

NOTITIA

ADNOVUM

BEMERKENSWERTES VON UND ÜBER ADNOVUM

Information im Griff

Software lässt sich nicht im Blindflug produzieren

Software-Entwicklung im Umbruch

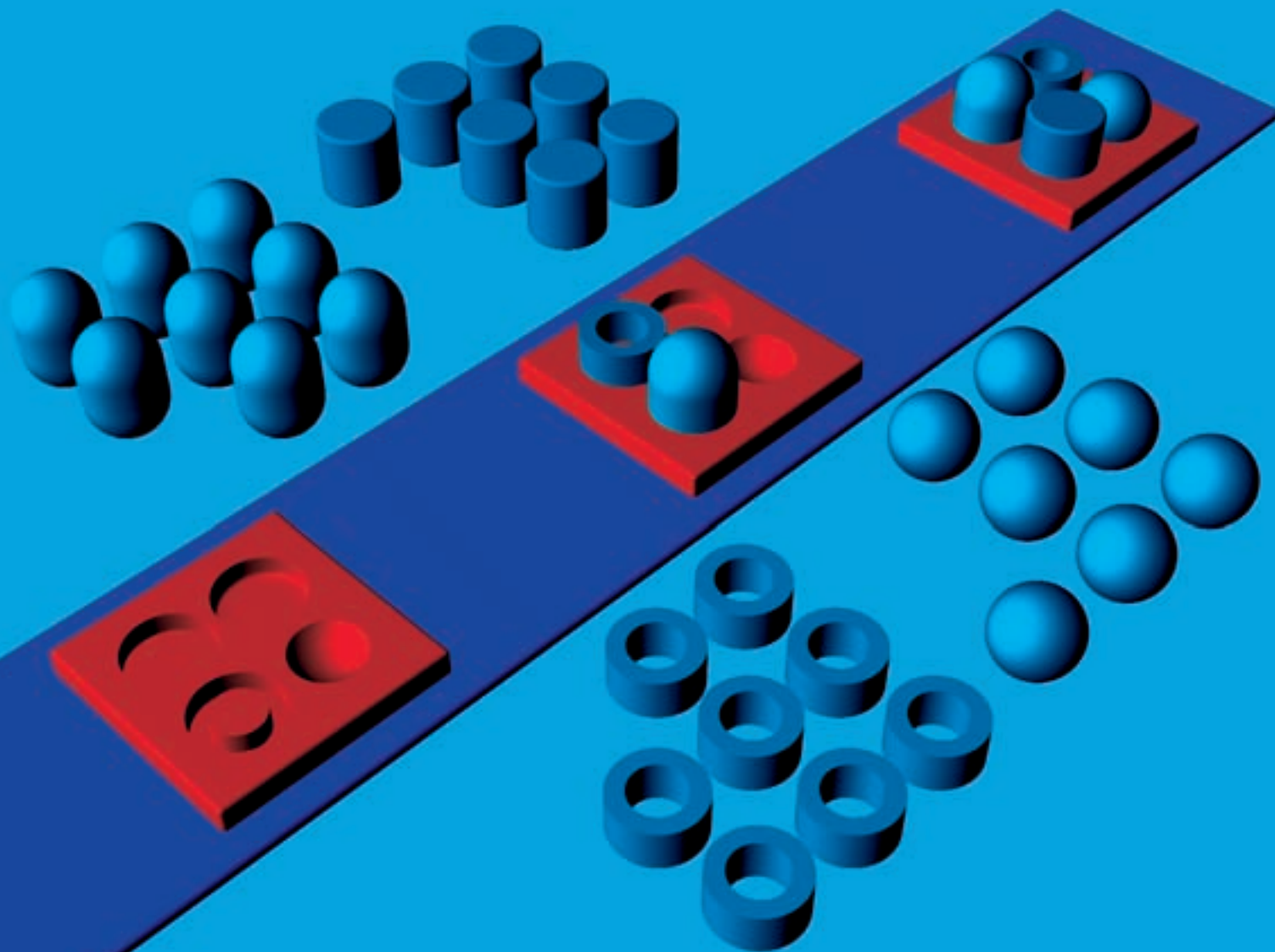
Von der Software-Entwicklung zur Software-Produktion

Callcenter++

Wie eine einfache CTI-Lösung zu einer integrierten Callcenter-Plattform wurde

FRÜHLING 2004, NR. 6

SOFTWARE-PRODUKTION





Liebe Leserin, lieber Leser

Der Spardruck macht sich seit einiger Zeit auch im Software-Bereich bemerkbar. Es heisst, gleich hohe Qualität für immer weniger Geld zu produzieren. In diesem Zusammenhang taucht das Thema «industrielle Software-Produktion» auf, und damit immer häufiger auch «Outsourcing in Billiglohnländer». Dabei wird leicht vergessen, dass Ansätze zur industriellen Fertigung von Software schon länger bestehen. Neue Technologien erlauben eine stärkere Arbeitsteilung in der Entwicklung, und die modulare Bauweise moderner Systeme verkleinert den Entwicklungsaufwand, da vorgefertigte Teile mehrfach eingesetzt werden können.

Den dadurch erreichten Einsparungen beim Programmieren stehen allerdings Auswirkungen in anderen Bereichen gegenüber. Modulare Systeme machen Konfiguration, Deployment und Support anspruchsvoller, und die ausgeprägtere Arbeitsteilung bringt zusätzliche organisatorische Probleme mit sich. So ist das Erarbeiten der Spezifikation immer noch der kostenintensivste Schritt des Entwicklungsprozesses. Dies wird auch in Zukunft so bleiben oder sich eher noch verstärken, denn ohne eine sorgfältige und detaillierte Beschreibung der Anforderungen kann ein Outsourcing-Partner seinen Beitrag nur ungenau oder gar nicht realisieren.

Das Programmieren selbst bildet zudem nur einen kleinen Teil des Software-Entwicklungsprozesses und in noch grösserem Masse des gesamten Software-Lebenszyklus. Da das Sparpotenzial in diesem Bereich schnell ausgeschöpft ist, müssen Optimierungen an einem anderen Ort gesucht werden. Die Erfahrung zeigt, dass die modernen Formen der Software-Produktion nicht nur an den technischen Schnittstellen neue Schwierigkeiten und Herausforderungen mit sich bringen, sondern auch an den menschlichen. Die Vorteile der industriellen Software-Fertigung können sich bei schlecht organisierten Abläufen leicht in ihr Gegenteil verkehren.

Arbeitsteilung und Kooperation müssen darum durch Systeme unterstützt werden, die es ermöglichen, die im gesamten Software-Lebenszyklus entstehenden Informationen zu sammeln und allen Beteiligten zugänglich zu machen.

In dieser Notitia stellen wir Ihnen unsere Ideen dazu vor, wie die Software-Entwicklung gestaltet werden muss, damit sie mit den neuen Bedingungen Schritt halten kann.

Stefan Arn

CEO AdNovum Informatik AG

Information im Griff

GRUNDLEGENDE UND NOCH NICHT ABGESCHLOSSENE VERÄNDERUNGEN, DIE ZUNEHMEND AUCH IM SCHWEIZER SOFTWARE-MARKT SPÜRBAR SIND, MACHEN DIE FÄHIGKEIT, QUALITATIV HOCHSTEHENDE LÖSUNGEN GÜNSTIGER ZU PRODUZIEREN, ZUM IMMER ENTSCHEIDENDEN ERFOLGSFAKTOR. EINE BEDEUTENDE ROLLE SPIELT DABEI DAS RICHTIGE NUTZEN VON RELEVANTEN INFORMATIONEN.

VON TOBIAS MURER

Damit ein Unternehmen seine Software-Produktion optimieren kann, muss es nachvollziehen können, in welchem Bereich aus welchem Grund Kosten entstehen. Dabei ist entscheidend, dass der gesamte Software-Lebenszyklus von der Idee bis zur Entsorgung betrachtet wird und sowohl Qualität wie auch Kosten über alle Phasen hinweg untersucht werden.

Viele Informationen, die im Laufe eines Software-Lebenszyklus anfallen, werden heute noch zu wenig genutzt. Dies kann verheerende Folgen haben, da Software-Produktion ohne den Zugriff auf kritische Informationen ähnliche Risiken birgt wie Fliegen im Blindflug ohne die Hilfe von Instrumenten. Software-Produzenten sollten sich deshalb vermehrt an anderen Disziplinen orientieren, in denen eine qualitäts- und kostenoptimierte Produktion ohne Informationssystem undenkbar wäre,

denn systematisches Erfassen, Vernetzen sowie kontinuierliches Auswerten und Nutzen von Informationen über den ganzen Software-Lebenszyklus hinweg verschaffen entscheidende Vorteile.

DIE INDIVIDUELLE KOMPETENZ UND MOTIVATION DES SOFTWARE ENGINEER IST UND BLEIBT DER ENTSCHEIDENDE FAKTOR BEI DER SOFTWARE-PRODUKTION.

Steigender Informationsbedarf

Die folgenden Trends zeigen, dass die Verfügbarkeit von relevanter und vernetzter Information während des ganzen Software-Lebenszyklus an Bedeutung gewinnt.

Technologie: Moderne Programmiersprachen, Anwendungsplattformen, Komponententechnologien, offene Standards und

Frameworks bilden heute die Grundlage für Software-Systeme. Diese Systeme bestehen aus einem losen, sogar noch zur Laufzeit erweiterbaren Verbund von Komponenten, die auf unterschiedlichen Technologien basieren und sich in ihrer Qualität markant unterscheiden können. Dies erschwert die Realisierung eines homogenen Gesamtsystems. Software als Komponentenverbund zu produzieren erfordert deshalb umfassende Informationen über die einzelnen Teile, ihre Qualität und ihr Zusammenspiel.

Organisation: Software wird seit längerem, heute aber immer öfter, in unterschiedlichen Kooperationsformen über Organisationsgrenzen hinweg produziert, beispielsweise wenn Open-Source-Komponenten eingesetzt werden. Da sich lokal nutzbares Kontext-

wissen nur schwer über Organisationsgrenzen austauschen lässt, entsteht ein Informationsverlust, mit dem auch innerhalb eines Unternehmens gerechnet werden muss, etwa zwischen Teams, die in verschiedene Phasen des Software-Lebenszyklus involviert sind. Deshalb müssen relevante Kontextinformationen sorgfältig aufbereitet werden.

Entwicklung der Kostenverteilung

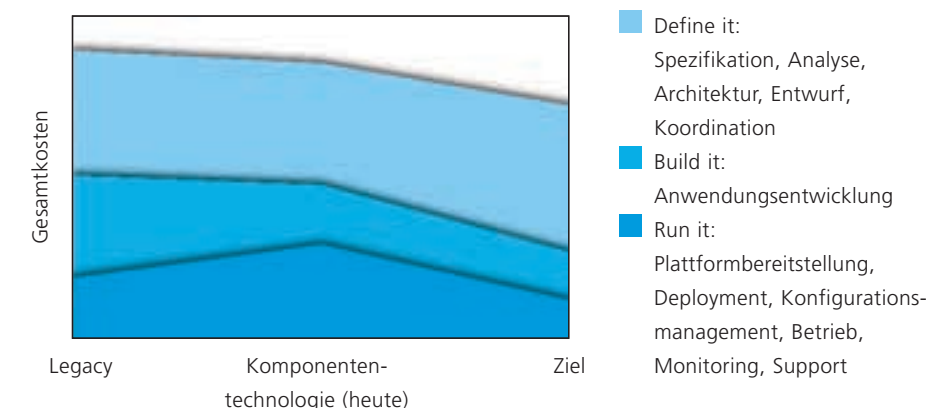
Die in den verschiedenen Phasen der Software-Produktion entstehenden Kosten entwickeln sich unterschiedlich und lassen sich nicht alle gleich gut optimieren. Dies kann mit einer Aufteilung in die folgenden drei Bereiche verdeutlicht werden:

Define it: Anforderungen in eine Spezifikation umzusetzen bleibt eine kritische und aufwändige Aufgabe. Das Potenzial für Kostenoptimierungen ist in diesem Bereich minimal, auch wenn eine höhere Verfügbarkeit relevanter Informationen Verbesserungen ermöglichen kann. Da vom Verfasser einer Spezifikation in Zukunft noch vermehrt eine Kombination von fundiertem Geschäfts- und Technologiewissen sowie eine ausgeprägte kommunikative Kompetenz verlangt werden, muss hier eher mit steigenden Kosten gerechnet werden.

Build it: Unter der Annahme, dass die geforderte Funktionalität ausreichend bekannt ist, lässt sich die reine Anwendungslogik dank neuer Technologien bereits heute mit deutlich reduzierten Kosten entwickeln und warten. Die vermehrte Nutzung des reichen Software-Fundus der Open Source Community verstärkt diesen Trend. Die Kosten lassen sich in diesem Bereich nur beschränkt weiter optimieren. Einsparungen durch den isolierten Einsatz günstiger, externer Ressourcen sind nur mit beträchtlichem

Zusatzaufwand im Define-it-Bereich möglich. Je nach Art des Vorhabens und der gewählten Kooperationsform sind aber durchaus Optimierungen realisierbar, indem Aufgaben an günstigere Ressourcen vergeben werden.

relevanten, vernetzten Informationen dank frühem und kontinuierlichem Feedback ein effizienteres und fehlerfreieres Vorgehen ermöglicht und somit einiges an Kosten vermieden werden kann.



Zeitliche Entwicklung der Kostenverteilung in der Software-Produktion.

Run it: Durch Mehraufwand für das Bereitstellen der Anwendungsplattform, die Integration von Plattform und Komponenten und das weit komplexere Konfigurationsmanagement werden Einsparungen bei der Entwicklung leider häufig nahezu kompensiert. Typischerweise fällt zusätzlicher Aufwand vor allem in Phasen an, die direkt auf die Entwicklung folgen (Release, Installation, Betrieb, Wartung usw.). In diesem Bereich besteht durchaus Potenzial für Kostenoptimierungen, da die stärkere Nutzung von

Blindflug mit limitierten Instrumenten

Im Unterschied zum Piloten stehen dem Software Engineer für seine Aufgabe vermeintlich unzählige Lösungswege zur Verfügung. Seine individuelle Kompetenz und Motivation ist und bleibt daher der entscheidende Erfolgsfaktor bei der Software-Produktion.

Eine wichtige Rolle spielen aber auch Werkzeuge, die dem Entwickler erlauben, sich auf die kreative Arbeit zu konzentrieren,

indem sie automatisierbare Schritte unterstützen. Von zentraler Bedeutung ist dabei ein Informationssystem, das – analog zu den Instrumenten im Cockpit eines Piloten – den Ingenieur kontinuierlich mit qualitativen Aussagen zur Software-Produktion versorgt.

Da die verfügbaren Werkzeuge meist auf spezifische Aufgaben ausgerichtet sind (beispielsweise die Entwicklungsumgebung für das Kodieren), skalieren sie schlecht über Grenzen im Lebenszyklus hinweg. Sie bieten nur einen limitierten Zugriff auf die Gesamtheit und Vernetzung der Informationen und können zum Beispiel folgende Fragen nicht beantworten: Welche Kundeninstallationen sind betroffen, wenn die Schnittstelle einer bestimmten Komponentenversion verändert wird? Zu welcher Komponentenversion gehört ein bestimmter Bugreport? (Weitere Fragen am Schluss des Artikels.)

Application Web – Informationssystem für die Software-Produktion

Um solche Fragen beantworten zu können, wird von der AdNovum das «Application Web» als Informationssystem für die Software-Produktion entwickelt und eingesetzt. Es orientiert sich beim Organisieren, Vernetzen und Aufbereiten von Informationen an der Web-Metapher und bietet im Endausbau die folgenden Funktionen:

Struktur: Abfragen und Navigation über das Informationsnetzwerk des gesamten Software-Lebenszyklus hinweg, auch in dem Zyklus entgegengesetzter Richtung, verfügbar in jeder Zyklusphase.

Tobias Murer

Tobias Murer treibt in der AdNovum den Bereich Java Engineering voran. Den ETH-Informatiker mit zusätzlichem Wirtschaftsabschluss interessiert das Zusammenspiel von technischen und organisatorischen Aspekten bei der Produktion von Software. Zudem gilt er in der AdNovum als zuverlässiger Coach für die kreative Verwendung von Technologie und Business Buzzwords.



Projektion: Nach dedizierten Kriterien extrahierter und aufbereiteter Überblick über Informationen, zum Beispiel ein Projektportal mit allen projektrelevanten Informationen oder ein Report über die Resultate sämtlicher Nightly Builds.

Kennzahlen: Quantitative Aussagen zur Software-Produktion.

Regeln, Metriken: Qualitative Aussagen zur Software-Produktion, automatisierte Überprüfung der Umsetzung von Konventionen und Regeln.

Simulation: Aufzeigen der Konsequenzen von Veränderungen.

Frühwarnsystem: Frühes und kontinuierliches Feedback dank stetiger Informationsauswertung.

Wissensbasis: Aufbereiten von Wissen nach geeigneten Kriterien. Bei formal erfasstem Wissen ist eine Plausibilisierung der Software-Produktion in Bezug auf das Wissen möglich.

Möglichkeiten und Grenzen der Informationsnutzung

Mit einem geeigneten Informationssystem lassen sich bei der Software-Produktion vor allem im Run-it-Bereich Kosten reduzieren, da Probleme früh erkannt und vermieden werden können, und zwar in erster Linie solche, die als ärgerlich empfunden werden und dadurch die entscheidende Motivation der Beteiligten beeinflussen.

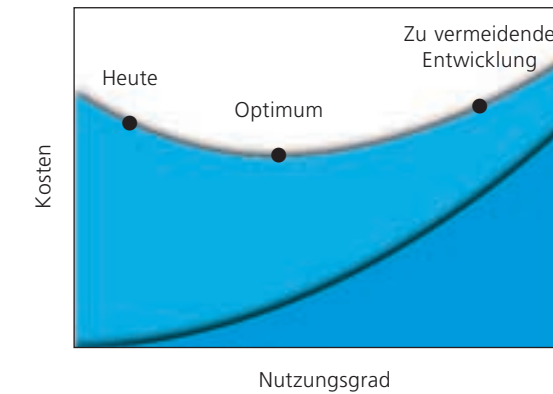
Die Vorteile des Informationssystems zeigen sich vor allem bei aufwändigen Kontextwechseln, etwa bei personellen Änderungen im Team, beim Überschreiten von Organisationsgrenzen, aber auch bei zeitlichen Grenzen, wenn Software nach einem Unterbruch weiterentwickelt wird. Die Möglichkeiten des Informationssystems sind jedoch beschränkt, da das bedeutende individuelle Kontextwissen der Beteiligten nicht aufbereitet werden kann. Anzustreben ist deshalb, auch einen Teil dieses Kontextwissens zu dokumentieren und so für andere nutzbar zu machen.

Zudem stellt sich die Frage, ob die mit Hilfe des Informationssystems erzielten Einsparungen nicht durch den Aufwand für die Aufbereitung der Daten kompensiert werden. Der geeignete Nutzungsgrad ist dann erreicht, wenn die Gesamtkosten von Run-it-Bereich und Informationsaufbereitung minimal sind. Da heute typischerweise zu wenig für die Informationsnutzung aufgewendet wird, fallen im Run-it-Bereich hohe Kosten an.

Ein Arbeiten am Optimum mit fokussiertem Aufwand ist möglich, ohne dass mit zu hohem Engagement der Bereich erreicht wird, in dem die Gesamtkosten wieder ansteigen.

Für das richtige Verhältnis muss das Augenmerk auf relevante, gut nutzbare und vorwiegend automatisch aufbereitbare Informationen gerichtet werden. Viele Daten fallen bereits automatisiert an, sie müssen für die Nutzung nur noch vernetzt und ausgewertet werden. Weniger relevante Informationen hingegen sollten nicht unnötig gesammelt werden. Der Wert von Richtlinien und Konventionen etwa, die nicht automatisiert überprüfbar sind, ist eher klein.

Eine optimale Informationsnutzung bildet die Basis für den Software-Prozess, der für die Produktion in hoher Qualität zu niedrigen Kosten eine zentrale Rolle spielt. Dabei muss unbedingt darauf geachtet werden, dass das geeignete Informationssystem das Fundament



Optimale Informationsnutzung.

für den Software-Prozess bildet und nicht umgekehrt.

Erste Erfahrungen mit dem Einsatz des «Application Web» sind positiv und motivierend. Interessant ist vor allem, dass das kontinuierliche qualitative Feedback den

■ Kosten für Run it:
Plattformbereitstellung,
Deployment, Konfigurations-
management, Betrieb,
Monitoring, Support

■ Kosten für Informations-
aufbereitung und -nutzung

Beteiligten das nicht zu unterschätzende Vertrauen gibt, dass sie sich auf dem Erfolg versprechenden Weg befinden, von dem man ja gerade bei der Software-Produktion sehr rasch und für lange Zeit unbemerkt abkommen kann. ■

Roadmap der Anwendungslandschaft

- Welche Plattform- und Anwendungskomponenten existieren jetzt und in einem Jahr?
- Welche Weiterentwicklungen von Komponenten sind geplant?
- Welche Komponente ersetzt wann welche andere?
- Wann wird eine Komponente nicht mehr verwendet (End of Life)?
- Wie wirkt es sich auf andere Komponenten aus, falls eine Komponente länger als geplant verwendet werden muss oder sich ihre Verfügbarkeit zeitlich verschiebt?

Übersicht, Kontrolle

- Wie viele Komponenten werden in wie vielen Versionen in welchen Projekten verwendet?
- Wie viele und welche Tests gibt es, und wie hoch ist der Grad der Testabdeckung in allen Projekten?
- Welche Probleme wurden durch Nightly Builds über alle Projekte hinweg entdeckt, sind alle Unit- und Integrationstests erfolgreich?
- Welche Bugs mit welcher Priorität existieren über alle Projekte hinweg?

Entwicklung

- Welche direkten oder indirekten Abhängigkeiten existieren zwischen verschiedenen Komponenten? Gibt es Konflikte?
- Welche Komponenten sind direkt oder indirekt betroffen, falls von einer Komponente eine neue Version verfügbar wird?
- Welche Komponenten, die gemäss Schnittstellen zueinander passen, funktionieren trotzdem nicht miteinander?
- Entsprechen Code-Qualität, Architektureigenschaften, Projektstruktur und Abhängigkeiten den vorgegebenen Regeln?

Release, Installation, Support

- Welche Versionen, Releases und Installationen gibt es von einer bestimmten Komponente?
- Welche Versionen sind bei welchen Kunden installiert?
- Welcher Release ersetzt bei einem Kunden welchen vorgängigen Release?
- Welche Komponenten, Releases und Installationen sind von einem dringenden OpenSSL Patch betroffen?
- Welche Installationen sind betroffen, wenn die Schnittstelle einer bestimmten Komponentenversion geändert wird?
- Zu welcher Komponentenversion gehört ein bestimmter Bugreport?

Software-Entwicklung im Umbruch

TOM SPRENGER ERKLÄRT IM GESPRÄCH MIT NOTITIA DEN WANDEL VON DER SOFTWARE-ENTWICKLUNG ZUR SOFTWARE-PRODUKTION.

INTERVIEW: BARBARA STAMMLER UND DORINA MAYRHOFER

NOTITIA: Seit die Internetblase geplatzt ist, bewegt sich die IT-Industrie in unruhigen Gewässern. Die Krise scheint hartnäckiger als erwartet. Gibt es Gründe dafür?

Tom Sprenger: Bevor man nach Erklärungen sucht, sollte man sich fragen, ob die aktuelle Situation nicht einfach Teil einer logischen Entwicklung eines noch jungen Technologiebereichs ist. Die Software-Branche hatte ihre kommerziell besten Jahre, als man das Potenzial im Bereich der Vernetzung von Informationen über das Internet entdeckte. Jede Lösung war eine teure Spezialanfertigung. Die Menge der Probleme ist heute noch dieselbe, die Lösungen sind mittlerweile weitgehend bekannt, und Software ist deshalb in vielen Bereichen zur Commodity geworden.

Meinen Sie damit, dass man mit Software kein Geld mehr verdienen kann?

Das kann man so nicht sagen. Es wird immer einen anspruchsvollen Markt geben (etwa Security), in dem man mit neuen Ideen und massgeschneiderten Entwicklungen existieren kann. Das dort verlangte, sehr spezifische Know-how rechtfertigt auch den Preis solcher Lösungen. In anderen Marktsegmenten (zum Beispiel Applikationen) muss allerdings ein Umdenken stattfinden. Das neue Zauberwort heisst Effizienz. Analysiert man andere Technologiebereiche, sieht man, dass ab einem gewissen Entwicklungsstand die Effizienz eine vitale Rolle zu spielen beginnt. Genau dort befindet sich die IT-Industrie heute.

War Effizienz nicht bereits schon in der schnelllebigen Internetzeit ein entscheidender Erfolgsfaktor?

In den Jahren der Hochkonjunktur bestand ein konstanter Nachfrageüberhang im Software-Markt. Sparen war kein Thema, und Ineffizienz in der Software-Entwicklung wurde leicht verziehen. Es ist eigentlich fast schon ironisch, wie eine Industrie, die sich in einem so schnelllebigen Umfeld so intensiv mit Leistungsoptimierung befasst, selbst immer noch auf so ineffizienten Prozessen basieren kann. Heute sehen sich IT-Unternehmen durch die stark unter Druck geratenen Budgets gezwungen, qualitativ hochstehende Software nicht nur schnell, sondern auch kostengünstig zu entwickeln. Wir erleben zurzeit eine Art industrielle Revolution in der Software-Industrie. Darum ist es auch treffender, von Software-Produktion als von Software-Entwicklung zu sprechen.

Was muss man sich unter dem Begriff «Software-Produktion» genau vorstellen?

Es geht darum, in bestimmter Zeit und Qualität aus vorgefertigten Basiskomponenten kundenspezifische Software-Lösungen herstellen zu können – ähnlich wie heute Autos gebaut werden. Nun ist komponentenorientierte Software-Entwicklung keine neue Idee, neu ist aber, dass dieses Denken auf das Problem als Ganzes angewandt und das Zusammenfügen der Komponenten in einen kontrollierenden Prozess eingebettet wird. Software-Produktion beschränkt sich ausserdem nicht nur auf die Entwicklung, sondern umfasst den ganzen Software-Lebenszyklus bis hin zu Deployment, Betrieb und Wartung. Dadurch erst wird der Kreislauf geschlossen und wichtige Informationen aus dem Lebenszyklus können zurück in die Entwicklung fliessen.

Erfolgreich produzieren kann man nur, wenn man den Prozess, die Komponenten und die eingesetzten Werkzeuge beherrscht. Zum anderen steht Kontrolle auch für Verifikation. Da Mängel einen weitreichenden Einfluss auf die Effizienz haben, muss das Fehlerrisiko so weit wie möglich minimiert werden. Fehler,



entsprechende Prozesse schon vorhanden sind. Diese Prozesse behalten weitgehend ihre Gültigkeit, werden aber mit dem Schritt zur Software-Produktion um Produktionsaspekte wie beispielsweise Kontroll- und Automatisationsmechanismen angereichert. Allerdings sollte man es auch wieder nicht übertreiben. Prozessdefinitionen können schnell eine Eigendynamik entwickeln und sich entgegen den Bestrebungen negativ auf die Effizienz auswirken. Das primäre Prozessziel muss das termingerechte Herstellen von Software sein. In den meisten Fällen genügt es bereits, wenn dafür ein Prozess informell spezifiziert wird. Viel wichtiger ist, dass alle Beteiligten eine einheitliche Sicht teilen und der Prozess auch gelebt wird.

Welche direkten Folgen hat der Paradigmenwechsel für die Anwendungs- und Entwicklungsplattform?

In verschiedenen Bereichen muss die Freiheit bewusst eingeschränkt werden. Für den Erfolg ist entscheidend, dass man sich an der Basis auf eine definierte Menge einfacher Software-Komponenten und Technologien mit bekannten Eigenschaften abstützt. Insbesondere müssen die Abhängigkeiten zwischen den

Entwicklungsaufwand reduziert, dafür aber die Anforderungen an die Konfiguration erhöht. Solche Lösungen können zudem zur Laufzeit des Systems komplexe dynamische Abhängigkeiten aufweisen, die es wiederum zu beherrschen gilt.

Kontrolle hängt oft von Wissen ab. Gilt diese Beziehung auch im aktuellen Kontext?

Hier liegt ein grosses Wertschöpfungspotenzial der Software-Produktion. Heute bleibt ein grosser Teil der Informationen, auf denen das benötigte Wissen basiert, aus verschiedenen Gründen ungenutzt. So werden diese Informationen meist explizit erfasst (zum Beispiel als Bestandteil der Dokumentation) und sind somit potenziell ungenau oder veraltet und in einem für die Produktion ungeeigneten Format gespeichert. Zudem geht entscheidende Information an Organisationsgrenzen verloren. Ein Teil der Anstrengungen muss dahin gehen, Strukturen und Werkzeuge zu schaffen, die es ermöglichen, die implizit als Teil des Produktionsprozesses entstehenden Informationen zu erkennen und über den ganzen Software-Lebenszyklus und alle Organisationsgrenzen hinweg jederzeit verfügbar zu machen.

Ist die Etablierung eines eigentlichen Produktionsprozesses nicht auch eine Voraussetzung für die erfolgreiche Software-Produktion?

Das ist prinzipiell korrekt. Obige Ausführungen gehen davon aus, dass die Software-Produktion auf einer bereits etablierten Entwicklungsinfrastruktur aufbaut und deshalb



die trotzdem auftreten, müssen möglichst früh im Produktionsprozess erkannt werden. Ausserdem setzt eine effiziente Produktion einen hohen Automatisierungsgrad voraus. Insbesondere betrifft dies das Generieren von Code aus höherwertigen Repräsentationen sowie das generelle Automatisieren repetitiver Tasks, etwa wenn Qualitätsmetriken periodisch berechnet werden.

Begünstigen moderne Technologien wie J2EE die Produktion von Software?

J2EE definiert eine standardisierte Plattform, die den initialen Entwicklungsaufwand markant senkt und eine einheitliche Sicht auf die Problemstellung und die möglichen Lösungen unterstützt. Daneben sieht die modulare J2EE-Architektur wieder verwendbare Komponenten in der Form von EJBs vor und erlaubt damit den modularen Aufbau komplexer Systeme aus vorgefertigten Teillösungen. Man darf aber nicht vergessen, dass diese Art von Architekturen zwar den

Welche Voraussetzungen sind für das erfolgreiche Produzieren von Software nötig?

Generell sind es dieselben wie für die Realisierung von Software bisher. Mit dem Schritt zur Software-Produktion kommen aber zwei fundamentale Punkte dazu: Kontrolle und Automatisierung. Dabei hat Kontrolle zwei Bedeutungen. Zum einen ist damit das Kontrollieren beziehungsweise das Beherrschen der Software-Produktion gemeint.



Tom Sprenger

Tom Sprenger, diplomierte Informatik-Ingenieur ETH, arbeitet seit 2000 in der AdNovum und ist CEO der Niederlassung in San Mateo, Kalifornien. Er befasst sich dort als so genannter «Technologie-Proxy mit Prinzipien» mit den Themen managbare System-Architekturen und Quality Engineering. Ausserdem stehen Fragen zur Strategie und zum Einsatz von Java-Technologien bei ihm ganz zuoberst auf dem Speisezetteln. Seine Kollegen nennen ihn «Dr. Dynamite».

Komponenten im Detail bekannt sein. Neue, eigene Komponenten sollen nur dann geschrieben werden, wenn noch keine mit der gewünschten Funktionalität oder in der verlangten Qualität verfügbar sind. Durch die systematische Organisation der Komponenten in einem normalisierten Namensraum können sie einheitlich integriert werden. Darüber hinaus lassen sich die Abläufe automatisieren. Ebenso wichtig ist die Wahl des richtigen Toolsets. Tools sollten nie die inhärente Komplexität verbergen, sondern ein effizientes

Basiskomponenten bestimmt, aus denen es aufgebaut ist. So impliziert der Entscheid, eine neue Komponente in der Produktion einzusetzen, stets einen initialen Testmehraufwand. Testwerkzeuge gibt es heute genügend, man muss aber genau wissen, wann man was mit welcher Technik testet. In diesem Zusammenhang wird immer von den durch Fehler verursachten Kosten gesprochen. Ich glaube aber, dass man dabei fälschlicherweise nur an die untere Grenze des Testaufwands denkt. Man darf nicht vergessen, wie teuer

ten. Es ist immer noch so, dass die eigentliche Entwicklung nur einen kleinen Teil des Aufwands in der Software-Produktion ausmacht. Den teuersten Teil bildet das Erstellen einer klaren Spezifikation. Outsourcing ohne Kontrolle und entsprechende Strukturen kann leicht zum Nachteil werden. Damit es erfolgreich ist, müssen bestimmte Voraussetzungen erfüllt sein, beispielsweise präzise Spezifikationen, Beherrschen von Abhängigkeiten, eine gemeinsame Sicht über Organisationsgrenzen hinweg und ein effizientes Testen zur Verifikation der im Outsourcing erstellten Komponenten.

« WIR ERLEBEN ZURZEIT EINE ART INDUSTRIELLE REVOLUTION IN DER SOFTWARE-INDUSTRIE. »

Produzieren ermöglichen. Zudem sollten sie zum vorhandenen Prozess passen und ihn nicht definieren. Wenn ein Toolset nicht fließend in die bestehenden Prozesse integriert werden kann, sondern einen Paradigmabruch verursacht, ist es wahrscheinlich nicht das richtige. Hochintegrierende Tools führen meist zu einem unnötigen Vendor Lock-in.

In der Praxis hat es sich bewährt, die Umgebungen in erster Linie aus einfachen, aber zuverlässigen Tools aufzubauen, die leicht in automatisierte Abläufe integriert werden können. Ergänzend sollten für bestimmte Aufgaben auch komplexere, grafische Tools zum Einsatz kommen, deren Stärken vor allem im Bereich der explorativen Analysen im Produktionsprozess liegen (Source Code Browsen, Dependency- und Profiler-Analysen usw.).

In der Definition von Software-Produktion wurde das Produzieren in bestimmter Qualität als Ziel genannt. Wie wird dies konkret umgesetzt?

Qualitäts-Engineering durch Testen ist ein absolut fundamentaler Aspekt der Software-Produktion. Die Qualität eines Gesamtsystems wird substantiell durch die Qualität der

die Entwicklung und insbesondere die Pflege von Testreihen sein können. Deshalb sehe ich beim Testaufwand ganz klar auch eine obere Grenze, die in Diskussionen oft unterschlagen wird. Wichtig ist es, richtig zu priorisieren, damit der Testaufwand nicht den Nutzen übersteigt.

Kann dem Margendruck nicht einfach durch Outsourcen der Software-Entwicklung in Billiglohnländer begegnet werden?

Natürlich können mit Outsourcing Kosten zum Beispiel im Bereich der Entwicklung eingespart werden. Allerdings muss man solche Anstrengungen im richtigen Kontext betrach-

Der Schritt in Richtung Software-Produktion hat offensichtlich Konsequenzen. Wird der Paradigmawechsel die heutige IT-Industrie verändern?

Viele der erwähnten Punkte sind eigentlich nicht neu. Bestrebungen in diese Richtung bestehen schon seit über zehn Jahren. Allerdings ist eine konsequente Umsetzung erst unter den heutigen Bedingungen realistisch. Die Veränderungen werden sich aber sicherlich auch auf das Berufsbild auswirken. Früher war der einzelne kluge Kopf entscheidend, heute der effiziente Prozess zusammen mit der richtigen Information, auf der der Prozess aufbauen kann. Die Arbeiten werden sich wie in anderen Industriezweigen stärker spezialisieren. Software-Architekten analysieren die Probleme und tragen die Verantwortung für



das Lösungsdesign, hochqualifizierte Ingenieure mit Spezialwissen übernehmen die eigentliche Entwicklung der einzelnen Software-Komponenten, während Komponenten-Assembler für einen grossen Teil von Integration und Konfiguration nach einem vorgegebenen Anwendungsschema zuständig sind.

Es wird sich somit einiges ändern in der IT-Landschaft. ■

Callcenter++

EINE URSPRÜNGLICH AUF KUNDENGESPRÄCHE AUSGERICHTETE PLATTFORM FÜR EIN PAAR DUTZEND AGENTEN DES CUSTOMER SERVICE CENTER DER UBS AG WURDE IN DEN LETZTEN JAHREN ZU EINER EIGENTLICHEN CALLCENTER-MIDDLEWARE MIT ZWEI EIGENSTÄNDIGEN APPLIKATIONEN AUSGEBAUT, FÜR DIE MEHR ALS 600 MITARBEITER AUTORISIERT SIND.

VON RUEDI BECKER

Im Jahr 2000 erhielt die AdNovum den Auftrag, in Zusammenarbeit mit der UBS eine bestehende Rich-Client-basierte CTI-Lösung (Computer Telephony Integration) für Kunden von UBS e-banking über Telefon als eine browserbasierte Thin-Client-Applikation völlig neu zu schreiben. In dieses Portal – heute CSC-Desk – mussten diverse Informationen und Applikationen möglichst einfach integriert und den Callcenter-Agenten übersichtlich zur Verfügung gestellt werden. Deshalb wurde ursprünglich eine Integrationsplattform mit einer Secure-Call-Transfer-Einbindung im Kern und einfachsten Funktionen realisiert.

Während die integrierten Interactive-Voice-Response-Lösungen (IVR) in der UBS erweitert wurden, entwickelte die AdNovum die Basisbausteine, welche die vollständige Einbindung der Sprachcomputer für Secure Call Transfers und den Datenzugriff auf das E-Banking-System ermöglichten.

Der Secure Call Transfer ist das eigentliche Herzstück der Applikation. Hat ein Kunde am Telefon seine Vertragsnummer und sein Passwort eingegeben, kann er über die Tastatur oder mit gesprochenen Anweisungen eine Vielzahl von Abfragen und Transaktionen ausführen, für die eine schwache Authentisierung genügt. Für Börsenaufträge, Vertragsmutationen und Ähnliches muss der Kunde mit einem Agenten verbunden werden. Damit er dabei seine Identifikationsmerkmale nicht nochmals eingeben muss, wurden CSC-Desk und die beiden Sprachcomputer in eine Single-Signon-Domäne integriert. Wechselt nun ein bereits angemeldeter Benutzer vom Sprachcomputer zu einem Callcenter-Mitarbeiter, kann dieser die Session des Anrufers übernehmen.

Für die Auswahl des Agenten ist das in der PBX-Telefonzentrale konfigurierte Skill-based Routing verantwortlich. Es stellt sicher, dass der Kunde jederzeit an einen freien Mitarbeiter mit den nötigen Produkt- und Sprachkenntnissen vermittelt wird. Sobald der

Callcenter-Agent den Anruf übernimmt, erscheint auf seinem Bildschirm eine Übersicht mit allen für den Gesprächsbeginn nötigen Kundeninformationen.

Zur Single-Signon-Domäne gehört auch die transparent in CSC-Desk eingebundene E-Banking-Lösung. Dank Webtechnologie konnten alle bereits online angebotenen

DANK WEBTECHNOLOGIE KÖNNEN BEREITS ONLINE ANGEBOtene FUNKTIONEN WIEDER VERWENDET WERDEN.

bankfachlichen Funktionen aus UBS e-banking wieder verwendet werden. Möchte ein Anrufer etwa Zahlungen ausführen, wird seine Session an den Agenten übergeben, der dann im Namen des Kunden Bankgeschäfte erledigen kann. Falls für gewisse Transaktionen eine starke Authentisierung nötig ist, muss der Anrufer dem Agenten ein weiteres Authentisierungsmerkmal mitteilen.

Im Lauf der Zeit kamen zu den Funktionen aus UBS e-banking bankfachliche Erweiterungen hinzu, zu deren Nutzung nur der Mitarbeiter der Bank berechtigt ist. Mainframe-basierte Transaktionen wie etwa die Bewirtschaftung der Authentisierungsmittel sowie Karten- und Vertragssperrungen wurden in CSC-Desk integriert.

Dank dem konsequenten Single-Signon-Domaining und der nahtlosen Integration aller Komponenten nehmen die Benutzer von CSC-Desk gar nicht wahr, dass die angebotenen Funktionen eigentlich aus unterschiedlichen Anwendungen stammen.

Backoffice-Integration und Neuaufteilung

Die im gleichen Gebäude arbeitenden Backoffice-Mitarbeiter schielten wohl schon lange auf die von den Kollegen an der Front so erfolgreich eingesetzte Applikation. Da der

Anfang der Vertragsverarbeitung bereits in CSC-Desk integriert worden war, lag eine Ausweitung auf Backoffice-Aufgaben nahe. Damit wuchs die Benutzerzahl der Callcenter-Applikation beträchtlich, die Hälfte der Anwender hatte allerdings gar keinen telefonischen Kundenkontakt. Im Backoffice-Bereich werden mit CSC-Desk eine Vielzahl verschiedener Dokumententypen zentral verwaltet und gedruckt und vor allem E-Banking-Verträge auf dem Mainframe erstellt.

Damit war CSC-Desk zu einer Anwendung mit zehn Benutzerrollen für Anwender mit unterschiedlichsten Aufgaben angewachsen. Sie wickeln telefonische Börsenaufträge ab, erledigen Supportanfragen, sperren EC- und Kundenkarten, nehmen am Telefon Vertragsänderungen entgegen und führen im Backoffice ohne direkten Kundenkontakt Mutationen aus.

Eine so heterogene Benutzergruppe hat natürlich unterschiedliche Ansprüche an die Verfügbarkeit der Anwendung. Backoffice-Mitarbeiter brauchen weder ein Telefonie-Login noch Zugriff auf UBS e-banking und

Ruedi Becker

Ruedi Becker, diplomierter Elektro-Ingenieur HTL, ist technischer Leiter des Callcenter-Projekts CSC-Desk, seit er in der AdNovum ist. Kleine Abstecker haben ihm Einblick in andere Anwendungen verschafft und, anlässlich eines Aufenthalts bei der AdNovum in Kalifornien, auch in die Abgründe der Middleware-Entwicklung.

Damit sein Leben nicht zu telefonlastig wird, verzichtet er auf ein Handy. Fit für Neuanforderungen macht er sich in seiner Heimat, dem schönen Glarnerland, bei seiner Familie und seinem liebsten Hobby, als Kornettist der Harmoniemusik Glarus.



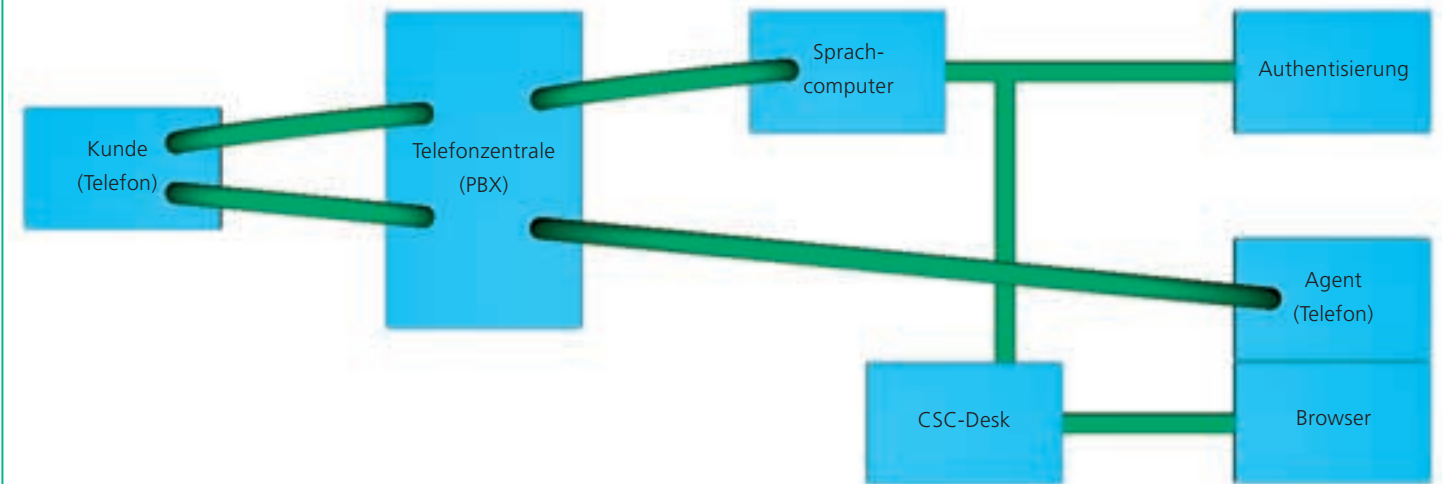
Ruedi Becker ist technischer Projektleiter bei der AdNovum.

Secure Call Transfer

Der Secure Call Transfer ist zweifellos das eigentliche «pièce de résistance» des CSC-Desk. Er ermöglicht es, den Bankkunden im Sinne einer Multi-Channel-Strategie die Telefonie neben Schalter und Internet als

gleichwertigen Kanal zur Verfügung zu stellen. Hat sich ein Kunde über das Telefon mit seinem Namen und Passwort angemeldet, kann der so hergestellte Security-Kontext mit Hilfe der Nevis-Web-Infrastruktur zum Callcenter-Agenten transferiert werden. Da die Grunddaten zum entsprechenden Vertrag

automatisch vom Mainframe geladen werden, erhält der Mitarbeiter des Customer Service Center damit nicht nur einen Überblick über alle nötigen Informationen, sondern er kann auch – solange die Verbindung besteht – im Namen des Kunden E-Banking-Funktionen ausführen.



arbeiten zur üblichen Bürozeit, First-Level-Agenten im Callcenter hingegen müssen an sieben Tagen in der Woche rund um die Uhr auf die Anwendung zugreifen können. Der hohe Integrationsgrad von CSC-Desk zieht eine gewisse Anfälligkeit nach sich. Fällt eine Applikation aus, kann CSC-Desk jedoch dank dem Konzept der «loosely coupled modules» mit eingeschränktem Funktionsumfang weiter benutzt werden.

Damit flexibel auf die unterschiedlichen Benutzeransprüche und Neuanforderungen aus Telefonie und Backoffice eingegangen werden kann, wurde Anfang 2003 entschieden, die Lösung in zwei eigenständige

Teilnehmer mehr: CSC-Backoffice, das als Stand-Alone-Applikation konzipiert wurde. Damit die arbeitsteilige Vertragsadministration von Callcenter und Backoffice weiterhin möglich ist, greifen beide Anwendungen auf dieselbe Datenbank zu. Da CSC-Desk nach wie vor die Funktion eines Portals wahrnimmt, werden einige Funktionen, wie zum Beispiel die Passwortbestellung, via Single Signon aus CSC-Backoffice bezogen.

Mehr und mehr Benutzer

Praktisch mit der Einführung einer zweiten Serverlinie im November 2003 versechsfachte sich die Anzahl gleichzeitig angemeldeter

Online-Banking-Anwendung der UBS natürlich nicht möglich. Trotzdem haben auch die Berater dieser Kunden CSC-Desk schätzen gelernt, womit die Anwendung in der Schweiz zu einem wichtigen Arbeitsmittel für UBS-Mitarbeiter im Callcenter-Umfeld geworden ist.

Flexibel und anpassbar

Die Vorteile der Architektur und der Realisierung zeigten sich auch, als die AdNovum abklärte, welche Komponenten beim Einsatz einer anderen PBX (Private Branch Exchange) beeinträchtigt wären und ob allenfalls Einschränkungen der Funktionalität in Kauf genommen werden müssten. Die mit wenig Aufwand durchgeführte Untersuchung bestätigte das Konzept der «loosely coupled modules», denn die Verbindung zwischen CSC-Desk und Telefonzentrale beschränkt sich auf ein einziges Interface mit wenigen Events. Ebenfalls zahlt sich aus, dass konsequent nur Industrie-Standards verwendet wurden.

Das leicht anpassbare System und nicht zuletzt die fast mit jedem Release steigenden Benutzerzahlen können als Indiz dafür angesehen werden, dass sich der Einsatz von Webtechnologien bewährt hat. Mit ihnen ist es möglich, bestehende Komponenten ohne grosse Probleme auszutauschen und auch in späteren Releases noch wirklich neue Funktionen einzubauen. ■

DANK DER KONSEQUENTEN VERWENDUNG VON INDUSTRIE-STANDARDS IST CSC-DESK UNABHÄNGIG VON BESTIMMTEN PBX-HERSTELLERN.

Anwendungen aufzuteilen und die im Backoffice benötigten Funktionen mit modernster J2EE-Technologie neu zu realisieren. Auch hier erwies sich der Ansatz der Lean-Integration, wie er im Jahr 2000 gewählt worden war, als richtig, denn so konnte der Backoffice-Teil unter Bewahrung der weiterhin gewünschten Integration von Funktionen für Agenten im Callcenter (Vertragsmutationen) herausgelöst werden. Damit umfasst die CSC-Single-Signon-Domäne seit August 2003 noch einen

Benutzer in CSC-Desk, da aufgrund einer Prozessoptimierung heute auch die Mitarbeiter von zentralisierten Beratungseinheiten CSC-Desk verwenden, um zusätzlich Kunden ohne E-Banking-Vertrag zu betreuen. Waren zuvor jeweils an die fünfzig Agenten gleichzeitig angemeldet, so waren es danach nahezu 300 (ohne Backoffice-Mitarbeiter).

Daneben hat sich das Benutzerschema von Grund auf verändert, denn für Kunden ohne E-Banking-Vertrag ist ein Login in die

Enterprise Grid Computing

UNBERECHENBARE GESCHÄFTSZYKLEN FÜHREN VIELE UNTERNEHMEN AN DIE GRENZE DER BELASTBARKEIT. VERÄNDERTE BEDINGUNGEN ERFORDERN IMMER SCHNELLERE ANPASSUNGEN, WODURCH BESTEHENDE INFORMATIK-SYSTEME OFT AN IHR LIMIT GERATEN. UM AUF ALLES VORBEREITET ZU SEIN, BAUEN IT-ABTEILUNGEN TEURE, REDUNDANTE INFRASTRUKTUREN MIT HOHEN KAPAZITÄTSRESERVEN AUF.

VON MASSIMO CASTELLI, ORACLE SOFTWARE (SCHWEIZ) GMBH

Oracle hat eine Antwort auf diese Herausforderungen: Enterprise Grid Computing. Oracle Database 10g, Oracle Application Server 10g und Oracle Enterprise Manager 10g Grid Control unterstützen Unternehmen dabei, sich veränderten Bedürfnissen in nützlicher Frist optimal anzupassen und gleichzeitig Investitionen in neue Technologien zu schützen. Enterprise Grid Computing reduziert die Kosten für die Datenverwaltung erheblich, vereinfacht das Management von Business-Applikationen und verbessert die Kapazitätsauslastung.

Ressourcen nach Bedarf

Grid Computing schaltet eine grosse Anzahl von Servern und Speichern zu flexibel abrufbaren Ressourcen zusammen, die sich für alle Rechenbedürfnisse eines Unter-

nehmens nutzen lassen. Als grundlegend neue Computerarchitektur erlaubt Grid Computing den schnellen und einfachen Aufbau einer grossen Computerinfrastruktur aus kostengünstigen Standardkomponenten wie Server Blades und Massenspeichern. Durch zentrales Clustering und Lastausgleich bietet Grid Computing höchste Leistung, Skalierbarkeit, Zuverlässigkeit und Sicherheit. Da Oracle Grid Computing plattformunabhängig ist, lässt sich jede Art von System zum Aufbau eines Enterprise Grid auf der Basis der Oracle Infrastruktursoftware 10g einsetzen.

Mit 10g bietet Oracle die nötigen Voraussetzungen für effizientes Grid Computing: Das Herzstück, Oracle Enterprise Manager 10g Grid Control, ist eine auf offenen Standards basierende Software, mit der sich die ganze IT-Infrastruktur von einer Konsole aus zentral verwalten, überwachen und steuern lässt – von der Datenbank über Anwendungen bis hin zu Storage-Systemen. Integrierte Management-Funktionen und die web-

Oracle Corporation

Oracle Corporation mit Sitz in Redwood Shores, Kalifornien, ist der weltweit grösste Anbieter für Unternehmenssoftware. Die vollständig internetfähige Produktlinie umfasst Software für Datenbanken, Applikationsserver, Unternehmensanwendungen sowie Entwicklungs- und Entscheidungstools. Zusätzlich bietet Oracle Beratung, Ausbildung und Support Services an.

In der Schweiz ist Oracle seit 1985 vertreten. Der Hauptsitz ist in Baden-Dättwil, weitere Geschäftsstellen befinden sich in Bottmingen, Bern und Genf.

Mehr zu Oracle: www.oracle.com

Grid herstellen. Die Datenbank kann von den Anforderungen kleiner und mittlerer Unternehmen bis hin zu grossen Applikationen mit mehreren tausend Terabyte skalieren.

Der Oracle Application Server 10g vervollständigt die Grid-Lösung. Dieser J2EE-Server ermöglicht das Erstellen von Rechnerverbänden, wodurch steigenden Anforderungen an Rechenleistung mit der Integration neuer Hardware begegnet werden kann.

Grid Computing im Dienst der Wissenschaft

Dass Grid Computing mehr als nur ein Schlagwort ist, zeigt das Beispiel der europäischen Organisation für Nuklearforschung in

Genf, CERN, die den weltweit leistungsstärksten Teilchenbeschleuniger mit entsprechend hohen und komplexen Ansprüchen an die IT-Infrastruktur entwickelt. Das ehrgeizige Projekt erfordert gewaltige Speicherressourcen für voraussichtlich mehr als zehn Petabytes Daten pro Jahr. Um diesen Ansprüchen an die Datenverarbeitung gerecht zu werden, hat sich das CERN schon früh für Oracle Grid Computing entschieden, da so die Nutzung und Verwaltung sowie das Clustern von Daten vereinfacht werden. ■

MIT FLEXIBEL ABRUFBAREN RESSOURCEN LÄSST SICH AUF VERÄNDERTE BEDINGUNGEN SCHNELL REAGIEREN.

basierte Konsole Database Control machen automatisch Vorschläge zur Kapazitäts-, Verfügbarkeits- und Performance-Steuerung im Grid und sorgen für die Verteilung der benötigten Rechenleistung zwischen den Speichersystemen und den angeschlossenen Rechnern.

Auch bei Oracle Database 10g steht das Zusammenschalten verschiedener Rechner im Vordergrund; so lässt sich aus Oracle-Installationen auf verschiedenen Plattformen wie Windows, Linux oder Unix ein Enterprise

Impressum

Herausgeber:

AdNovum Informatik AG
Corporate Marketing
Röntgenstrasse 22
CH-8005 Zürich
Telefon 01 272 61 11
Telefax 01 272 63 12
E-Mail info@adnovum.ch
www.adnovum.ch

Verantwortlich und Redaktion:

Barbara Stammli
Dorina Mayrhofer

Gestaltung und Realisation:

Rüegg Werbung, Zürich

Fotografie:

Matthias Auer, Zürich