsprachige Reengineering-Konferenz etabliert und findet vom 3. – 5. Mai 2010 zum 12. Mal in Bad Honnef statt. Die wachsende Beteiligung am WSR führte 2004 zur Gründung der GI-Fachgruppe Software-Reengineering (<http://www.uni-koblenz.de/sre/>).*

Die Themen des WSR behandeln die wissenschaftliche Entwicklung und Erprobung von Methoden und Techniken der Software-Wartung und des Reengineering sowie deren Anwendung in konkreten Projekten zur Software-Evolution. Diese Methoden und Techniken umfassen statische und dynamische Programm-Analysen, Clone-Erkennung, Architektur-Erkennung, Reverse-Engineering und Programmverstehen, Software-Visualisierung, Software-Tests, Produktlinien-Identifikation, Programm-Restrukturierung und Refactoring, Qualitätssicherung und Software-Migration. Erfahrungsberichte aus der Praxis behandelten die Entwicklung und Anwendung industrieller Wartungswerkzeuge, große Migrationsprojekte und projektbegleitende Qualitätserhebung und -sicherung. Regelmäßig ergänzt wurden Tool-Präsentationen aus Praxis und Universität.

Eine wesentliche Leistung des WSR ist die intensive Zusammenarbeit von Wissenschaftlern und Praktikern. IT-Architekten und Entwickler aus der Praxis berichten über ihre Reengineering-Vorhaben und ihre Projekterfahrungen. Forscher nehmen Praxisanforderungen auf, um sie in ihren Arbeiten zu berücksichtigen und um praktische Probleme mit ihren Prototypen zu lösen. Ein Beispiel ist die Analyse realer, mehrere Sprachen umfassender Anwendungssysteme großer Versicherungsunternehmen. Umgekehrt werden aktuelle Forschungsergebnisse, z. B. verschiedene Ansätze zum Programmverstehen oder zur Clone-Erkennung und -Eliminierung, in direkten Anwendungsbezug gebracht und von Praktikern aufgenommen. Aus den gemeinsamen Treffen sind auch diverse Kooperationen zwischen Forschungseinrichtungen, Software-Häusern und -Anwendern hervorgegangen. So wird aktuell in dem BMBF-geförderten SOAMIG-Projekt ([www.soamig.de](http://www.soamig.de)) von Praktikern und Wissenschaftlern der Fachgruppe ein transforma-

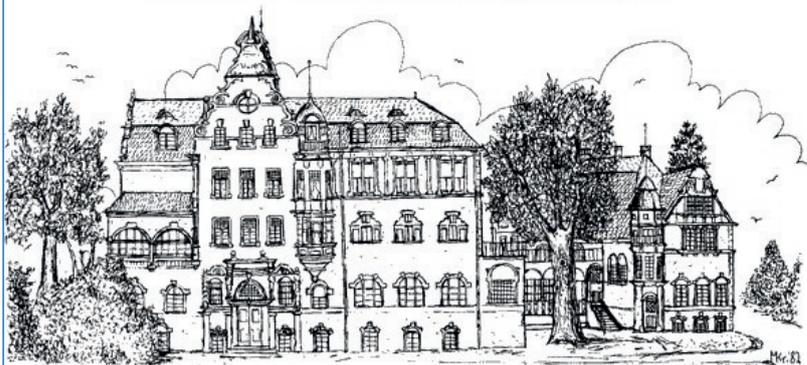
tionsbasiertes Vorgehen zur Migration in Service-orientierte Architekturen entwickelt.

Sowohl aktuelle Forschungsaktivitäten, als auch moderne Verfahren der Software-Industrie setzen hierzu auf modellgetriebene Techniken. Hierbei haben die Entwicklung effizient analysierbarer Software-Repositories sowie die Bereitstellung und Anwendung leistungsstarker Anfrage- und Transformationstechniken zentral zur Weiterentwicklung von Reengineering-Methoden und zum Bau mächtigerer Reengineering-Werkzeuge beigetragen. Repository-Strukturen (Metamodelle) zur grob-granularen Analyse mehrsprachiger Software-Systeme wie auch Strukturen zur fein-granularen Analyse einzelner, auch präprozessor-gesteuerter, Sprachen existieren ebenso wie Repository-Strukturen zur Repräsentation von Software-Architekturen.

Weitere aktuelle Forschungsgebiete im Kontext der Anwendungsmodernisierung umfassen die Weiterentwicklung fundierter Reengineering-Methoden und Techniken sowie deren Einbettung in Software-Entwicklungsumgebungen, um bereits während der Software-Erstellung konstruktiv die Langlebigkeit von Software-Systemen gewährleisten zu können. Die Verwendung großflächiger Software-Landschaften in Unternehmen erfordert die Anpassung der Reengineering-Techniken auf die Analyse, Wartung und Evolution von Unternehmensarchitekturen und Systemlandschaften. Die generelle Verbesserung der Zukunftsfähigkeit vorhandener Software-Systeme bedarf darüber hinaus klarer und detaillierter Kriterien zur reproduzierbaren Einschätzung der Wartbarkeit, die nur im Rahmen empirischer Untersuchungen erhoben werden können. Die Bereitstellung entsprechender Kriterienkataloge sowie passender Analyseverfahren ermöglicht die Zertifizierung von Software-Systemen vor dem Qualitätsziel Wartbarkeit, so dass Risiken der Software-Evolution nachvollziehbar, überprüfbar und entlang neutraler Kriterien einschätzbar werden. ■

## 12. Workshop Software-Reengineering

[www.uni-koblenz.de/sre/wsr2010](http://www.uni-koblenz.de/sre/wsr2010)



3.–5. Mai 2010

Bad Honnef

### Rainer Gimnich

IBM Software Group,  
SOA Advanced Technologies  
([gimnich@de.ibm.com](mailto:gimnich@de.ibm.com))

### Andreas Winter

Carl von Ossietzky Universität Oldenburg,  
Abteilung Software-Engineering  
([winter@se.uni-oldenburg.de](mailto:winter@se.uni-oldenburg.de))

# Mainframe- Ausbildung im Fokus – 600 Universitäten sind weltweit dabei

Das IBM Hochschulprogramm mit der besonderen Zielrichtung, die Mainframe-Ausbildung zu stärken, investiert weltweit mehrere Millionen Dollar in den Ausbau von Programmen für Hochschulen, Universitäten und Berufsakademien. Seit der Einführung des Programms in 2004 sind mehr als 600 Universitäten vertreten mit ca. 50.000 ausgebildeten Studenten. Ziel ist es, die Mainframe-Kenntnisse im Markt zu verbreitern und den Studenten erste Kenntnisse und praktische Fähigkeiten zu vermitteln. Dies ermöglicht vielen Studenten, gesteigerte Berufschancen und einen zukunftssicheren Job im Arbeitsmarkt zu finden, und gleichzeitig den Mainframe-Nutzern und –Kunden, weiterhin gute ausgebildete Mitarbeiter einzustellen.



Abb. IBM System z 10

Die steigende Zahl von teilnehmenden Hochschulen in diesem Programm reflektiert auch die große Bedeutung der Mainframe-Plattform in den Wachstumsmärkten mit neuen Kunden wie der Westpac Bank in Neuseeland, HDFC Bank in Indien, Kookmin Bank, Korea Exchange Bank etc. Zu den neuen Universitäten in der Initiative gehören die University of Pretoria

(Südafrika), Centro Paula Souza und FIAP in Brasilien sowie Southeast University in China. Schwerpunkt an den Hochschulen ist die Ausbildung im Rechenzentrumsbetrieb, Enterprise System Management und die Bedeutung von IBM System z / Mainframe Technologie bei den Themen niedrige Energiekosten, Serverkonsolidierung, Anwendungsmodernisierung, Anwendungsentwicklung und Cloud Computing.

In Deutschland nehmen bereits 22 Hochschulen an der IBM Mainframe Academy Initiative teil. Hierzu zählen die Universitäten Karlsruhe, Tübingen, Darmstadt, Dresden und Berlin. Die Lehrveranstaltungen konzentrieren sich auf die Betriebssysteme z/OS, z/VM, z/Linux, JCL, Datenbanken DB2, Transaktionsmonitore wie IMS und CICS, als auch Java und

WebSphere auf dem Mainframe. Hierbei geht es um die Modernisierung von PL/I-, Assembler- und Cobol-Anwendungen, die Integration in SOA und um moderne Cloud-Computing-Umgebungen. Gearbeitet wird in den Vorlesungen mit modernen eclipse-basierenden Werkzeugen in Form von Seminaren, Summer camps, eintägigen Veranstaltungen, Tutorien und Ringvorlesungen.

Studenten mit Mainframe-Kenntnissen können im IBM Student Opportunity System (<https://www.ibm.com/developerworks/university/students/sos/>) ihre Erfahrungen mit zukünftigen Arbeitgebern austauschen. Professoren können ihre Studenten bei der Jobsuche entsprechend unterstützen. Weitere Informationen zur IBM System z Academic Initiative finden sich unter [www.ibm.com/university/systemz](http://www.ibm.com/university/systemz).

Eine weitere Möglichkeit, in die spannende Mainframe-Welt einzutauchen, bietet die zSummer University im IBM Labor Böblingen. Die zweiwöchige Intensivschulung rund um das Thema IBM Mainframe und System z richtet sich bundesweit an Studierende der Informatik und verwandter Studiengänge und fand 2009 bereits zum vierten Male statt. Das Programm stellt hohe Anforderungen an die Teilnehmer. Die IBM-Dozenten aus dem IBM Labor und weiteren Geschäftsbereichen der IBM (Hardware/Software/Beratung) erarbeiten ein dichtes Vortrags- und Workshopprogramm zu Hard- und Software-Komponenten im Mainframe-Umfeld. Die Studenten hören Fachvorträge zu den Themenbereichen Architektur, Storage, Betriebssysteme, Firmware, Middleware sowie deren Einsatzgebiete bei den Kunden. Außerdem lernen sie natürlich die IBM näher kennen. Mit den neu erworbenen Kenntnissen lösen die Studenten dann spezielle Aufgaben und setzen diese in Kundenszenarien um. Mit IBM Kunden lernen die Studenten im Weiteren Anwendererfahrungen und -anforderungen. Mehr Informationen finden sich auf der Website [http://www-05.ibm.com/de/entwicklung/jobs/summer\\_uni.html](http://www-05.ibm.com/de/entwicklung/jobs/summer_uni.html).

Seit über 45 Jahren steht die Mainframe-Technologie für hochzuverlässige Computer-Transaktionen im High-Volumen-Umfeld bei Banken, Versicherern, Luftlinien, Verwaltung, Handel etc. Mainframe Server haben im Segment über 250.000 \$ einen Marktanteil von 30 % entsprechend IDC (2009). Durch die intensive Zusammenarbeit mit den Hochschulen weltweit wird sichergestellt, dass die Kunden weiterhin die Mainframe-Qualifikationen am Markt bekommen und gleichzeitig die Mainframe-Stärken ausbauen können.

Weitere Informationen zur Mainframe Community und Ausbildung:

<http://www-03.ibm.com/systems/z/advantages/charter/skills.html>

<http://znxtgen.org>

## Olaf Hendrik Kappel

Sales Leader WebSphere Software auf IBM System z Nordosteuropa, IBM Deutschland (kappel@de.ibm.com)

# Java Anwendungen auf dem Mainframe

## Sparen Sie Zeit und Geld mit Java auf dem Mainframe!

Java als Programmiersprache ist den Kinderschuhen entwachsen. Kurze Entwicklungszeiten, Portierbarkeit, Sicherheit und ein sehr breites Einsatzspektrum sind nur einige Vorteile der objektorientierten Sprache. Warum nicht auch die Vorteile von Java in der Massendatenverarbeitung nutzen? COBOL, PL/1 oder Assembler werden zunehmend auf ein Ressourcen-Problem stoßen, da es immer weniger Programmierer gibt, die diese Sprachen beherrschen.

Bei einem unserer Kunden setzen wir Java Anwendungen auf dem Mainframe in geschäftskritischen Geschäftsbereichen seit mehreren Jahren erfolgreich ein. Mit neuesten Technologien und einem speziell entwickelten Java-Framework lässt sich Java sowohl im Batchbetrieb als auch im Onlinebetrieb effizient anwenden. Ob in der Unix System Services Umgebung, als Batchprogramm zur Verarbeitung von Massendaten, als Stored Procedure Sprache für DB2 oder in anderen Einsatzgebieten.

Wir finden die Lösung für komplexe Anforderungen!

**Java Mainframe Aktion**  
Kostenlose Beratung und Präsentation  
von Java Anwendungen.  
Weitere Infos unter [www.tricept.de/mainframe](http://www.tricept.de/mainframe)  
oder telefonisch unter 0711/52 08 92-68

# Wie man das Außergewöhnliche in alltägliche Dinge hineinbringt.

Schon in einem Jahr werden mehr als 100 Millionen Zeilen Software-Code die Elektronik eines ganz normalen Autos steuern – bei einem Passagierflugzeug über 1 Milliarde. Wir erreichen bald einen Punkt, an dem ein Auto, eine Bank oder ein Flugzeug mehr als nur die Summe ihrer Einzelteile sind. Was all diese Dinge tatsächlich auszeichnet, ist die zugrundeliegende Software – dieses unsichtbare Netz, das alles mit Intelligenz durchdringt. Kein Wunder, dass im letzten Jahr schon 66 % aller entwickelten Produkte mit eingebetteter Software liefen. Heute ist Software von zentraler Bedeutung, um Unternehmen strategisch auszurichten. Aber: 41 % aller Software-Projekte scheitern immer noch daran, den Geschäftsnutzen und die Rentabilität zu steigern. IBM hat die einmalige Erfahrung, die Ressourcen und die Lösungen, damit international führende Unternehmen Software effizient entwickeln und bereitstellen können.

Smarte Unternehmen brauchen intelligente Software, Systeme und Services.

Also: Machen wir den Planeten ein bisschen smarter. Wie, erfahren Sie unter [ibm.com/delivery/de](http://ibm.com/delivery/de)

