NOTITIA

AdNovum

Bemerkenswertes von und über AdNovum

EAI oder «sprechen Sie Mainframe?»

Lösungsansätze für EAI und Mainframe-Integration

Open Source in der AdNovum

Vorteile und Grenzen des Einsatzes von Open Source Software

IT-Security Programm-Management

Evaluation und Optimierung der IT-Sicherheit

Frühling 2005, Nr. 8

Integration und Security

Liebe Leserin, lieber Leser

In vielen Unternehmen ist der Mainframe weiterhin eine ganz zentrale Ressource. In erster Linie ist er der zentrale Data Tier, übernimmt zum Teil aber auch Applikations- beziehungsweise Business-Logik. Diese starke Position und die relative Einfachheit, mit der sich in Java dezentrale Applikationen erstellen lassen, machen eine sichere und transparente Integration des Mainframes nötig.
Die Terminalapplikationen der EDV-Frühzeit stellten nur geringe Anforderungen an die Bandbreite und liefen auf dem Mainframe selbst. Lediglich die Präsentation war Sache des

EAI oder «sprechen Sie Mainframe?»

Enterprise Application Integration (EAI) ermöglicht das Zusammenspiel moderner Anwendungen mit der oft über Jahre gewachsenen, heterogenen IT-Landschaft einer Unternehmung. Eine optimale Lösung berücksichtigt die bestehende technische und organisatorische Situation und wahrt Erweiterbarkeit und Offenheit des Gesamtsystems.

Von Kornel C.C. Wassmer

Die Einbindung bestehender (Mainframe-/Legacy-)Applikationen in eine moderne J2EE-Architektur kann als typische Aufgabe eines Integrators bezeichnet werden. Ausgehend von bewährten und bekannten Multi-Tier-Architekturen im dezentralen Umfeld mit Präsentations-, Business-Logik- und Persistenz-Schicht, gibt es bei der Adaption und Integration dieser oft über viele Jahre hinweg gewachsenen Strukturen interessante Probleme zu lösen, was die Arbeit für den Software-Ingenieur zu einer spannenden Herausforderung macht. Neben technischen Aspekten spielen in diesem Umfeld oft auch «kulturelle» Faktoren und unterschiedliches technisches Verständnis der beteiligten Entwickler eine Rolle.

In den folgenden Abschnitten beleuchten wir einige typische und mögliche Lösungen für solche Integrationsprojekte. Ausgangspunkt ist dabei stets die durchaus repräsentative Annahme, dass der Mainframe echte Business-Funktionen zur Verfügung stellt, also mehr ist als eine einfache (hochverfügbare) Datenbank. Diese Business-Funktionen können als so genannte Transaktionen aufgerufen werden und sind möglicherweise sogar mit Hilfe von Terminals beziehungsweise Terminal-Emulatoren direkt nutzbar. Sie sind in der Regel in COBOL, PL/1, RPG oder verwandten Sprachen implementiert.

Es entspricht dem Best-Practice-Ansatz, dass die J2EE-Applikation einer klassischen Multi-Tier-Architektur konform in Präsentation, Business-Logik und Persistenz-Tier gegliedert ist, und der Mainframe sozusagen als spezieller Business-Tier auftritt. Als weitere Randbedingung eines solchen Systems können Sicherheitsstrukturen identifiziert werden, wie sie zum Beispiel von einem Secure Reverse Proxy etabliert und von Applikationsservern transparent zur Verfügung gestellt werden, wobei es keine Rolle spielt, ob ein HTML-Browser oder ein Rich Client zum Einsatz kommt.

Integration

Die typische Integrationsaufgabe, das heisst die Implementierung einer durchgängigen und nahtlosen Interaktion der dezentralen Applikation (Client) mit dem Mainframe (Server) kann in zwei Teilbereiche gegliedert werden.

> Die Implementierung eines Abstraktionslayers für den transparenten Austausch eines proprietären Systems bleibt Wunschdenken.

Erstens müssen die grundlegenden technischen Voraussetzungen geschaffen werden, damit Bits und Bytes zwischen den Systemen fliessen können. In diesen Bereich gehören Themen wie TCP/IP, Sockets, Low Level Libraries, Client APIs oder Messaging.

Terminals. Der Grossteil der «Arbeit» erfolgte auf dem Mainframe und wurde vom Mainframe-Hersteller implementiert.
In der CORBA-Welt war der Mainframe noch auf eine sehr «bewusste» Art integriert, und die Entwickler besassen recht genaue Kenntnisse des Mainframes. In der Java-Welt ist der Mainframe nun gewissermassen hinter dem Applikationsserver verborgen, was die Integration anspruchsvoller macht, ist der Mainframe doch prinzipiell stark herstellerabhängig.
Die Mainframe-Integration muss hohen Anforderungen an Betreibbarkeit, Skalierbarkeit und Sicherheit genügen. Da auch auf Seiten des Applikationsservers Security-Vorkehrungen getroffen werden müssen, verlangt die Mainframe-Integration immer auch die Koordination und Abstimmung der unterschiedlichen Sicherheitsmechanismen. Damit aus den verschiedenen Teillösungen eine zuverlässige Sicherheitsarchitektur entstehen kann, braucht es unternehmensweit ein einheitliches Verständnis der Sicherheit. Wie man mit solider Kenntnis der Einzellösungen und der Fähigkeit, diese richtig zu koordinieren, zu einer abgerundeten Lösung kommen kann, ist im Hintergrundbericht zum Security Programm-Management beschrieben.
Im Zusammenhang mit der Sicherheit ist Open Source ein wichtiges Thema, steuert sie doch immer wieder relevante Technologien dazu bei. Wie die AdNovum dazu steht, erfahren Sie aus dem Interview mit dem CTO.

Stefan Arn

CEO AdNovum Informatik AG

Zweitens erfolgt auf Basis der technischen Strukturen eine semantische Koppelung beider Systeme, indem Sessions, Datenstrukturen, Meldungsprotokolle und Security-Mechanismen adaptiert und für beide Seiten kompatibel abgebildet werden. Dabei trifft der Integrator gerade im proprietären Umfeld, wie es typischerweise bei Gross- und Grösst-Umgebungen der Fall ist, ganz verschiedene Ausprägungen dieser Eigenschaften an. Während die technische Anbindung in der Regel kaum anspruchsvoll ist, stellt die semantische Integration eine grössere Herausforderung dar. In diesem Bereich können Fehlentscheidungen gravierende Konsequenzen haben, wenn zum Beispiel Security-Lücken entstehen, die Verfügbarkeit schlecht wird, der Ressourcenverbrauch zu gross wird, die Komplexität nicht mehr managebar ist oder die betriebliche Qualität und Stabilität mangelhaft wird. Gerade auf den ersten Blick verlockende Ansätze, die oft auf Drittkomponenten basieren, bergen die Gefahr, dass die erwähnten Punkte zu Problemen werden. Angeführt werden können hier: Ignorieren des Session-Paradigmas des Mainframes («der unendlich schnelle Logon»), Remodellierung der Datenstrukturen («die Applikationsentwickler verstehen sich nicht mehr») oder aufgebrochene Security Constraints («die verlorene User ID»). Speziell beim Versuch, ein Legacy-System mit geeigneter Abstraktion und Modellierung komplett zu verbergen, damit es später transparent ausgetauscht werden könnte, sollten beim Ingenieur die Alarmglocken läuten.

Wer integriert wen?

Bei der Verbindung einer J2EE-Applikation mit Mainframe-Strukturen stellt sich die Frage, auf welcher Seite der Adapter gebaut wird: Ist der Mainframe ein echter, technologiekonformer Service auf dem Business-Bus oder abstrahiert ein spezieller Wrapper/Connector den Mainframe?

Für beide Spielarten muss somit beurteilt werden, was es bedeutet, mit der zur Verfügung stehenden Technologie («Werkzeugkasten») eine Schnittstelle zur jeweils anderen Welt zu bauen, was mit der Beurteilung der typischen Problemfelder einfach beantwortet werden kann. Entweder werden J2EE-Strukturen ohne Java auf dem Mainframe gebaut, damit sich der Mainframe wie ein EJB- oder CORBA-Service präsentiert, oder aber J2EE-seitig müssen CommAreas und Meldungen interpretiert und proprietäres Connection Handling und Protokolle implementiert werden, wenn die Integration dezentral vorgenommen werden soll (wird der J2EE Client auf dem Mainframe selbst betrieben, entschärft sich die Aufgabe leicht).

Es liegt auf der Hand, dass der zweite Ansatz vorzuziehen ist. Protokollimplementierung und Datenstruktur-Handling sowie Typkonvertierung, Connection Handling und Security-Mechanismen sind klar eine Domäne der dezentralen (J2EE-)Welt; mit JCA (Java Connector Architecture) stehen auch nutzbare Strukturen zur Verfügung.

Natürlich können diese Bereiche auch in Mainframe-Umgebungen und Legacy-Systemen angegangen werden, nur entspricht es in der Regel nicht dem Aufgabengebiet eines Applikationsentwicklers, sich auf diesem Systemlevel zu bewegen. Programme dieser Stufe werden üblicherweise vom Hersteller oder System-Provider zur Verfügung gestellt, während im dezentralen Umfeld Kompetenz in Systemtechnik zur Grundbildung des Ingenieurs gehört, der die Aufgabe hat, die Applikation zu bauen.

> Die Integration erfolgt situativ mit ingenieurmässigem Vorgehen, das die konkreten Strukturen und Anforderungen umsetzt.

Java * COBOL = C++

Wir haben bereits erwähnt, dass der zur Verfügung stehende Werkzeugkasten ein wesentlicher Aspekt bei der Beurteilung von Integrationsfragen ist. J2EE bietet in diesem Bereich bereits ausgereifte Mechanismen wie JCA. Manchmal sind aber diese Möglichkeiten nicht ausreichend, weil zum Beispiel systemtechnische Randbedingungen und Eigenheiten des zu integrierenden Mainframes eine

Adaption erfordern, die über die Mächtigkeit der Patterns von J2EE hinaus gehen oder deren Möglichkeiten übersteigen. Für diese Fälle steht in der dezentralen Welt ein adäquates Werkzeug zur Verfügung, das erst noch nahtlos in ein J2EE-System eingefügt werden kann: C/C++!

Die flexibelste – aber zugegebenermassen auf den ersten Blick nicht einfachste – Art der Integration von Systemen ist unseres Erachtens die Implementierung eines echten Gateway als eigenständiger Service. Das Gateway adaptiert auf Service-Seite die Anforderungen von J2EE (zum Beispiel CORBA-Service) und auf Client-Seite den Mainframe.

Dieser Ansatz ist ideal, wenn für den Zugriff auf den Mainframe dezentral bereits ein API besteht zum Beispiel auf Basis von RPC (Remote Procedure Call), CICS External Call Interface oder einer proprietären Implementierung, welche dann in der Form einer C/C++ Library vorliegt. Der Aufbau eines Gateway darf überhaupt als elegante Lösung bezeichnet werden, kann doch damit am genauesten einer echten Service-Architektur nachgelebt werden, indem Systeme via Standard-Protokolle (IIOP, HTTP) gekoppelt und dadurch einzelne Komponenten für weitere Umsysteme verfügbar werden.

> Die professionelle Nutzung des in offenen Systemen verfügbaren Werkzeugkastens ist ein Schlüsselfaktor für betreibbare Systemintegrationen.

Kornel C. C. Wassmer

Kornel Wassmer, diplomierter Informatik-Ingenieur ETH, arbeitet seit 1996 in der AdNovum. Bei der Arbeit ist der Head of Development immer zur Stelle, wenn in der Entwicklung etwas anzubrennen droht, in seiner basellandschaftlichen Wohngemeinde als Feuerwehrofficer und Gemeinderat, wenn es wirklich, übungshalber oder politisch brennt. Sein Büro heisst denn auch «Pyrodrom» und sein Entwicklermotto «Software-Bau ist Ernsteinsatz».

Security

EAI im Umfeld von State-of-the-Art Security, wie sie typischerweise in Multi-Tier-Umgebungen zu finden ist (Java, Internet-Technologie, Secure Reverse Proxy), wirft auch in dieser Hinsicht spannende Fragen auf. Es ist zentral in einer solchen Architektur, dass die Credentials eines Benutzers früh in der Kette im Access Tier bei der Authentisierung entstehen («End-to-End») und nach einer Authentisierung pro Benutzer eine Session entsteht, die oft über mehrere Tiers und Services hinweg Ressourcen alloziert. Auf der anderen Seite stellt eine Legacy-Applikation meist restriktive Anforderungen an Login (Authentisierung auf dem Mainframe), Autorisierung (beispielsweise Funktionsberechtigung und Datenraum-Autorisierung) und Session-Verhalten. Es ist nun die Aufgabe des Integrators, diese beiden Welten in möglichst idealer Weise zusammenzubringen. Auch hier zeigt sich, dass Mainframe-Applikationen in der Regel nicht dafür geeignet sind, zum Beispiel SSL-Verbindungen und Zertifikatsinfrastrukturen anzubieten, was wiederum für den Bau von Integrationsstrukturen im dezentralen System spricht. Wenn die technische Connectivity bereits vorhanden ist, nur Standard-Protokolle und Zugänge vorhanden sind, oder wenn auf Seite Mainframe keine Kompetenz in diesem Bereich verfügbar ist, kann auch diesbezüglich ein Gateway oder – falls aus technologischer Sicht sinnvoll – eine Integration direkt im J2EE Application Server (via JCA) die Lösung sein.

Eine interessante Option diesbezüglich ist der Ansatz, ein echtes Security Gateway als Maschine physisch neben den Mainframe zu stellen («der 30-cm-Kabel-Ansatz»). Damit können durch Separierung auf Stufe Netzwerk Sicherheitsstrukturen etabliert werden, wie sie z.B. auch durch eine DMZ (Demilitarized Zone) im Internet-Access-Bereich bekannt sind.

Make or Buy?

Auch bei Integrationsprojekten stellt sich natürlich die Frage nach dem Einsatz von Standardkomponenten oder Produkten, die versprechen, die beschriebenen Probleme zu lösen. Als Entscheidungshilfe kann hier der Grad der Proprietärität der Mainframe-Infrastruktur dienen.

> Der Einsatz von XML als Integrationswerkzeug muss wohlbegründet und technisch fundiert sein.

Je proprietärer die Strukturen, Mechanismen, Sicherheitsmerkmale und Semantik, desto unwahrscheinlicher ist der Erfolg eines Produktes (wie viele verschiedene Anbieter können eine Integration einer weltweit einzigen, proprietären Umgebung verkaufen?). Daraus lässt sich folgern, dass proprietäre Strukturen (v.a. Applikationen) proprietär integriert werden sollten, da die Arbeit nur einmal gemacht wird und nicht wiederverwendbar ist.

«Make or Buy» kann auch von einer anderen Seite beurteilt werden. Generische Integrations-Frameworks nehmen für sich in Anspruch, auch proprietäre Strukturen optimal adaptieren zu können, was nicht zuletzt auch zu kurzen Release-Zyklen führt. Hier ist natürlich darauf zu achten, dass nicht «effizient unbrauchbare Software» entsteht, das heisst, dass das Entwicklerziel Oberhand über Wichtigeres wie Performance, Komplexität und Betreibbarkeit gewinnt. Dass gerade herstellerspezifische Frameworks einen Vendor- oder Kompetenz-Lock-in provozieren können, sei nur am Rande erwähnt. Beim Einsatz von Frameworks muss genau untersucht werden, in welcher Form vermeintliche Abstraktionen wirklich etabliert werden. Typische Fragen in diesem Bereich betreffen transparente, gesicherte Mainframe-Sessions, was oft zur Annahme führt, dass Mainframe-Applikationen stateless seien oder State unendlich schnell aufgebaut werden könne.

Ein weiterer Punkt ist die Frage, ob das Framework einen adäquaten Abstraktionslevel für den Applikationsentwickler bietet, gängig sind selbstbeschreibende Strukturen wie XML. Neben den Vorteilen für Versionierung und Validierung machen selbstbeschreibende Datenstrukturen aber nur Sinn, wenn der Konsument in der Lage ist, diese Meta-Information tatsächlich auch zu nutzen. In der Regel wird aber der Applikationsentwickler diese Struk-

Kornel C. C. Wassmer berät Projekte in Integrationsfragen.

turen zerlegen, die Meta-Information verwerfen und nur die rohen Daten benutzen, weil er sonst seine Funktionen gar nicht implementieren kann respektive weil er nicht in der Lage ist, bereits auf Stufe Mainframe-Integration im Business-Tier die Applikation nur noch im Meta-Modell zu beschreiben. Im Weiteren zeigt die Erfahrung, dass auf die durch selbstbeschreibende Meldungen vorgenommene technische Entkoppelung der Datentypen von Client und Server meist verzichtet werden kann, da die wesentlichen Abhängigkeiten ohnehin nicht verborgen werden können und Änderungen auf Server-Seite (Mainframe), die Auswirkungen auf die Datenstrukturen haben, im professionellen Umfeld planbar sind.

Anzufügen ist im Hinblick auf Security-Anforderungen, dass generische Client-Server-Frameworks oft an ihre Grenzen stossen, wenn sie die implizite Propagierung von Security-Kontext nicht zulassen oder die End-to-End Security-Kette aufbrechen, weil entweder kein sicherer Transport wie SSL zur Verfügung steht oder wenn sie sogar noch Meldungsmanipulationen (Mapping/Transformationen) zulassen.

Die verlockenden Features eines solchen Frameworks – ohne grossen Aufwand eine Verbindung zwischen Client- und Server-System zu etablieren – erweisen sich dann als unüberwindbare Hürde, was unter Umständen gravierende Sicherheitsmängel des Gesamtsystems nach sich zieht, wodurch das Framework zu einer Fehlinvestition werden kann.

EAI hat also einerseits Einfluss auf die ganze Systemarchitektur, wird andererseits aber auch stark durch die gewünschte Systemarchitektur geprägt (Werkzeugkasten). Die Erfahrung zeigt, dass in heute üblichen Systemumgebungen termingerecht elegante und schlanke Lösungen implementiert werden können, die massgeblich zum Gesamterfolg eines Projektes beitragen, ohne dass bei Betrieb und Wartung Abstriche gemacht werden müssen.

> Flexibilität und Dynamik von Frameworks sind in der Umsetzung oft nicht nötig, da die Strukturen planbaren Änderungen unterliegen.

Fazit

Enterprise Application Integration ist eine echte Ingenieur-Disziplin, sie adaptiert immer die aktuelle technische (und organisiatorische) Situation und insbesondere auch die verfügbare Kompetenz. Performance, Betreibbarkeit und Komplexität einer Lösung sind ein Mass für eine gute Integration. Daneben sollten aber auch allfällige Vendor-Lock-ins beachtet werden. Optimale Integrationen berücksichtigen die Eigenschaften des Mainframes, sie versuchen nicht, eine uniforme Sicht über die ganze Welt zu legen.

Die Erweiterbarkeit und Offenheit der Integration und damit auch des Gesamtsystems bleiben dabei stets gewahrt. Das hier beschriebene, situative Vorgehen macht es auch möglich, dass die Migrationsfähigkeit eines Gesamtsystems über den ganzen Lifecycle hinweg gewahrt bleibt, weil keine neuen starren Strukturen geschaffen werden.

Open Source in der AdNovum

Notitia unterhielt sich mit Stefan Wengi über den Umgang mit Open Source in der AdNovum und die Möglichkeiten und Grenzen für den Einsatz von OSS-Produkten in Kunden-Projekten.

Interview: Barbara Stammler

NOTITIA: Das Thema Open Source Software (OSS) wird seit einiger Zeit breit diskutiert. Die AdNovum wird allerdings kaum mit OSS in Verbindung gebracht. Spielt OSS bei Ihnen eine untergeordnete Rolle?

Stefan Wengi: Ganz im Gegenteil, Open Source Software hat in der AdNovum eine sehr zentrale Bedeutung. Bereits in den Anfangsjahren der AdNovum wurde im damals stark universitär geprägten Umfeld OSS eingesetzt. Ich erinnere mich, dass während meines Praktikums 1992/93 verschiedene OSS Tools vor allem in der Entwicklung eine wichtige Rolle spielten. Unser geringes Exposure bezüglich OSS hat sicher nicht zuletzt damit zu tun, dass wir lieber umsetzen als Strategien auf Halde zu produzieren.

In welchen Bereichen setzt die AdNovum heute OSS ein?

Ich unterscheide drei wesentliche Bereiche: unsere eigene Infrastruktur, die Entwicklungsumgebung mit dem entsprechenden Tooling und unsere Kunden-Projekte.

In unserer eigenen Infrastruktur setzen wir seit einem Jahr Linux auf dem Desktop ein, daneben betreiben wir Services wie Bugzilla, Twiki oder Postfix und verwenden Pakete wie OpenOffice oder Mozilla.

Die Entwicklungsumgebung ist sehr OSS-lastig, dies nicht zuletzt, weil die Entwickler selbst stark in diese Richtung tendieren. Beispiele für eingesetzte OSS Tools sind: Eclipse, Ant, XEmacs, Automake, gcc, PMD, Perl, ejbgen und xdoclet.

Im Bereich der Kundenprojekte ist sicher das Nevis Framework in erster Linie zu nennen mit OpenSSL, dem Apache Web-Server oder der EJBCA als Paradebeispielen. Kundenspezifische Applikationslösungen basieren häufig auf Struts, verwenden Log4j als Tracing Engine und werden bei Eignung und Kundenwunsch auf JBoss oder Tomcat betrieben.

> «Vor allem in der Entwicklung spielen Open Source Tools eine wichtige Rolle.»

Sie setzen seit einiger Zeit Linux auf dem Desktop ein. Welche Erfahrungen haben Sie damit gemacht?

Vielleicht noch kurz warum wir uns für Linux entschieden haben: Unsere Arbeitsplatzrechner mussten ersetzt werden, und gleichzeitig benötigten wir lokal auf dem Desktop dringend mehr Leistung. Die Intel-x86-Architektur ist in ihrem Preis-Leistungs-Verhältnis ungeschlagen, was den Entscheid für die Hardware relativ einfach machte. Als Betriebssystem kam Windows eindeutig nicht in Frage, und auf der Suche nach einem UNIX-basierten OS fiel uns der Entscheid für Linux als am weitesten verbreitetes UNIX auf der Intel-Plattform leicht.

Die bisherigen Erfahrungen mit Linux muss man als gemischt bezeichnen. Wir haben performanceseitig eine deutliche Verbesserung auf dem Desktop erreicht, bezüglich Funktionalität wurden die meisten Erwartungen erfüllt, und das Echo der Entwickler ist durchwegs positiv. Auf der anderen Seite haben sich einige Defizite hinsichtlich Qualität und Gesamtkonsistenz manifestiert. So funktionieren einige Dinge im Bereich Netzwerkintegration nicht auf Anhieb so, wie man sich das vorstellt und wie es zum Teil verkauft wird. Bezüglich Qualität stellen wir massive Schwankungen zwischen den verschiedenen Release-Versionen einer Distribution fest.

In welchen Bereichen wird OSS (bewusst oder unbewusst) nicht oder selten eingesetzt?

Es ist auch heute noch so, dass unsere primäre und bevorzugte Plattform Sun Solaris ist. Wir haben zwar im vergangenen Jahr bei einigen Kunden Linux-Projekte mit Pilotcharakter realisiert. Der Anteil am Gesamtvolumen ist momentan aber nicht sehr beeindruckend, obwohl die Erfahrungen eigentlich positiv waren.

Ein Bereich mit kleiner Bedeutung von Open Source Software sind die Informationssysteme mit grossen relationalen Datenbanken. Da gibt es

aus der OSS-Welt schlicht nichts, was sich zum Beispiel mit den Enterprise-Qualitäten von Oracle vergleichen lässt. Ausserdem ist dieser Bereich zu Business-kritisch, als dass man bereit wäre, bestehende, bewährte Mechanismen von heute auf morgen zu ersetzen.
Als letztes Beispiel möchte ich das Gebiet der clusterfähigen J2EE Container nennen, ein sehr aktuelles Thema vor allem auch in Kombination mit Grid Computing oder Blades. Was ich bezüglich Clustering-Mechanismen in OSS Containern bisher gesehen habe, hat mich noch nicht wirklich begeistert. In diesem Bereich ziehen wir zurzeit kommerzielle Produkte vor.

Sie verwenden also sehr viele Bausteine aus der Open-Source-Welt. Arbeitet die AdNovum auch aktiv an Open-Source-Projekten mit?
Es gibt verschiedene Open-Source-Projekte, an denen wir direkt mit Beiträgen oder indirekt durch Bug Reports oder Security Alerts mitarbeiten. So haben wir in den letzten Monaten einige Erweiterungen an der EJBCA an die Community zurückgegeben. Auch an OpenSSL oder den Apache Web-Server leisten wir immer wieder Beiträge. Die Erfahrung lehrt allerdings, dass nicht alle Beiträge in den jeweiligen Communities verstanden werden oder willkommen sind. Deshalb konzentrieren wir uns vor allem darauf, eine Open-Source-Strategie in beschränktem, kontrolliertem Rahmen zu verfolgen.

Wie ist diese Aussage zu verstehen?
Kunden, die eine Nevis-Lizenz erworben haben, arbeiten zum Teil sehr eng mit uns in Projekten zusammen. Dabei entsteht immer wieder Code, der von allgemeinem Interesse ist. Dieser fliesst bei entsprechendem Einverständnis in die betroffene Nevis Komponente zurück, so dass alle unsere Kunden davon profitieren können. Der Quellcode von Nevis ist insofern offen, als die Lizenznehmer ein Anrecht darauf haben. Dadurch können sie insbesondere Security Reviews durchführen.

Welches sind für Sie die wesentlichen Gründe, die für einen Einsatz von Open Source Software sprechen?
Für uns als Firma mit einem sehr ausgeprägten Technologiefokus spielt die Verfügbarkeit des Source Code eine ganz zentrale Rolle. In vielen Fällen hilft uns das bei der Auswahl, weil wir relativ schnell eine Aussage über die Qualität eines Projektes machen können. In allen Phasen erleichtert die Verfügbarkeit des Source Code die Fehlersuche und -behebung beträchtlich. Daneben sind natürlich das Risiko und die Abhängigkeit von der Unterstützung durch einen Hersteller niedriger.
Als zweiten wesentlichen Grund würde ich die in Open-Source-Projekten häufig vorhandene Dynamik bezeichnen. Die involvierten Entwickler vertreten oft einen ähnlichen Mindset wie wir, und der Kontakt zu ihnen ist einfach herzustellen. Dadurch kommen wir zu einem direkten Draht ins Engineering, was uns hilft, Ideen einzubringen und Probleme möglichst schnell aus dem Weg zu räumen. Dieser Draht zu den Entwicklern kann bei kommerziellen Produkten oft nicht etabliert werden.
Schliesslich spielen auch Security-Überlegungen eine sehr wichtige Rolle. Wir wissen alle, dass «Security by Obscurity» schlecht ist. Das trifft auch für nicht offengelegten Source Code zu. Wer garantiert mir denn, dass in einer SSL-Implementierung des Anbieters Y nicht irgendwo eine Backdoor enthalten ist, und wer überprüft diese Implementierung regelmässig auf Schwachstellen?

> «Bezüglich Qualität gilt für OSS genau dasselbe wie für kommerzielle Software: Es gibt sehr gute, sehr schlechte und alles dazwischen.»

Wann ist beim Einsatz von Open Source Software Vorsicht angebracht?
Ich finde es sehr gefährlich, eine Strategie zu verfolgen, die ausschliesslich auf OSS basiert. Es ist ja nicht so, dass OSS immer und überall besser als kommerzielle Software ist. Gewisse extreme OSS-Anhänger neigen zu einer Einseitigkeit, die aus meiner Sicht in einem Umfeld mit professionellem Anspruch unbedingt zu vemeiden ist. Genau so wichtig scheint mir die Bewertung der verschiedenen Projekte bei der Evaluation. In vielen Bereichen gibt es OSS-Lösungen wie Sand am Meer, was die Suche schwierig macht.
Ergänzend muss man betonen, dass die Suche nach geeigneter OSS das eigentliche Projektziel nicht überflügeln darf. Den Einsatz von Open-Source-Projekten, die sehr neu sind oder nicht mehr sehr aktiv weiterbearbeitet werden, vermeiden wir. Hingegen bewerten wir einen länger bestehenden, guten Track Record als positiv.

Prominente Open-Source-Projekte werden ständig von verschiedensten Personen und Institutionen nach Schwächen durchsucht. Erkannte Schwachstellen werden typischerweise in kürzester Zeit bekanntgegeben und eliminiert. Das ist ein sehr vernünftiger Umgang mit einer Problematik, die sich für jede Art von Software stellt, egal ob OSS oder kommerziell.

Stefan Wengi

Stefan Wengi, dipl. Informatik-Ingenieur ETH, arbeitet seit 1997 in der AdNovum. Seit dem Sommer 2002 amtet er als CTO mit einem breiten Aufgabenspektrum. Dazu gehören schwerpunktmässig die Definition der eigentlichen Technologiestrategie, die Bereitstellung einer guten Arbeitsbasis für die Entwickler, Mitarbeit bei der Weiterentwicklung des Nevis Framework, die Wahrnehmung von Beratungsmandaten bei Kunden und die Begleitung von Projekten auf der Technologieseite.
In seiner Freizeit macht er auf den Rollerblades die Radwege im Kanton Zürich unsicher und versucht, sich im Bereich KUK (Kultur und Kulinarik) weiterzubilden.

Wie bewerten Sie die Qualität von OSS?

Bezüglich Qualität gilt für OSS genau dasselbe wie für kommerzielle Software: Es gibt sehr gute, sehr schlechte und das ganze Spektrum dazwischen. Damit ist implizit auch bereits gesagt, dass es sicher nicht zielführend ist, Software vereinfachend zu kategorisieren als «OSS = gut» und «kommerzielle Software = schlecht».

Natürlich ist es beim OSS-Modell einfach, schnell etwas von geringer Qualität zu publizieren, das spricht sich typischerweise in der Entwicklergemeinschaft herum. Und es gibt sicher genügend Beispiele für qualitativ hochstehende Open Source Software wie etwa den Apache Web-Server, Mozilla oder BSD.

Wie bewerten Sie aus Ihrer Sicht die Kostenseite?

Die Manager unter unseren Lesern muss ich leider enttäuschen: Wir operieren nicht mit hochentwickelten TCO-Modellen (Total Cost of Ownership), weil ich schlicht nicht glaube, dass sie uns etwas bringen würden.

Der Einsatz von Open Source Software in der Entwicklung erspart uns hohe Lizenzkosten und ermöglicht es, eine Vielfalt von Tools einzusetzen, die wir uns in der kommerziellen Variante wohl kaum alle leisten würden.

Für die Nevis Middleware bewerte ich die Kostenseite ganz klar als positiv. Dank dem Multiplikatoreffekt und einem bewussten Einsatz von OSS können wir unseren Kunden ein sehr flexibles Framework zu einem konkurrenzfähigen Preis anbieten und haben das System bei der Fehleranalyse voll im Griff.

Bei der ganzen Diskussion um die Kosten darf man nicht vergessen, wo diese grösstenteils anfallen, nämlich in der Wartung und in noch stärkerem Mass bei der Ablösung. Gerade bei letzterer verspreche ich mir durch die mehrheitlich hohe Standard-Compliance von OSS einen sehr positiven Effekt auf der Kostenseite.

Welche Faktoren machen den Einsatz von OSS erfolgreich?

Primär muss die Kompetenz vorhanden sein, mit der enormen Menge an «gratis» verfügbarem Code etwas Konstruktives anzufangen. Das beginnt mit einer gesunden, technologiefokussierten Haltung bei der Evaluierung von in Frage kommenden Open-Source-Produkten für einen konkreten Einsatz und führt dann nahtlos über zu einer vernünftigen Hemmungslosigkeit beim Eintauchen in fremden Code.

> «Die Verwendung von Open Source Software ermöglicht uns auch in der Zukunft kurze Innovationszyklen im Interesse unserer Kunden.»

Dann ist es aber auch ganz wichtig, dass man in der Lage ist, zu erkennen, wann Open Source Software fehl am Platz ist und nicht eingesetzt werden soll.

Welche Vorteile bietet OSS den Kunden von AdNovum?

Unsere Kunden, die heute erfolgreich mit AdNovum-Software arbeiten, profitieren sicher davon, dass wir OSS-Produkte verwenden. So ist beispielsweise die Fülle von Features, die wir ihnen im Nevis Framework bieten können, nicht zuletzt der eingesetzten Open Source Software zu verdanken. Die Verwendung von OSS ermöglicht uns auch in der Zukunft kurze Innovationszyklen im Interesse unserer Kunden.

Die Security-Aspekte habe ich ja bereits angesprochen. In diesem Kontext darf nicht unerwähnt bleiben, dass wir im Rahmen unserer Wartungsverträge die Security Alerts in der Community aktiv verfolgen und unsere Kunden über die konkreten Auswirkungen informieren. Dass wir bei Erkennen einer Sicherheitslücke in kürzester Zeit einen Patch-Release zur Verfügung stellen, gehört natürlich mit dazu.

Kunden, welche die AdNovum um Unterstützung beim Einsatz von Open Source Software angehen, bieten wir die Serviceleistung eines professionellen Release Managements und Packagings. Selbstverständlich implementieren unsere Entwickler im Rahmen eines solchen Auftrags auf Kundenwunsch auch spezifische Erweiterungen. ■

IT-Security Programm Management

Security, insbesondere die IT-Security, umfasst ein sehr breites Themenspektrum, das sich keineswegs auf Technologie beschränkt. Die beteiligten Menschen und die Organisation sind im Gesamtkontext eines Unternehmens nicht weniger wichtig.

Von Michael Müller

Die AdNovum Informatik ist bekannt für die Durchführung komplexer IT-Engineering-Projekte. Vor und während deren Umsetzung fungiert die AdNovum immer häufiger auch als kompetenter Beratungspartner für sicherheitsrelevante Aspekte, welche diese Projekte selbst nur indirekt betreffen. Oft führt eine initiale Beratung dazu, dass der Kunde eine genauere Evaluation seiner IT-Sicherheit wünscht, als deren Resultat schliesslich ein Massnahmenkatalog entsteht. Die Kombination aus Beratung und Gesamtleitung der IT-Sicherheitsprojekte bei einem Kunden fassen wir unter dem Begriff Security Programm-Management zusammen.

Hauptziel des Security Programm-Managements ist es, sicherzustellen, dass Organisation, Prozesse und Technologien im Security-Bereich aufeinander abgestimmt sind und sich gemeinsam in der durch die IT-Sicherheitsstrategie vorgegebenen Richtung bewegen.

Ein Security-Programm verschafft Klarheit über die vorhandene IT-Security-Infrastruktur, die Organisation und die Prozesse, identifiziert die wesentlichen Probleme, definiert die notwendigen Massnahmen und setzt diese nach der Genehmigung durch das Management auch erfolgreich um.

Security-Programm der AdNovum

Ein von der AdNovum durchgeführtes Security-Programm besteht aus zwei Hauptphasen. In der ersten Phase (Konzeptphase) werden eine Bestandesaufnahme, eine Problemanalyse, Lösungsvorschläge sowie eine Roadmap erarbeitet. In der zweiten Phase (Umsetzungsphase) geht es darum, die Roadmap erfolgreich umzusetzen.

Konzeptphase

Der erste Schritt der Konzeptphase besteht in der Erarbeitung einer Bestandesaufnahme, in welcher die bestehende IT-Security anhand der drei Dimensionen Organisation/Prozesse, Menschen und Technologie untersucht wird. Im einfachsten Fall sind alle Prozesse, die Organisation, die vorhandenen Technologien und die gesamte IT-Security-Architektur schriftlich dokumentiert und die Realität entspricht genau diesen Beschreibungen. Selten ist es jedoch so einfach, und die grosse Herausforderung der Bestandesaufnahme besteht darin, die gelebte IT-Sicherheit anhand von Interviews festzuhalten. Es ist wichtig, dass die Interviews möglichst breit gefächert durchgeführt werden. Mitarbeiter aus dem Management (auch aus dem Nicht-IT-Management), der IT-Security-Organisation, IT-Operations, Entwicklung, Test, Architektur und nach Bedarf weiteren Abteilungen müssen befragt werden. Aus Effizienzgründen lohnt es sich, in den Bestandes-Interviews auch direkt auf bekannte Probleme einzugehen.

Die nachfolgenden Beispiele aus der Praxis zeigen, dass oft Koordinations- und Verbesserungsbedarf besteht.

- Im Interview mit dem Management kommen die hohen Kosten zur Sprache. Das Management bekommt die IT-Security als Blackbox zu sehen und versteht in vielen Fällen nicht, welche Risiken die vorhandenen Sicherheitsprojekte genau abdecken und ob diese im Hinblick auf die Risikoverminderung ihr Geld wert sind. Immer wieder kommen auch Themen wie regulatorische Vorgaben oder Business Continuity im Katastrophenfall auf, da diese Themen für das Management am ehesten greifbar sind.
- Die Entwicklungsabteilung stellt fest, dass zu wenig klar ist, welche Vorgaben von Seiten der IT-Security zwingend sind, und dass zu wenig Know-how für die effiziente Umsetzung vorhanden ist. Die IT-Sicherheit hat keine hohe Priorität, wichtig ist, dass die geplanten Marktvorhaben nicht gefährdet werden.
- Die IT-Architektur-Abteilung bekommt ständig Anfragen betreffend IT-Sicherheitsarchitektur, verfügt aber nicht über genügend Ressourcen, um detaillierte Vorgaben zu definieren und diese über einen Sign-off-Prozess in die Projekte einfliessen zu lassen.
- Die Hauptverantwortliche der IT-Security berichtet über die erarbeiteten Vorgaben (Policies, Prozeduren usw.), die bisher zu wenig in die Praxis umgesetzt wurden. Es ist nicht klar, wer für die Umsetzung veranwortlich ist, und es fehlen klare Aufgabendefinition und Abgrenzung der Kompetenzbereiche. Den erarbeiteten Vorgaben wird aus Sicht der IT-Sicherheit zu wenig Beachtung geschenkt oder einer «unsicheren» Lösung aus politischen Gründen den Vorzug gegeben.

> Das IT-Security Programm-Management stellt sicher, dass Organisation, Prozesse und Technologien die IT-Sicherheitsstrategie unterstützen.

Als Resultat der Bestandesaufnahme entsteht eine Gesamtübersicht der IT-Security, wie

Michael Müller

Michael Müller studierte an der ETH Zürich Vermessungsingenieur und arbeitete danach als Management-Berater für Grossfirmen in ganz Europa. Sein vertieftes IT-Security Know-how erwarb er als Entwickler im Bereich Security in der AdNovum, im Nachdiplomstudium Informatik an der ETH Zürich und durch die gerade bestandene CISSP-Prüfung. Seit einem Jahr ist er in der AdNovum als Security Programm-Manager für diverse Sicherheitsprojekte verantwortlich. Neben der Arbeit treibt er Sport, bildet sich weiter oder ist auf Reisen.

Michael Müller ist als IT-Security Programm-Manager für Sicherheitsprojekte verantwortlich.

sie in den vorhandenen Dokumenten definiert und gemäss den Aussagen in den Interviews gelebt wird.

Nach der Aufnahme der bestehenden IT-Sicherheit beginnt die Problemanalyse. Auf dem Hintergrund der grossen Erfahrung der AdNovum im Bereich IT-Security und der durchgeführten Interviews können die wichtigsten Probleme sehr effizient identifiziert werden. Die Abstimmung der ersten Erkenntnisse mit allen Parteien ist zentral. Die Art der Probleme ist dabei sehr vielfältig, die Spannweite reicht von technischen bis hin zu organisatorischen auf Managementstufe.

Zusätzlich zu den konkret in den Interviews angesprochenen Sachverhalten können sich weitere Problemstellungen herauskristallisieren wie etwa:

- Netzwerkzonenplanung: Eine Zonenplanung ist nur ansatzweise vorhanden. Eine DMZ für den Zugriff vom Internet auf das Intranet ist vorhanden, doch stehen alle internen Systeme in der gleichen Netzwerkzone. Die Verbindungen innerhalb dieser Zone sind unverschlüsselt und nicht authentisch, dadurch können zum Beispiel versendete Passwort-Hashes von einem Angreifer abgefangen und beliebig oft zur Ausstellung eines Authentisierungskontextes wieder verwendet werden.
- Business Continuity Planning: Datenbackup an einem zweiten Standort ist vorhanden, doch wurde dem Thema Organisation und Planung im Notfall zu wenig Beachtung geschenkt.
- Prozesse: Es existiert kein IT-Risikomanagement-Prozess. Ohne diesen kann nicht sichergestellt werden, dass die kosteneffizientesten Massnahmen gewählt werden.
- Benutzerverwaltung: Die Benutzerdaten werden grösstenteils dezentral in den Applikationen verwaltet, dabei ist nicht sichergestellt, dass beim Austritt eines Mitarbeiters die ihm zugewiesenen Rechte überall gelöscht werden.
- Passwörter: Die Mitarbeiter müssen sich viele Passwörter merken (und periodisch erneuern), die für jede Applikation wieder eingegeben werden müssen, darunter leiden Effizienz und Sicherheit.

Nach der Identifikation und Beschreibung der Probleme müssen diese gewichtet und kategorisiert werden. Dabei gilt es, die Risiken richtig einzuschätzen und zusammen mit den Businessverantwortlichen zu priorisieren.

Basierend auf der Problemanalyse werden zu Beginn noch wenig detaillierte Lösungsvorschläge ausgearbeitet, zum Beispiel:

- Aufbau von IT-Security Basisdiensten, wie Authentisierung, Autorisierung, Audit-Trail, PKI usw.
- Definition eines Prozesses für die Umsetzung der IT-Security-Vorgaben in den Projekten (Sign-off-Prozess).
- Aufbau einer zentralen Benutzerverwaltung (Secure Identity Management).
- Einführung einer starken Benutzerauthentisierung für Web-Applikationen mit X.509-Zertifikaten der hauseigenen PKI, gleichzeitig Aufbau eines Single Sign-on.
- Vorschläge für die Absicherung (Verschlüsselung, Authentisierung) von kritischen Knoten-Knoten-Verbindungen in unsicheren Netzwerken.
- Definition der noch fehlenden Rollen in der IT-Security-Organisation, wie zum Beispiel IT-Security-Architekt, IT-Security-Risikomanagement und IT-Security-Engineering.

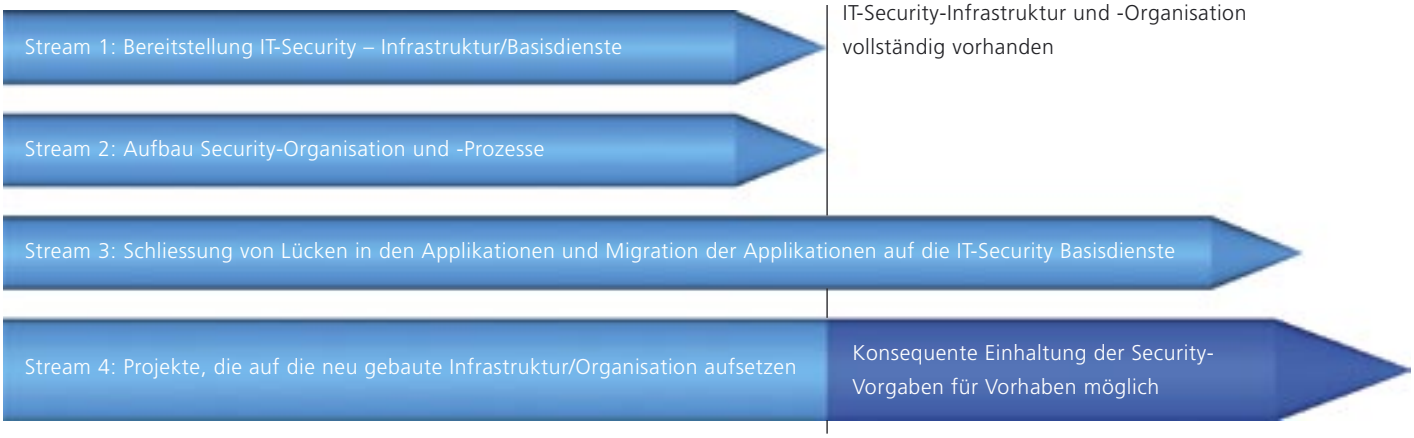
und von Prozessen, wie beispielsweise IT-Security-Risikomanagement, IT-Security-Sign-off und Notfallplanung.

Neben dem Aufbau der Infrastruktur und der neuen Organisation darf nicht vergessen werden, dass in den bestehenden Applikationen, Netzwerken, Systemen usw. teilweise grössere Lücken vorhanden sind, die nach einer Risikoabschätzung unbedingt geschlossen werden müssen. Dabei handelt es sich beispielsweise um die Verwendung schwacher Verschlüsselungsmethoden, eine grosse Anzahl unnötig offener Ports, die unsichere Übertragung von Passwörtern, die Verwendung interaktiver technischer User oder eine ungenügende Nachvollziehbarkeit aufgrund fehlender Audit-Trails. Neben dem unmittelbaren Schliessen von Lücken können auch mittel- und langfristige Migrationen bestehender Applikationen auf die im Stream 1 aufgebauten Sicher-

Erfahrung in der Durchführung komplexer IT-Security-Projekte zum Tragen und das in der Konzeptphase aufgebaute Know-how kann in die Projekte einfliessen.

Die Details der Umsetzungsphase hängen naturgemäss stark von den Erkenntnissen aus der Konzeptphase ab. Je nach Ausrichtung und Schwerpunkt der Lösungsvorschläge bewegt sich die Umsetzungsphase zwischen rein organisatorischen und hochgradig technischen Projekten.

Dem Security Programm-Manager kommt in der Umsetzungsphase die Rolle des Gesamtkoordinators zwischen den einzelnen Projekten, den Security-Verantwortlichen und den verschiedenen involvierten Organisationseinheiten zu. Der Aufwand für diese Funktion darf nicht unterschätzt werden und kann zusammen mit den Beratungsaufgaben leicht eine 100%-Stelle umfassen.



Die vier Hauptstreams der Programm-Roadmap.

Die Lösungsvorschläge müssen detailliert werden, dabei zählt der IT-Security Programm-Manager auf die langjährige Erfahrung der AdNovum in der Konzeption und Umsetzung von komplexen IT-Sicherheitsprojekten. Diese Erfahrung stellt sicher, dass die Lösungsvorschläge und Massnahmen zu einer realistischen Roadmap führen.

Wie in der Abbildung aufgezeigt, kann die ausgearbeitete Roadmap in vier grobe Streams unterteilt werden:

Im ersten Stream wird die IT-Security-Infrastruktur aufgebaut. Dazu gehören die zentralen Security-Basisdienste (etwa Authentisierungs- und Autorisierungsservice, Audit-Trail, PKI) sowie ein zentrales und sicheres Identity Management.

Der zweite Stream befasst sich mit dem Auf- oder Umbau der Security-Organisation

heitsbasisdienste zum Stream 3 gezählt werden. Der vierte Stream umfasst alle Projekte, die auf die neue Infrastruktur, Prozesse und Organisation aufbauen können.

Wie man sich unschwer vorstellen kann, sind von einem derartigen Vorhaben sehr viele Personen und Organisationseinheiten betroffen. Für eine erfolgreiche Realisierung der Roadmap in der Umsetzungsphase ist es deshalb unerlässlich, dass sie vollumfänglich durch das Management getragen wird.

Umsetzungsphase

Die realistische Definition der Roadmap und der Projektaufträge unter Einbezug der Unternehmenskultur und der vorhandenen Ressourcen bildet die solide Basis für eine erfolgreiche Umsetzung. Wird die Roadmap durch die AdNovum umgesetzt, kommt ihre

Fazit

Die Beurteilung und Verbesserung der gesamten IT-Sicherheitslandschaft ist eine komplexe Aufgabe, die nur interdisziplinär gelöst werden kann. Der Einbezug der Unternehmenskultur und aller Parteien vom Business-Management über die IT-Verantwortlichen bis zu den IT-Security-Officers ist zentral bei der Umsetzung eines IT-Security-Programms. Der Programm-Manager braucht neben politischem Taktgefühl und vertieften Kenntnissen des ganzen Spektrums der Sicherheit auch Erfahrung in der Koordination grosser Projektvorhaben. Die AdNovum stellt nicht nur eine kompetente Sicherheitsberatung der Kunden sicher, sondern kann die daraus resultierenden Lösungsvorschläge aufgrund ihrer umfassenden Engineering-Erfahrung auch erfolgreich umsetzen. ■

Industrialisierung in Nadelstreifen

Auf der Suche nach neuem Wertschöpfungspotenzial unterziehen heute viele Finanzdienstleister ihre Fertigungstiefe einer Überprüfung. Sie vollziehen damit einen Schritt, den die Fertigungsindustrie schon vor Jahren erfolgreich gegangen ist.

Patrick Comboeuf, Siemens Schweiz AG, Zürich

Was um die Jahrtausendwende vor allem von Marketing-affinen IT-Dienstleistern und Softwareunternehmen als Vision propagiert wurde, ist heute Tatsache: Banken überprüfen ihre Fertigungstiefe entlang der Wertschöpfungskette. Mit der damit einhergehenden Offenheit für neuartige brancheninterne und -fremde Kooperationen folgen sie dem Beispiel der verarbeitenden Industrie, die diesen Transformationsprozess in den 80er- und 90er-Jahren mit zum Teil einschneidenden Auswirkungen gemeistert hat. Entscheidend für den Erfolg, elementare Bankprozesse nach industriellen Parametern auszurichten, ist ein gesamtheitlicher Ansatz, der auch mittelfristig noch nachhaltige Effizienz- und Flexibilitätspotenziale zu erschliessen vermag.

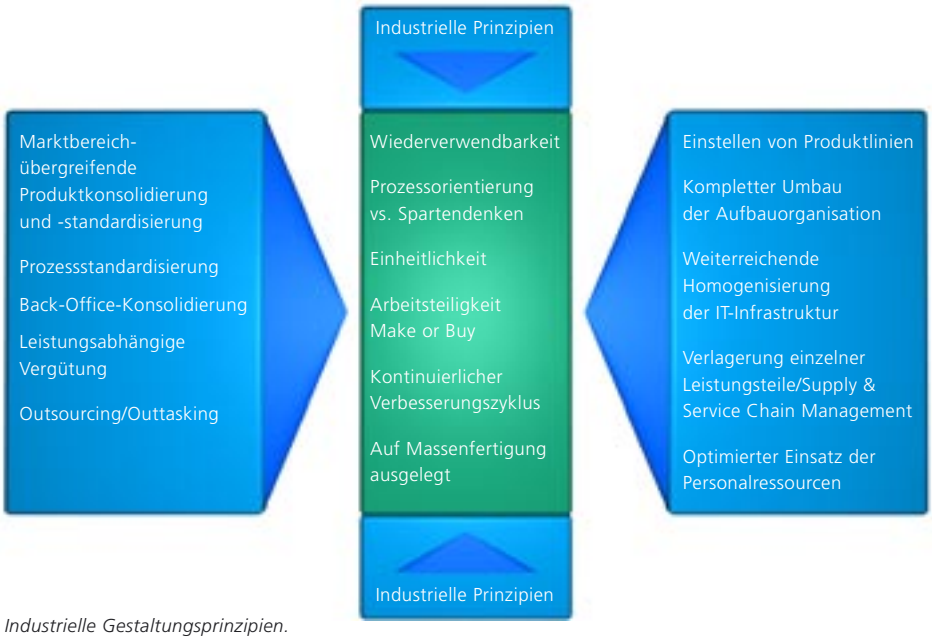
Während die Industrie auf so genannte «Lean Production» setzte, lagen die Schwerpunkte bei Banken in anderen Bereichen. Gerade um die Jahrtausendwende forcierten sie Kundensegmentierung, Servicedifferenzierung sowie die Erweiterung ihrer Vertriebskanäle – alles Initiativen, wo insbesondere den Kosten keine übermässige Relevanz zukam.

Damit diese erhöhten Anforderungen an die Kosteneffizienz erfüllt werden können, muss sich das moderne Bankmanagement verstärkt am Vorbild von Industrie und Handel orientieren. Wie diese Branchen sollen auch im Finanzdienstleistungssektor industrielle Gestaltungsprinzipien konsequent angewendet und die Wertschöpfungskette einem Redesign unterzogen werden mit einem grösseren Anteil an Fremdprodukten und einer allgemein geringeren Fertigungstiefe.

Kurzfristig werden die meisten Banken ihr Augenmerk auf die Optimierung ihrer Wertschöpfungskette und die prägnante Definition der Kernkompetenzen richten. Dabei gilt es, Geschäftsfelder zu wählen, die sich für Kooperationen in und ausserhalb der Branche eignen. Der Handlungsbedarf ist gross: Derzeit werden nach Meinung vieler Experten fast 70 Prozent der Wertschöpfung bankintern erbracht. Professor Dr. Hans Geiger vom Institut für schweizerisches Bankwesen an der Universität Zürich rechnet jedoch damit, dass sich diese «ausgesprochen grosse Fertigungstiefe innerhalb weniger Jahre auf 50 Prozent und weniger reduzieren wird». Damit steht sowohl die Business- als auch die Betriebsseite einer jeden Bank vor spannenden Herausforderungen in den kommenden Monaten.

Siemens Business Services

Siemens Business Services ist ein international führender IT-Service-Anbieter. Der Siemens-Bereich bietet Leistungen entlang der gesamten IT-Dienstleistungskette aus einer Hand, vom Consulting über die Systemintegration bis zum Management von IT-Infrastrukturen. Beim Outsourcing und der IT-Wartung zählt Siemens Business Services zu den Top-ten-Anbietern weltweit. Im Geschäftsjahr 2004 betrug der Umsatz rund 4,7 Mrd. EUR, 76 Prozent wurden ausserhalb des Siemens-Konzerns erzielt. Derzeit beschäftigt das Unternehmen weltweit ca. 36.100 Mitarbeiter (www.siemens.com/sbs).



Industrielle Gestaltungsprinzipien.

Impressum

Herausgeber:
AdNovum Informatik AG
Corporate Marketing
Röntgenstrasse 22
CH-8005 Zürich
Telefon 044 272 61 11
Telefax 044 272 63 12
E-Mail info@adnovum.ch
www.adnovum.ch
Verantwortlich und Redaktion:
Barbara Stammler, Manuel Ott
Gestaltung und Realisation:
Rüegg Werbung, Zürich
Fotografie:
Gerry Nitsch, Zürich