

# Modernisierung einer ERP-Lösung durch Migration der technologischen Infrastruktur

Helge-Frank Zimpel, Sage bärer GmbH



Helge-Frank Zimpel begleitet bei der Sage bärer GmbH als Produktmanager Technology & Innovation die strategische Weiterentwicklung der Technologieplattform bärer open access (boa).

Ein Technologiewechsel des eingesetzten ERP-Systems ist oftmals kostenintensiv und mit hohem Umstellungsaufwand verbunden. Nur die wenigsten Anwender sind daher bereit, allein aus technischen Gründen ihre eingesetzten ERP-Lösungen zu modernisieren. Ein Umstieg auf die neueste Produktgeneration lässt sich nur über neue betriebswirtschaftliche Funktionen rechtfertigen, welche den Unternehmen helfen, ihre Prozesse zu optimieren. Somit halten die Anwender im gehobenen Mittelstand oft lange an ihrer Software fest.

Mit dem Aufkommen des Internets sahen sich die ERP-Hersteller mit grundlegend neuen technologischen Anforderungen an die Software konfrontiert. Einige Anbieter haben auf diese Veränderungen mit der kompletten Neuentwicklung von ERP-Systemen reagiert bzw. über Zukäufe entsprechende Systeme erworben.

## In diesem Beitrag lesen Sie:

- wie ein ERP-Paket auf eine Java-Plattform minimiert wurde,
- wie der fachliche 4GL-Code weitergenutzt wird,
- welchen Nutzen offene Standards und Open-Source-Technologien schaffen.

Eine Neuentwicklung ist jedoch ein lang andauernder Prozess, bis das Funktionsniveau gegenüber den bereits am Markt etablierten Systemen erreicht ist. Zudem ist sie mit beträchtlichen Investitionen durch die jeweiligen Hersteller verbunden, für die der Return on Invest weit in der Zukunft liegt. Für die Anwender kommt der Umstieg auf ein neuentwickeltes Nachfolgeprodukt oft einer Neuimplementierung gleich.

Andere Anbieter hingegen haben sich für eine evolutionäre Weiterentwicklung ihrer Produkte entschieden. Einer der möglichen Ansätze ist die Verwendung automatischer Portierungswerkzeuge, um die häufig in veralteten Programmiersprachen realisierten Lösungen auf eine aktuelle Programmierumgebung wie Java oder .NET zu portieren. Nachteil hierbei ist es, dass sich die Transformation vom Client/Server-Konzept auf eine neue, webbasierte Architektur mit diesen Werkzeugen trotzdem schwierig gestaltet. Es gelingt mit diesem Ansatz zwar, den Code zu portieren, nicht jedoch der Umstieg auf eine serverbasierte und webfähige Architektur. Viele Anbieter haben bis heute lediglich Teilbereiche ihrer Software modernisiert, indem sie etwa ihre Lösung mit einem optisch zeitgemäßen Frontend versehen haben oder einzelne, aktuelle Technologien wie die Unterstützung von Web-Services in ihre Produkte eingebaut haben.

## Eine alternative Migrationsstrategie

Auch der ERP Hersteller Sage bärer sah sich im Herbst 2003 vor der Herausforderung, sein ERP-Paket auf

eine neue technologische Basis zu stellen. Ausgangsbasis war ein System, das in klassischer Client/Server-Architektur, dem gängigen Architektur Paradigma der 90-er Jahre, entwickelt worden war. Durch Modernisierung der systemseitigen Infrastruktur, kombiniert mit der konsequenten Nutzung von Komponenten aus dem Open Source-Umfeld, konnte im Jahr 2004 mit der „dynamic“-Generation eine komplett Web-basierte Lösung vorgestellt werden, deren Grundlage eine dreischichtige, Server-zentrierte Architektur bildete. Die Laufzeitumgebung wurde hierbei vom Client auf den Server verlagert und ein Java-basierter und webfähiger Rich Client angebunden. Hierzu wurde die in C implementierte Runtime-Umgebung des bis dahin eingesetzten 4GL-Entwicklungswerkzeugs von Prolifics [1] entsprechend modifiziert. Dies war möglich, da der Quellcode des Entwicklungs-Tools unter einer BSD-ähnlichen Open Source-Lizenz für die Linux-Plattform zur Verfügung stand [2]. Für die Nutzung von Windows als Server-Betriebssystem wurde der C-Code rückportiert.

Zur CeBIT 2007 präsentierte Sage bärer mit der bärer open access-Plattform (kurz: boa) die nächste Evolutionsstufe: Mit einer modularen, Java-basierten Laufzeitumgebung auf dem Server sowie einer in Eclipse integrierten Entwicklungsumgebung konnte das bislang eingesetzte 4GL-Tool komplett abgelöst werden. Der bestehende 4GL-Code der ERP-Anwendung kann aber weiterhin innerhalb der neugeschaffenen Infrastruktur genutzt werden und wird nun innerhalb der Java Virtual

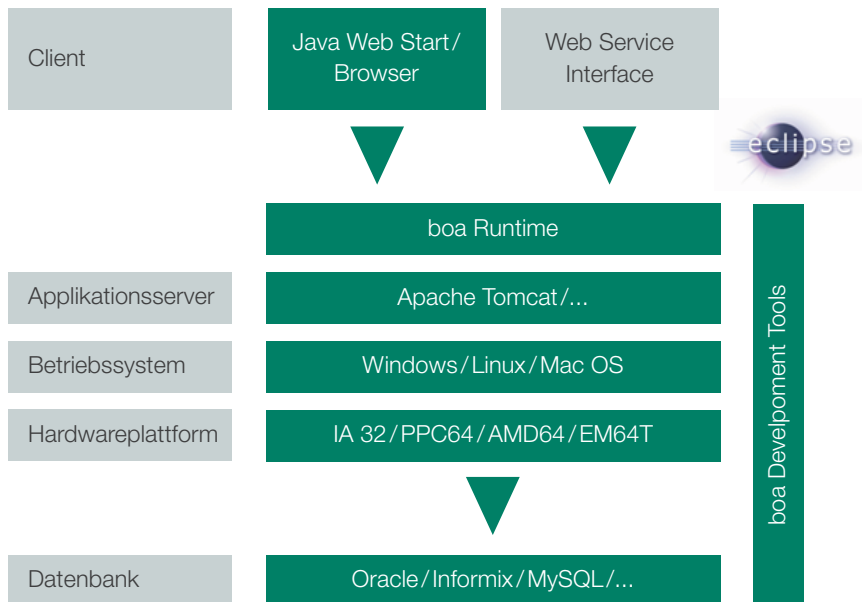


Bild 1: Web-basierte und plattformunabhängige Architektur.

Machine ausgeführt. Hierzu wurde zum einen ein Transcompiler entwickelt, der den bestehenden 4GL Code der ERP-Anwendung „on the fly“ in Java-Code überführt. Als Werkzeug für die Implementierung des Transcompilers wurde JavaCC (ein Java Parser Generator) genutzt. Des Weiteren wurde die Funktionsbibliothek des ehemals eingesetzten 4GL-Werkzeugs in Java re-implementiert. Die Vorgehensweise, für die Weiterentwicklung des ERP-Systems weiterhin eine 4GL-Sprache zu verwenden, folgt dem allgemeinen Trend, die Java-Plattform um die Unterstützung dynamischer Skriptsprachen zu erweitern. Derartige Sprachen sind leicht erlernbar und zeichnen sich allgemein dadurch aus, dass es keine strenge Typisierung von Variablen gibt. Kurze Round-Trip-Zyklen erlauben einen effizienten Entwicklungsprozess.

Die auf Basis des hier beschriebenen Ansatzes migrierte ERP-Anwendung umfasst ca:

- 4,8 Millionen Codezeilen
- 1000 Datenbanktabellen
- 1800 Programmmasken

## Anwendernutzen

Die neugeschaffene technologische Infrastruktur bietet mittelständischen

Unternehmen eine ganze Reihe an Vorteilen:

- Niederlassungen und Außendienst können per Internet flexibel auf die zentral betriebene Software zugreifen.
- Die komplett in Java entwickelte Technologiebasis ist plattformunabhängig und kann somit auf einer Vielzahl von Prozessorarchitekturen und Betriebssystemen ausgeführt werden.
- Durch Verzicht auf Bindung an bestimmte proprietäre und lizenzkostenpflichtige Technologien und Infrastrukturen einzelner Hersteller behalten die mittelständischen Unternehmen die notwendige Flexibilität und Entscheidungsfreiheit. Sogenannte Lock-In-Effekte werden hierdurch unterbunden.
- Durch die Unterstützung und Verwendung von Open Source-Technologien wird die Total Cost of Ownership deutlich reduziert. So ist ein Betrieb unter Linux, sowohl auf dem Client als auch auf dem Server möglich. Mit MySQL wird auf Datenbanksseite ebenfalls eine Open Source-Alternative unterstützt.
- Durch Integration von Web Services-Technologien ist die Plattform SOA-fähig. Gekapselte Funktionen

der ERP-Software lassen sich in Form von Services offenlegen und somit von außen ansprechbar machen. Die Nutzung von Standards garantiert, dass dies anwendungsübergreifend und unabhängig von bestimmten Plattformen und Programmiersprachen funktioniert.

- Die Unterstützung von Unicode ermöglicht es, Schrift- und Zeichensysteme wie zum Beispiel Chinesisch darzustellen.

## Bausteine der Plattform

Im Einzelnen besteht die boa Plattform aus den folgenden Komponente:

### Rich-Thin-Client

Ergonomisch stehen Web-Clients oftmals hinter klassischen Applikationen zurück. Sogenannte Rich Internet Applications (RIA) gewährleisten zum einen die hohe Usability traditioneller Desktop Anwendungen, haben aber zum anderen die Vorzüge einer klassischen Web-Anwendung (einfache Distribution, Plattformunabhängigkeit). Der Client erlaubt inkrementelle Updates und somit eine schnelle Reaktion auf einzelne Benutzereingaben. Der Rich-Thin-Client der Plattform ist auf Basis von Java / Swing implementiert und kann per Java Web Start oder mittels des Java Plugins innerhalb eines Web-Browsers genutzt werden. Der Client ist um Plugins erweiterbar, welche z.B. die Integration mit lokalen Office-Anwendungen erlauben.

Die Kommunikation mit dem Server geschieht mittels des HTTP-Protokolls. Somit stellen im Netz zwischen Client und Server befindliche Firewalls oder Proxy-Server kein Problem dar.

### Modularer Server

Der Server ist konform zum J2EE Standard und somit innerhalb gängiger Java-Applikationsserver ausführbar. Aufgrund seiner modularen Architektur ist der Server flexibel um weitere Bausteine erweiterbar und für unterschiedliche Einsatzszenarien konfigurierbar. Mittels Load-Balancing lässt sich die Anwenderlast auf mehrere Ser-

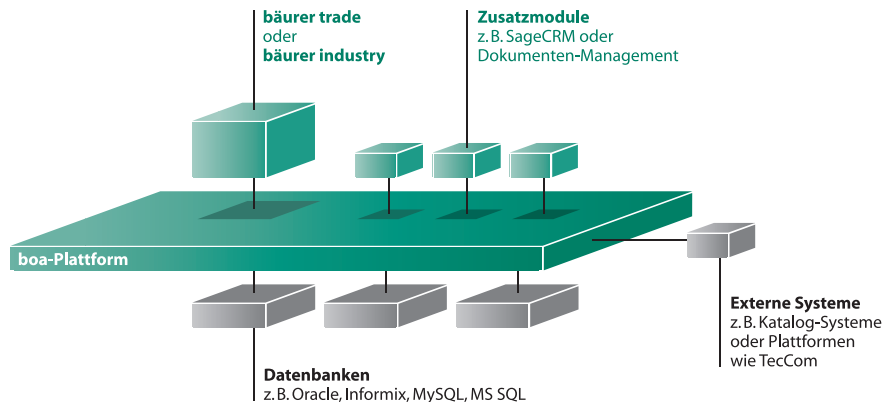


Bild 2: Einbindung zusätzlicher Lösungskomponenten externer Systeme.

ver verteilen und zudem Ausfallsicherheit gewährleisten.

Der Server basiert auf einer modularen Plugin-Architektur und ist aus einzelnen Server Components (kurz: SCOM) aufgebaut.

#### Entwicklungsumgebung

Die Entwicklungsumgebung ist auf Basis der Open Source Entwicklungsplattform Eclipse realisiert. Dieses Werkzeug kann auch von Partnern und Kunden genutzt werden. Eclipse wurde mittels sogenannter Plugins um spezifische Komponenten erweitert. Mit einem grafischen Screen-Editor lassen sich per Drag-and-Drop-Masken gestalten oder anpassen. Der Code-Editor ermöglicht die Entwicklung von Business-Logik in der für die Programmierung der Business-Logik verwendeten 4GL-Sprache. Eine Testumgebung und ein Debugger komplettieren das Entwicklungssystem. In Eclipse lassen sich zusätzlich weitere Plugins einbinden, etwa zur Integration mit Versionsverwaltungssystemen wie Subversion. Zusätzlich wurde auf Basis von JUnit ein Testframework bereitgestellt, mit dem sich Unit-Tests für den 4GL-Code implementieren und durchführen lassen.

#### Report Builder

Mit JasperReports wurde ein führendes Open Source-Reporting-Werkzeug in die Plattform eingebunden, für das komfortable graphische Werkzeuge für die Formulargestaltung zur Verfügung stehen. Durch Trennung von Datenaufbereitung und Layout erhalten die

Anwender ein hohes Maß an Flexibilität. Im ersten Schritt werden die Belegdaten in XML aufbereitet. Anschließend werden diese mit einem Formular-Layout zusammengeführt und hieraus der Ausgabebeleg erzeugt. Die Trennung von Daten und Layout ermöglicht es, unterschiedlichste Ausgabeformate zu verwenden. Weiterhin lassen sich bei Bedarf fremde Tools zur Weiterverarbeitung der XML-Daten anbinden.

#### Konkreter Nutzen durch SOA

Die Plattform ist durch Unterstützung von Web Services Technologien SOA-fähig. Dies bringt dem Mittelstand jenseits technologischer Schlagworte bereits heute konkrete Nutzenpotentiale [3]. So sind in die Plattform weitere optionale Lösungskomponenten (CRM, Dokumentenmanagement, eCommerce) auf Geschäftsprozessebene mit dem ERP integrierbar. Weiterhin lassen sich interne und unternehmensübergreifende Prozesse automatisieren, z.B. Bestellprozesse innerhalb der Lieferkette.

#### Fazit

Der Hauptvorteile des beschriebenen Migrationsansatzes sind:

- Investitionsschutz, da der bestehende fachliche Anwendungscode weitergenutzt werden kann.
- Der Austausch von Technologien ist mit weniger Risiko behaftet als die Neuentwicklung einer Applikation.
- Es wurde eine Basis geschaffen,

welche es erlaubt, die Plattform evolutionär weiterzuentwickeln und künftig weitere Technologien zu integrieren.

#### Literatur

- [1] Prolifics Panther: <http://www.prolifics.com/panther.htm>.
- [2] Prolifics Open Source Software for Linux: <http://sourceforge.net/projects/possl/>.
- [3] Catterfeld, C. ; Zimpel, H.: SOA im Mittelstand. In: ERP Management (Ausgabe 1/2008), S. 46-48.

#### Schlüsselwörter

ERP-Modernisierung, Service-orientierte Architekturen (SOA), Rich Internet Applications (RIA), Java, Eclipse

#### Modernisation of an ERP solution by migrating its technological infrastructure

Upgrading a running ERP system with a technological successor is often expensive and associated with high switchover costs. Consequently, only a very small minority of users is willing to modernize a productive ERP solution simply on technological grounds. An upgrade to the latest product generation can only be justified if it also offers new business management functions that help the enterprise optimize its processes. This is the reason larger SME users tend to cling on to often antiquated applications. The described alternative modernization approach of Sage bäurer shows that functional standstill or even regression do not necessarily have to be the consequence of a technological change.

#### Keywords:

Modernisation of ERP systems, Modernisation of ERP systems, Service-oriented Architecture (SOA), Rich Internet Applications (RIA), Java, Eclipse

#### Kontakt

Sage bäurer GmbH  
 Auf der Steig 6  
 78052 Villingen-Schwenningen  
 Tel.: +49 (0) 7721 9980-0  
 Fax: +49 (0) 7721 9980-200  
 E-Mail:  
[helge.zimpel@sagebaeurer.de](mailto:helge.zimpel@sagebaeurer.de)  
[www.sagebaeurer.de](http://www.sagebaeurer.de)