

Thomas Schneider

# **SAP-Performanceoptimierung**

Analyse und Tuning von SAP-Systemen

# 1 Performancemanagement einer mySAP.com-Lösung

Kunden, die Lösungen der SAP einsetzen, verbinden dies mit einer hohen Erwartung an die Zuverlässigkeit und Wartbarkeit dieser Lösungen. Sie rechnen damit, dass die SAP die maßstabsetzenden Standards, die sie mit SAP R/3 erreicht hat, auch für mySAP.com-Lösungen (CRM, SCM, BI etc.) einhalten kann. Dazu bietet die SAP nicht nur ihre bewährte Plattform für ERP- und E-Business-Lösungen mit der performantesten Architektur auf dem Markt an, sondern auch – laut Meinung von Analysten – ein richtungweisendes Servicekonzept.

Architektur und Servicekonzept stellen wir in diesem einleitenden Kapitel vor. Im ersten Abschnitt gehen wir auf die Architektur ein, mit der mySAP.com-Lösungen realisiert werden, und umreißen das systemseitige Optimierungspotenzial. Es wird an dieser Stelle bewusst auf technische Details verzichtet, die nur einem Techniker verständlich sind. Im zweiten Abschnitt stellen wir uns organisatorischen Fragen des Betriebs einer mySAP.com-Lösung, wie die Erstellung eines Überwachungs- und Optimierungskonzeptes mit dem SAP Solution Manager. Dabei spielen zum einen ein Konzept für eine kontinuierliche Überwachung der Verfügbarkeit und Performance des Geschäftsprozesses und zum anderen die Methode des Service Level Managements eine herausragende Rolle.

## Wann sollten Sie dieses Kapitel lesen?

Dieses Kapitel sollten Sie lesen, wenn Sie ein Überwachungs- und Optimierungskonzept für eine mySAP.com-Lösung aufbauen wollen. Wir empfehlen die Lektüre als Überblick grundsätzlich, bevor Sie sich in die weiteren Kapitel des Buches vertiefen.

## 1.1 Die Architektur von mySAP.com

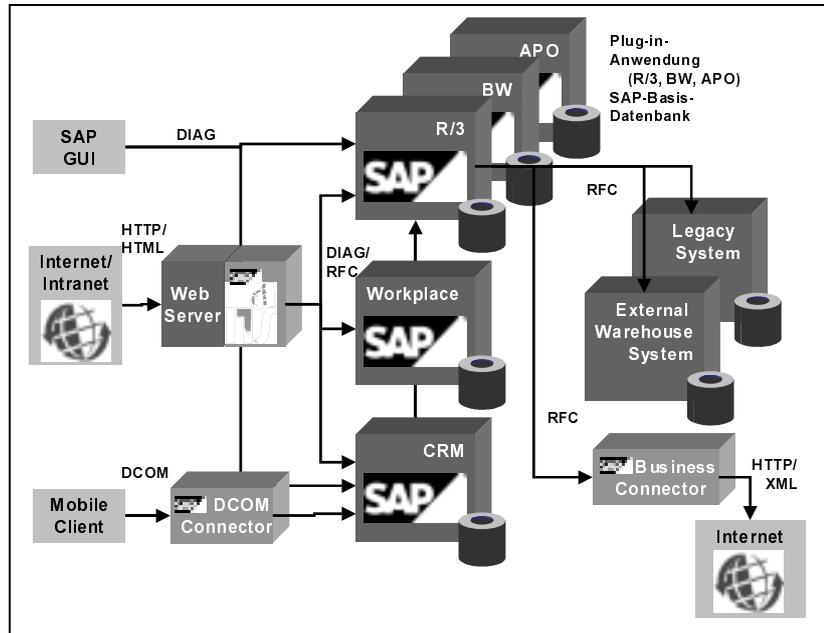
Im Folgenden wird die Architektur von mySAP.com behandelt. Zunächst wird dabei auf die mySAP.com-Lösungen und -Komponenten eingegangen. Anschließend folgt ein Abschnitt über die Client/Server-Architektur.

### 1.1.1 mySAP.com-Lösungen und mySAP.com-Komponenten

In der Vergangenheit baute die SAP-Software auf einer einzigen Technologiekomponente – der *SAP-Basis* als Grundlage von SAP R/3 – auf. Mit *mySAP.com* werden die Geschäftsprozesse eines Unternehmens heute nicht nur in einem einzigen R/3-System abgebildet, sondern in mehreren

Aufbau einer  
mySAP.com-  
Lösungslandschaft

Softwarekomponenten. Abbildung 1.1 zeigt ein Beispiel für Komponenten, die eine mySAP.com-Lösung bilden.



**Abbildung 1.1** Technische Realisierung einer mySAP.com-Lösung mit mehreren Softwarekomponenten (Beispiel)

Wie ein SAP R/3 verfügen auch viele der neuen mySAP.com-Softwarekomponenten über die SAP-Basis mit der *Development Workbench* und den Basiswerkzeugen wie dem *Computer Center Management System (CCMS)*, sowie über eine eigene Datenbank (mit der dreistelligen Datenbank-ID, die in einer mySAP.com-Lösungslandschaft eindeutig sein muss). Anwendungsspezifische Softwareteile ergänzen das SAP-Basis-System und bilden somit ein SAP R/3, ein SAP BW oder ein SAP APO. Namentlich sind dies:

- ▶ SAP R/3 Enterprise
- ▶ SAP Advanced Planner and Optimizer (SAP APO)
- ▶ SAP Business Information Warehouse (SAP BW) und SAP Strategic Enterprise Management (SAP SEM)
- ▶ SAP Customer Relationship Management (SAP CRM) und SAP Enterprise Buyer Professional (SAP EBP)
- ▶ SAP Workplace (bis 2.11)

Im Text dieses Buches bezeichnen wir diese Softwarekomponenten mit SAP-Basis als *SAP-Systeme*.

Die mySAP.com-Lösungen werden komplettiert durch weitere eigenständige Softwarekomponenten ohne die herkömmliche SAP-Basis, z. B.:

- ▶ SAP Internet Transaction Server (SAP ITS)
- ▶ SAP liveCache
- ▶ SAP Business Connector (SAP BC)

Den Schwerpunkt der mySAP.com-Technologie bilden also die verschiedenen Komponenten und deren Zusammenspiel. Abbildung 1.1 zeigt ein Beispiel, wie die technische Grundlage einer mySAP.com-Lösung aussehen könnte.

Jede mySAP.com-Komponente kann autark operieren (d.h., z.B. unabhängig von anderen gestartet, gestoppt und gewartet werden). Der Datenaustausch zwischen den Komponenten erfolgt über *Remote Function Calls (RFCs)*. Zum elektronischen Austausch von betriebswirtschaftlichen Dokumenten hat die SAP einen eigenen Standard geschaffen: Zur Übertragung werden beliebige Daten in so genannte *IDocs* umgewandelt und über RFC versendet. Über RFC können Komponenten unterschiedlicher Versionen miteinander kommunizieren.

**Anbindung von Systemen über RFC**

Damit ein SAP R/3 und beispielsweise ein SAP APO oder SAP BW miteinander arbeiten können, müssen in das SAP R/3 als Plug-in bezeichnete Softwaretransporte implementiert werden.

Auf der linken Seite von Abbildung 1.1 finden sich die Schnittstellen, über die sich Benutzer an das System anmelden können (GUI-Schnittstellen). Dies sind der *SAP GUI for Windows Environment*, der *SAP GUI for Java Environment* sowie der *SAP GUI for HTML*. Spezialisierte Benutzer verwenden den SAP GUI für Windows oder Java Environment. Der Nachteil des SAP GUI for Windows Environment ist, dass er auf dem Desktop-Computer installiert werden muss. (Der Java-GUI muss ebenfalls auf dem Desktop-Computer installiert werden, allerdings ist die Installation deutlich weniger aufwendig als die des SAP GUI for Windows). Typische Benutzer des Windows- oder Java-GUIs sind Controller, Planer und Mitarbeiter in Vertriebszentren (z. B. im Telefonverkauf).

**Benutzeranbindung**

Gelegentliche Benutzer (Internet- oder Intranet-Benutzer) melden sich dagegen über einen Web-Browser an. Diese Art der Anmeldung hat den Vorteil, dass auf dem Desktop-Computer kein spezielles GUI-Programm installiert werden muss. Alle Ein- und Ausgabemasken werden in einem

Web-Browser im HTML-Format dargestellt. Die Kommunikation zwischen dem Web-Browser und der SAP-Applikationsebene wird über die Internetebene vermittelt (s. unten).

Eine dritte Möglichkeit des Zugriffs besteht über so genannte *Mobile Clients*, Laptop-Computer oder Handheld-Devices, die nicht ständig mit dem zentralen System verbunden sind, sondern periodisch mit diesem Daten austauschen. Dies wird z.B. für Mitarbeiter im Außendienstverkauf oder -service verwendet. Der Zugriff erfolgt z.B. über einen CRM-Server und einen DCOM-Connector.

**mySAP Workplace** Der *mySAP Workplace* vereinheitlicht den Zugriff auf alle beteiligten Systeme einer Systemlandschaft. Ein Benutzer meldet sich nur einmal am Workplace an, und der Workplace-Server stellt ihm eine so genannte »Portalseite« zusammen, auf der er alle Einstiegsinformationen findet, die er für seine tägliche Arbeit benötigt: zum einen sind dies Verweise auf Transaktionen der mySAP.com-Komponenten und auch auf Nicht-mySAP.com-Komponenten, Internet- und Intranet-Links sowie auf weitere Informationen (so genannte *MiniApps* oder *iViews*). Aus dieser Portalseite heraus kann der Benutzer alle täglichen Arbeiten ausführen, ohne sich darum kümmern zu müssen, in welchen Systemen er sich anmelden muss. Mit dem mySAP Workplace können sowohl Portalseiten für Mitarbeiter als auch für Kunden und Partner erstellt werden.

### 1.1.2 Client/Server-Architektur

Die Technik einer mySAP.com-Komponente beruht auf einer mehrstufigen Client/Server-Architektur, die in Abbildung 1.2 dargestellt ist. Dabei dient die Präsentationsebene (Frontend) der Datenein- und -ausgabe durch den Benutzer. In der Applikationsebene werden die eigentlichen Berechnungen ausgeführt, die den Geschäftsprozess repräsentieren. Die Datenbankebene dient der permanenten Speicherung und Bereitstellung der Daten.

**Präsentations-ebene** Präsentationsserver werden in der Regel als Ein-Personen-Rechner (PC) eingerichtet. Als GUI-Programm dient der klassische SAP GUI (SAP GUI für Windows oder Java Environment), der auf dem Desktop-Computer installiert werden muss, oder ein Web-Browser (z. B. von Microsoft oder Netscape).

**Applikations-ebene** Hat ein Benutzer seine Dateneingabe beendet, so sendet der Präsentationsserver diese Daten an die *Applikationsebene*. Die Applikationsebene beinhaltet die Applikationslogik und die Präsentationslogik. Die Applikationsebene bildet inklusive der integrierten Internetanbindung den SAP Web Application Server (SAP Web AS, ab SAP-Basis 6.10).

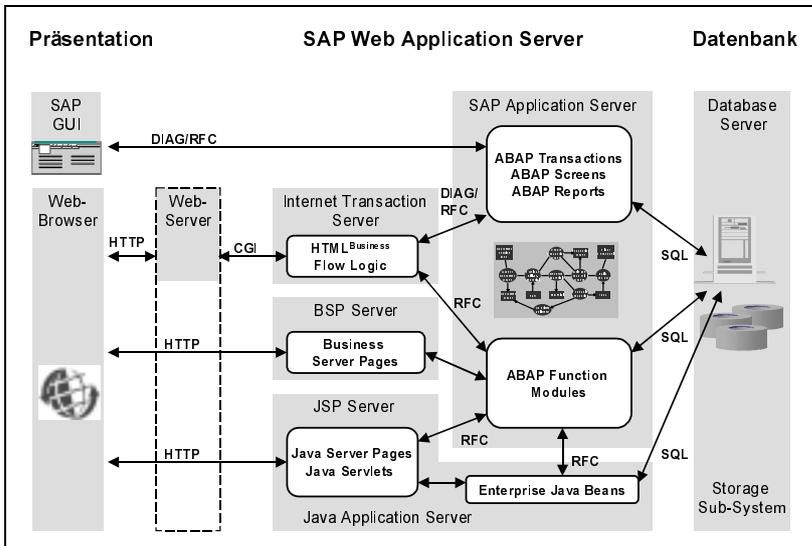


Abbildung 1.2 Client/Server-Architektur (SAP Web Application Server)

Die Applikationslogik ist in Transaktionen, Screens, Reports oder in Funktionsbausteinen gekapselt.

Für die Anbindung von Internetbenutzern an die SAP-Applikationsebene stehen drei Möglichkeiten zur Verfügung. Zum einen ist dies die Lösung unter Verwendung des SAP Internet Transaction Servers (SAP ITS) und eines Web-Servers eines Drittanbieters (z.B. Microsoft oder Netscape). Typische Anwendungen des SAP ITS sind Self-Service-Anwendungen im Bereich des Personalwesens (ESS-Anwendungen wie Zeiterfassung und Reiseabrechnung), Business-to-Business-Procurement-Anwendungen (Enterprise Buyer Professional) und Internet-Vertrieb (Produktkatalog, Online-Store im SAP R/3). Der SAP Internet Transaction Server sowie Internet-Anwendungen wie Produktkatalog, Online-Store und Employee Self Services sind seit SAP R/3 3.1 Bestandteil der R/3-Auslieferung. SAP R/3 4.6 erreicht die durchgängige Web-Fähigkeit der gesamten Funktionalität. Dabei stehen Easy Web Transactions (EWTs, mit den SAP-eigenen Script-Sprachen HTML<sup>Business</sup> und Flow Logic), Internet Application Components (IACs), SAP GUI for HTML und Web-RFC als Techniken zur Verfügung, HTML-Seiten zu generieren.

Das zweite Programmiermodell ist das der Business Server Pages (BSP), in dem mit ABAP oder JavaScript als Script-Sprache HTML-Seiten dynamisch generiert werden. Technisch hat dieses Programmiermodell den Vorteil, dass keine weitere Softwarekomponente installiert werden muss; Busi-

ness Server Pages werden direkt in den »normalen« SAP-Applikationsinstanzen generiert. Ein gesonderter Web-Server ist nicht zwingend erforderlich; aus Sicherheitsgründen empfiehlt sich aber, einen separaten Web-Server als Sicherheitspuffer zu installieren. Business Server Pages sind Bestandteil der SAP-Basis 6.10.

Die dritte Möglichkeit, den Internetbenutzer an die SAP-Applikationsebene anzubinden, ist die Verwendung eines JSP-Servers oder eines Java Application Servers. Die verwendeten Sprachen sind HTML und Java, die Programmiermodelle sind Java Server Pages (JSP) oder Java-Servlets. Aufgrund der Kompatibilität von Java-Entwicklungen können Sie den von der SAP ausgelieferten In-Q-My Java Application Server durch einen Java Application Server eines Drittanbieters (z. B. von BEA oder IBM) ersetzen, sobald dieser von der SAP freigegeben ist. Beispiele für Lösungen, die dieses Programmiermodell zur Realisierung der Präsentationslogik verwenden, sind mySAP CRM Internet Sales 3.0 und Enterprise Portal 5.0.

In Zukunft können Teile der Anwendungslogik ebenfalls in Java in einem Java Application Server realisiert sein (als sog. Enterprise JavaBeans, EJB). Zur Zeit des Drucks dieses Buches sind größere Anwendungen mit dieser Technik nicht realisiert.

**Datenbankebene** Sind für die Bearbeitung einer Benutzeranfrage Daten notwendig, die sich noch nicht im Hauptspeicher des Applikationsservers befinden, so werden diese vom Datenbankserver gelesen. Die relationale Datenbank ist das Medium zur permanenten Speicherung von Daten. Neben der SAP-eigenen Datenbank (SAP DB) unterstützt die SAP die Verwendung von Datenbanken wichtiger Hersteller (IBM, Oracle, Microsoft).

**Performanceaspekte und Tuningpotenzial** Da Präsentationsserver in der Regel als Ein-Personen-Rechner (PC) eingerichtet werden, besteht keine weitere Notwendigkeit des Tunings in diesem Bereich, sofern diese über die für das jeweilige SAP-Release empfohlene Hardwareausstattung verfügen.

Anders sieht die Situation in den darunter liegenden Ebenen der Client/Server-Architektur aus. Hier teilen sich die anfallenden Anfragen Prozesse und Speicherbereiche.

**Prozesskonfiguration** Dies stellen wir am Beispiel der Applikationsebene dar: Aufträge von Benutzern oder von anderen Systemen werden im Applikationsserver von *SAP-Workprozessen* bearbeitet. Typischerweise werden SAP-Systeme so konfiguriert, dass sich im Mittel 5–10 aktive Benutzer einen SAP-Dialogworkprozess teilen. Dabei gehen wir davon aus, dass die Benutzer etwa 10-mal so lange brauchen, um Daten am Bildschirm einzugeben und die

Ergebnisse zu interpretieren, wie das SAP-System benötigt, um die Benutzeraufträge zu bearbeiten. Daher sollten immer genug freie Workprozesse zur Verfügung stehen, um Benutzeranfragen ohne Verzögerung bearbeiten zu können. Starten einzelne Benutzer Reports mit einer sehr langen Antwortzeit (vielleicht sogar mehrere gleichzeitig), so belegen sie Workprozesse über mehrere Minuten hinweg. Die verbleibenden Workprozesse reichen dann nicht mehr aus, um die Anfragen anderer Benutzer zügig zu bearbeiten, und es kommt zu Wartezeiten. Das SAP-System sieht keine Möglichkeit vor, Benutzer zu priorisieren. Wenn ein Engpass entsteht, wartet der Vorstandsvorsitzende ebenso auf seine Antwort wie der Sachbearbeiter. Allerdings bieten Methoden der Lastverteilung die Möglichkeit, Applikationsserver für bestimmte Benutzergruppen zu reservieren. Neben dem Dialog- oder Online-Betrieb verarbeitet eine mySAP.com-Komponente auch noch Hintergrund-, Verbuchungs- und Druckaufträge. Für diese Aufgaben existieren eigene SAP-Workprozessstypen. Durch Tuningmaßnahmen in diesem Bereich wird die Last auf dem System bedarfsgerecht verteilt. Auch für die Server der Internet- und der Datenbankebene müssen die Prozesse zur Parallelverarbeitung von Anfragen geeignet konfiguriert werden.

Auf allen Ebenen sorgen *Puffer* (auch *Caches* genannt) dafür, dass Daten, die einmal in einen Server geladen wurden, anschließend im Hauptspeicher des Servers vorgehalten werden und für nachfolgende Anfragen zur Verfügung stehen. Im Falle der Applikationsebene werden z.B. Programme, Tabellen- und Felddefinitionen und Inhalte von Customizing-Tabellen in den Puffern vorgehalten. Die richtige Einstellung dieser Puffer gewährleistet, dass weniger Daten vom Datenbankserver gelesen werden müssen. Das Lesen über den SAP-Puffer ist etwa um den Faktor 10 bis 100 schneller als das Lesen über den Datenbankserver.

**Puffer (Caches)**

Beim Tuning der Datenbank unterscheidet man drei Bereiche. Zunächst einmal ist dies die richtige Einstellung der Datenbankpuffer sowie anderer Datenbankparameter. Der zweite Bereich ist das Optimieren des Festplatten-Layouts der Datenbank, um die Last möglichst gleichmäßig auf die Festplatten zu verteilen und so Wartesituationen beim Schreiben auf die Festplatte bzw. beim Lesen von der Festplatte zu vermeiden. Der dritte Aspekt beim Datenbanktuning ist die Optimierung lang laufender, »teurer« SQL-Anweisungen.

**Datenbanktuning**

Die Netzübertragungsgeschwindigkeit und der Datendurchsatz zwischen den Ebenen der Client/Server-Architektur ist von wesentlicher Bedeutung. Die Leistungsfähigkeit der gesamten mySAP.com-Lösung wird dadurch beeinflusst.

**Netzwerk**

Die mySAP.com-Architektur ist so konzipiert, dass der größte Datenverkehr zwischen Applikations- und Datenbankebene fließt. Durch die Optimierung teurer SQL-Anweisungen kann dieser zwar reduziert werden, dennoch wird in der Praxis die Applikations- und Datenbankebene durch ein *Local Area Network (LAN)* verbunden sein.

Der Datenverkehr zwischen Präsentations- und Applikationsebene ist dagegen möglichst gering, denn die Netzwerkverbindung hier kann sowohl ein LAN als auch ein *Wide Area Network (WAN)* sein.

**Internet** Bei der Verwendung des Web-Browsers als GUI ist bei der Programmierung darauf zu achten, dass möglichst wenig Daten zwischen Präsentations- und Internetebene übertragen werden. Die Gefahr, dass durch die Generierung aufwendiger HTML-Seiten lange Netzwerklaufzeiten den Benutzer beeinträchtigen, ist deutlich größer als bei der Verwendung des klassischen SAP GUI (der das SAP-eigene DIAG-Protokoll verwendet). Das Tuningpotenzial hängt stark von dem verwendeten Programmiermodell ab. Wird die Internetebene als reine »Übersetzungsebene« zwischen Präsentations- und Applikationsebene verwendet (wie etwa beim SAP GUI for HTML), so beschränkt sich das Optimierungspotenzial auf die Konfiguration. Je mehr Logik in die Internetebene verlagert wird (z.B. Feldprüfungen etc.), desto höher ist auch die Notwendigkeit der Programmanalyse in der Internetebene.

**Hardware** Sowohl auf dem Datenbankserver als auch auf den Applikationsservern muss überprüft werden, ob ausreichend *Hardware* (CPU und Hauptspeicher) zur Verfügung steht, um die anstehende Last zu bewältigen. Darüber hinaus muss sichergestellt werden, dass die Betriebssystem- und Netzwerkparameter optimal eingestellt sind, um eine gute Hardwareperformance zu erreichen. Zu diesem Punkt sei auch auf die Werkzeuge und die Literatur der Hardwarehersteller verwiesen.

**Skalierbarkeit** mySAP.com-Komponenten sind als Client/Server-Systeme skalierbar. Unter vertikaler *Skalierbarkeit* versteht man die Tatsache, dass die Softwarekomponenten aller Ebenen sowohl zentral auf einem Rechner installiert werden können als auch über mehrere Rechner verteilt. (Da nicht alle Softwarekomponenten für alle Betriebssysteme freigegeben sind, ist die zentrale Installation aller Komponenten allerdings nur auf ausgewählten Plattformen möglich.)

Innerhalb einer Client/Server-Ebene lässt sich die anfallende Last auf mehrere logische Instanzen verteilen, die auf unterschiedlichen Rechnern laufen können, was man als horizontale Skalierbarkeit bezeichnet. So

wird die Präsentationsebene in der Regel auf Personal-Computer (PCs) oder Terminal-Server verteilt. Die Applikationsebene wird durch die SAP-Instanzen realisiert, die Internetebene durch ITS-Instanzen und Instanzen des Web-Servers. Im Prinzip ist es bei einigen Datenbanksystemen (Oracle und DB2/390) möglich, auch die Datenbankebene durch das Einrichten mehrerer Datenbankinstanzen zu parallelisieren. In der Praxis wird von dieser Möglichkeit allerdings praktisch kein Gebrauch gemacht.

Die Client/Server-Architektur erlaubt es, die Zahl der Applikations-, Internet- und Präsentationsserver fast beliebig zu erhöhen, um bei steigender Benutzeranzahl den wachsenden Anforderungen gerecht zu werden. Die Datenbankebene wird jedoch von einem Datenbankserver gebildet (die Ausnahmen ORACLE Parallel Server und DB2/390 werden hier nicht besprochen). Gewisse Abläufe im Datenbankserver können prinzipiell nicht verteilt werden (z. B. die Sperrverwaltung). Die Erfahrung zeigt, dass Performanceprobleme in großen R/3-Installationen mit mehr als zehn Applikationsservern meist durch Engpässe auf dem Datenbankserver hervorgerufen werden. Dem Tuning der Datenbank kommt daher mit zunehmender Systemgröße eine wachsende Bedeutung zu. Wenn ein System eine gewisse Zeit produktiv ist, sind häufig die meisten Tuningmaßnahmen wie Puffereinstellungen, Lastverteilung etc. optimiert. Das Tuning teurer SQL-Anweisungen gewinnt dagegen mit wachsender Datenbankgröße an Bedeutung und ist ein ständig fortlaufender Prozess.

## **1.2 Das Überwachungs- und Optimierungskonzept für eine mySAP.com-Lösung**

Nachfolgende Abschnitte widmen sich dem Überwachungs- und Optimierungskonzept für eine mySAP.com-Lösung. Zunächst sollen die Anforderungen an ein solches Konzept behandelt werden.

### **1.2.1 Anforderungen an ein Überwachungs- und Optimierungskonzept**

Um die Erwartungen eines Benutzers an eine mySAP.com-Anwendung zu erfüllen, ist ein Überwachungs- und Optimierungskonzept notwendig. In dem Maße wie sich die SAP heute ihren Kunden nicht mehr als Softwareanbieter, sondern als Lösungsanbieter profiliert, hat sich auch die Aufgabe des Monitorings geändert. Statt vom klassischen System Monitoring sprechen wir heute vom Solution Monitoring, das nicht mehr nur die einzelnen Systemkomponenten überwacht, sondern den Geschäftsprozess (Business Process) als Ganzes.

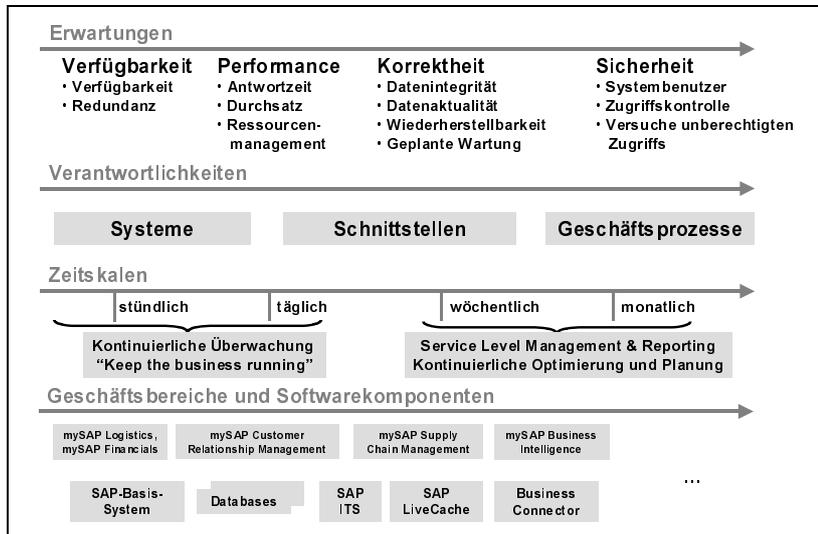


Abbildung 1.3 Anforderungen an die Überwachung einer mySAP.com-Lösung (Solution Monitoring)

**Erwartungen aus Sicht der Benutzer**

Die Benutzer einer mySAP.com-Anwendung sind entweder die Mitarbeiter, Kunden oder Partner (z.B. Zulieferer) des Unternehmens, das die Anwendung besitzt. Damit ist die Zufriedenheit der Benutzer eines der ersten Ziele, das beim Betrieb der Anwendung gesichert sein muss. Damit fordern wir von unserem Überwachungskonzept, dass es dazu führt, diese Erwartungen zu erfüllen.

Fragen Sie sich also: Was sind Ihre Erwartungen an eine Internet-Anwendung, mit der Sie z.B. eine Warenbestellung oder eine Banktransaktion tätigen wollen? Sie werden auf vier Forderungen kommen: Die Anwendung soll erstens verfügbar sein, wenn Sie sie brauchen, sie soll zweitens mit einer angemessenen Performance funktionieren, sie soll drittens korrekt laufen (mit anderen Worten: Sie wollen das Produkt geliefert bekommen, das im Internet abgebildet war, und es soll der Preis auf der Rechnung mit dem übereinstimmen, den Ihnen das Angebot genannt hat), und viertens wollen Sie sich darauf verlassen können, dass die Anwendung sicher ist, d.h., dass niemand Ihre Daten manipulieren kann.

**Verfügbarkeit**

Um die *Verfügbarkeit* – die erste Erwartung eines Benutzers an eine Anwendung – zu gewährleisten, bieten die meisten auf dem Markt befindlichen System-Management-Plattformen eine Überwachung von Hardware- und Softwarekomponenten an. Dies reicht aber nicht aus, um zu gewährleisten, dass der Geschäftsprozess verfügbar ist. Ein Benutzer

empfindet eine Anwendung auch dann als »nicht verfügbar«, wenn eine Kommunikationsstörung zwischen den Komponenten oder ein schwerer Anwendungsfehler eine Dateneingabe oder -abfrage unmöglich macht. Eine Verfügbarkeitsüberwachung muss also garantieren, dass der Geschäftsprozess verfügbar ist, nicht nur einzelne Komponenten.

Schlechte *Performance* einer E-Business-Anwendung ist eine leichte Möglichkeit, einen Kunden zu verärgern. Die Performance betrifft dabei den Dialogteil der Anwendung, d.h. den Teil, in dem Daten eingegeben und gespeichert werden, sowie die automatische Hintergrundverarbeitung, die Daten weiter verarbeitet, ohne dass der Kunde online ist. Schlechte Performance der Dialoganwendung trifft den Kunden sofort bei der Dateneingabe, schlechte Performance der angeschlossenen Hintergrundanwendungen trifft ihn indirekt, weil etwa das von ihm bestellte Produkt nicht in der versprochenen Zeit geliefert wird.

**Performance**

Die Performance der Dialoganwendungen und der Hintergrundanwendungen sind in der Regel unabhängig voneinander zu bewerten. Zur Performanceüberwachung gehört die bedarfsgerechte Ressourcenzuteilung, wenn beide Arten von Anwendungen auf einem technischen IT-System laufen (was normalerweise der Fall ist). In der Regel wird man Dialoganwendungen höhere Priorität einräumen als Hintergrundanwendungen. Ausnahmen bestätigen dagegen die Regel: Hintergrundanwendungen mit einer festen Deadline beanspruchen wiederum die höchste Priorität. Als Beispiel führen wir den Druck von Lieferunterlagen (Lieferscheine, Adressaufkleber, Rechnungen etc.) an. Wenn diese nicht zu einem bestimmten Zeitpunkt fertig sind, ergibt sich unter Umständen eine Lieferverzögerung von 24 Stunden (Postausgang!).

Um die *Integrität* (oder Korrektheit) einer Anwendung zu garantieren, sind folgende Bereiche in einem Überwachungskonzept zu berücksichtigen:

**Integrität**

► **Integrität der Daten**

Verletzungen der Datenintegrität sind z.B. korrupte Daten in der Datenbank, Netzwerkfehler, Fehler in den Schnittstellen.

► **Aktualität und Konsistenz von Softwarekomponenten und Daten**

Werden Softwarekomponenten oder Stamm- und Customizingdaten nicht korrekt aktualisiert, kann es zu Inkonsistenzen in den Belegen kommen, z.B. wird im Internet-Katalog ein Preis angezeigt, der nicht mit dem im Abrechnungssystem gespeicherten übereinstimmt – der Kunde bekommt also eine falsche Rechnung.

► **Sicherung (Backup) von Datenbanken und Dateisystemen und deren Zurücksetzbarkeit (Recoverability)**

Sind die in der produktiven Datenbank abgelegten Inhalte durch einen Hardware- oder Softwaredefekt »unrettbar« zerstört, so müssen diese durch ein Backup wiederhergestellt werden können.

**Sicherheit** Die Überwachung der *Sicherheit* ist in der Regel so aufwändig wie das Sicherheitskonzept selbst. In diesem Buch wollen wir nicht auf diesen Bereich eingehen.

Das klassische System Monitoring überwacht bestenfalls jede Softwarekomponente einzeln. Angesichts der Vielzahl von Komponenten, die an einer Lösung beteiligt sein können, kann es jedoch vorkommen, dass zwar jede Komponente für sich korrekt arbeitet, der Geschäftsprozess dem Endanwender dennoch nicht performant, korrekt und sicher zur Verfügung steht. Dies kann z.B. durch eine fehlerhafte Kommunikation zwischen Komponenten geschehen. Ein Solution Monitoring muss daher Geschäftsprozess-spezifische Überwachungsteile haben, die komponentenübergreifend den Geschäftsprozess überwachen (zweiter Bereich in Abbildung 1.3).

Im dritten Bereich von Abbildung 1.3 ist dargestellt, dass ein Überwachungskonzept sich auf verschiedene Zeithorizonte beziehen muss. Mit anderen Worten: Alert-Informationen sind auf unterschiedlichen Zeitskalen interessant. Machen wir dies anhand eines Beispiels klar: Wenn eine Komponente »herunterfällt«, so ist das eine Information, die auf der Zeitskala »Minuten« relevant ist. Der Information, dass in der vergangenen Nacht das Backup schief gegangen ist, muss im Bereich von Stunden nachgegangen werden. Wenn das System schließlich einen Alarm auslöst, weil es den Administrator darüber informiert, dass das extrapolierte Datenbankwachstum dazu führen wird, dass in vier Wochen der Plattenplatz nicht mehr ausreicht, so ist das ebenfalls eine wertvolle Information, die den Administrator zum Handeln bewegen sollte (vielleicht kann er ja noch eine Archivierung in Gang bringen, bevor neue Platten bestellt werden). Allerdings wird niemand wollen, dass deshalb vier Wochen lang der Alertmonitor eine rote Ampel zeigt. Ein Überwachungskonzept muss daher sowohl ein kurzfristiges Monitoring vorsehen – eine Überwachung von Ausnahmesituationen – als auch ein mittel- und langfristiges Reporting- und Optimierungskonzept.

**Zentrales Monitoring und Erweiterbarkeit**

Schließlich – diesen Aspekt des Überwachungskonzeptes stellt der vierte Bereich von Abbildung 1.3 dar – ist die stetig ansteigende Anzahl von IT-Lösungen und Komponenten eine Herausforderung für die Administra-

tion in den Rechenzentren; die Anzahl hat sich von dem »überschaubaren« SAP R/3 (mit SAP-Instanzen, Datenbank, Hardware/Betriebssystem) zu einer ständig wachsenden Technologiepalette entwickelt – Produkte eingeschlossen, die die SAP nicht selbst produziert, sondern für die sie als Reseller auftritt. Als Konsequenz dieser Entwicklung wird jeder Rechenzentrumsleiter eine Methode für die Überwachung fordern, die sowohl zentral ist (alle Informationen in einem Werkzeug) als auch erweiterbar auf neue Komponenten hin.

Folgende Forderungen an ein Überwachungs- und Optimierungskonzept haben wir bisher zusammengestellt (Abbildung 1.3):

**Zusammenfassung**

- ▶ Ein Überwachungs- und Optimierungskonzept muss die Bereiche Verfügbarkeit, Performance, korrekter Betrieb und Sicherheit abdecken.
- ▶ Ein Überwachungs- und Optimierungskonzept darf sich nicht darauf beschränken, einzelne Hardware- und Softwarekomponenten zu betrachten. Es muss vielmehr den Informationsfluss von Geschäftsprozessen über Komponenten hinweg überwachen und optimieren.
- ▶ Die Überwachung muss verschiedene Zeitskalen im Blick haben: Von Fehlersituationen, die innerhalb kürzester Zeit behoben werden müssen, bis zu einer langfristigen Trendanalyse und Planung.
- ▶ Das Konzept muss alle Geschäftsprozesse und Komponenten wenn möglich in einem zentralen Werkzeug erfassen.

Um diese Anforderungen an die Überwachung von mySAP.com-Lösungen in den Griff zu bekommen, hat die SAP das Solution-Management-Programm mit dem SAP Solution Manager als zentralem Werkzeug aufgesetzt. Dieses stellen wir in den nächsten Abschnitten vor.

### **1.2.2 Werkzeuge und Methoden für das Überwachungs- und Optimierungskonzept**

Für die Überwachung und die Performanceanalyse umfasst das SAP-Basis-System eine Reihe leistungsfähiger Monitoring-Programme, die ständig von erfahrenen Performanceexperten der SAP weiterentwickelt werden. Dies sind zunächst eine zentrale Monitoringarchitektur, die die kontinuierliche Systemüberwachung aller Hard- und Softwarekomponenten übernimmt, ein zentraler Workload-Monitor und weitere Expertenwerkzeuge zur detaillierten Analyse.

Die kontinuierliche Systemüberwachung prüft, dass alle Komponenten verfügbar sind und performant arbeiten. Ist dies nicht der Fall, so wird ein Alarm ausgelöst. Die kontinuierliche Überwachung kann über die zentrale Monitoringarchitektur automatisiert werden. Zur kontinuierlichen

**Zentraler Überwachungsmonitor**

Systemüberwachung verwenden Sie den zentralen CCMS-(Computer Center Management System-)Überwachungsmonitor (Transaktionscode RZ20). Dabei definieren Sie ein SAP-System als zentrales Monitoringsystem, in dem dann die Fehlermeldungen aus allen mySAP.com-Komponenten einlaufen. Bisher sind in den *zentralen Überwachungsmonitor* die Daten über die SAP-Instanzen, Datenbanken, Betriebssysteme sowie über weitere mySAP.com-Komponenten, wie den SAP ITS, eingebunden. Weitere Data Supplier, auch für Nicht-SAP-Komponenten wie Catalog-Software, Clarify und Commerce One werden angeboten.

**Zentraler Workload-Monitor**

Der zentrale *Workload-Monitor* erlaubt Ihnen eine Übersicht über die Lastverteilung in Ihrer mySAP.com-Lösung über alle Softwarekomponenten hinweg. Bei der technischen Analyse lassen sich z. B. Probleme auf der Datenbank, im SAP-Basis-System oder auf dem Internet Transaction Server identifizieren und analysieren. Bei der Anwendungsanalyse lassen sich Transaktionen, Programme und Benutzer identifizieren und analysieren, die eine hohe Systembelastung darstellen.

**Expertenmonitore zur Performanceanalyse**

Weitere *Expertenmonitore* zur Performanceanalyse von SAP-Systemen sind im Performancemenü (Transaktionscode STUN) zusammengefasst:

**Werkzeuge · Administration · Monitor · Performance** oder  
**Werkzeuge · CCMS · Control/Monitoring · Performance menu**

In Tabelle 1.1 sind die Performancemonitore zusammengestellt, die Ihnen zur Basis- und Anwendungsanalyse zur Verfügung stehen.

Monitore für die technische Analyse	
Betriebssystemmonitor (ST06)	Auslastung der CPU und des physischen Hauptspeichers
Datenbankmonitor (ST04)	Auslastung der Datenbankpuffer; Datenbanksperren und sonstige Wartesituationen; Schreib- und Lesezugriffe auf die Festplatten; Überwachung von SQL-Anweisungen
R/3-Speicherkonfigurationsmonitor (»Setup/Buffer«, ST02)	Auslastung der R/3-Puffer und sonstiger Speicherbereiche, R/3-Workprozesse
Workprozess-Übersicht (SM50)	Auslastung der R/3-Workprozesse
Workload-Monitor (ST03)	Übersicht über die Lastverteilung im R/3-System. Bei der technischen Analyse lassen sich z. B. Probleme auf der Datenbank, der R/3-Speicherverwaltung oder den R/3-Puffern identifizieren und analysieren.

**Tabelle 1.1** SAP-Performancemonitore

Monitore für die Applikationsanalyse	
Workload-Monitor (ST03, ST03N, ST03G)	Übersicht über die Lastverteilung im R/3-System. Bei der Applikationsanalyse lassen sich Transaktionen, Programme und Benutzer identifizieren und analysieren, die eine hohe Systembelastung darstellen.
Anwendungsmonitore (ST07, ST14)	Überwachung des Ressourcenverbrauchs nach R/3-Modulen
SQL-Trace (ST05); ABAP-Trace (SE30)	Trace-Funktionen zur detaillierten Analyse von ABAP-Programmen

**Tabelle 1.1** SAP-Performancemonitore (Forts.)

Softwarekomponenten der SAP, die nicht auf der SAP-Basis beruhen, werden mit eigenen Expertenwerkzeugen zur Administration, Überwachung und Analyse ausgeliefert.

Alle Performancemonitore der SAP verfügen über offene *Schnittstellen*, die es den Partnern der SAP ermöglichen, Performancedaten über SAP-Systeme abzurufen. Dies bedeutet, dass Sie, wenn Sie eine externe System-Management-Software zur Systemüberwachung verwenden, auf die Performancedaten der SAP zugreifen und somit Ihre mySAP.com-Lösung systemseitig überwachen können. Als Beispiele für Überwachungswerkzeuge, die auf SAP-Performancedaten zugreifen, seien OpenView (HP), Tivoli (IBM) oder Patrol (BMC Software) genannt. Einschränkend sei genannt, dass mit diesen Produkten nur eine systemseitige Überwachung möglich ist (Outside-in-Ansatz). Eine Anwendungsüberwachung ist mit diesen Produkten nicht möglich. Diesen Punkt führen wir im nächsten Abschnitt weiter aus.

**Schnittstellen**

Mit Hilfe dieser leistungsfähigen Werkzeuge können Sie nun ein Überwachungs- und Optimierungskonzept aufbauen, das im Wesentlichen aus drei Bausteinen besteht: erstens dem Service Level Management, indem die Leistungsziele Ihrer Anwendung definiert und kontrolliert werden, zweitens dem Konzept für eine kontinuierliche Überwachung, die Abweichungen vom Normalbetrieb erkennt (wenn möglich präventiv) und den normalen Betriebszustand möglichst umgehend wiederherstellt, und drittens dem Optimierungskonzept, das aufbauend auf den Ergebnissen des Service Level Management und der kontinuierlichen Überwachung die kontinuierliche Optimierung der Anwendung umfasst. In den folgenden Abschnitten finden Sie konkrete Anleitungen, wie Sie diese drei Bausteine planen.

**Monitoring-Services**

### 1.2.3 Service Level Management

Neben dem kurzfristigen Alert-Monitoring ist das so genannte Service Level Management (SLM) zur langfristigen Überwachung und Optimierung von zentraler Bedeutung. Dieses wird bereits von vielen IT-Organisationen zum Management der Beziehungen zwischen den einzelnen Service Providern und dem Geschäftsprozessinhaber eingesetzt. Als Service Level Management bezeichnet man eine strukturierte proaktive Methode, die das Ziel hat, den Benutzern einer IT-Anwendung ein adäquates Serviceniveau zu garantieren – in Übereinstimmung mit den betriebswirtschaftlichen Zielen des Auftraggebers und bei optimalen Kosten. Diese Methode beinhaltet klar definierte, überprüfbare Ziele und eine klare Kommunikation zwischen den Geschäftsprozessinhabern und den Betreibern einer Lösung (dies können für Server, Datenbanken, Netzwerke etc. mehrere interne oder externe Betreiber sein). Das Service Level Management besteht zunächst aus einem Service Level Agreement, in dem die oben zu erreichenden Ziele bezüglich Verfügbarkeit, Performance, Korrektheit und Sicherheit definiert werden und auch festgelegt wird, wie die Erreichung der Ziele gemessen und kommuniziert werden soll. Das Service Level Reporting berichtet über die Zielerreichung in einem festgelegten Zeitraum. Primäres Ziel des Service Level Reportings ist es also festzustellen, ob die festgelegten Betriebsziele erreicht wurden, und potenzielles Optimierungspotenzial aufzuzeigen.

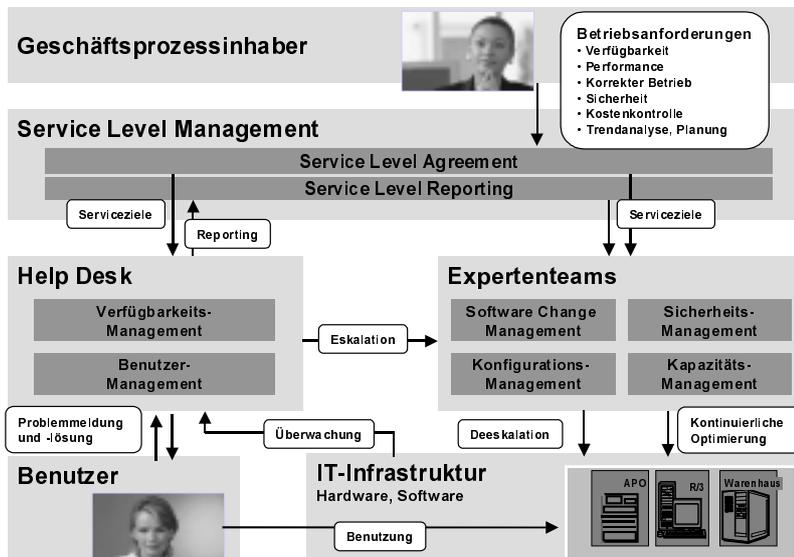


Abbildung 1.4 Wirkzusammenhänge im Umfeld von Service Level Management, Alert-Monitoring und kontinuierlicher Systemüberwachung

Hinzu kommt, dass der Geschäftsprozessinhaber die Zufriedenheit der Endbenutzer bei optimal niedrigen Kosten (cost of ownership) erreichen möchte. Ein Service Level Management sollte also – neben der Überwachung von Verfügbarkeit, Performance, Korrektheit und Sicherheit – auch die Kosten, z. B. für Hardware und Personal, transparent machen. Die – in der Praxis oft schwierige – Kommunikation zwischen den Geschäftsprozessinhabern und den Service Providern in den Griff zu bekommen, ist eine weitere Anforderung an ein Service Level Management.

### **Service Level Management einrichten**

Um in Ihrem SAP-Projekt zu einem erfolgreichen Service Level Management zu kommen, vereinbaren Sie zunächst ein Service Level Agreement, in dem Sie die Ziele festhalten, die von den einzelnen Servicepartnern erreicht werden sollen, die Maßnahmen festlegen, die im Falle des Nichterreichens von Zielen getroffen werden sollen und vereinbaren, wie die Erreichung der Ziele gemessen und dargestellt werden soll. Im Detail sollte ein Service Level Agreement folgende Vereinbarungen umfassen:

- ▶ Definition der Betriebszeiten
- ▶ Datenbanksicherungen und -wiederherstellungen
- ▶ Performance
- ▶ Inhalt des Reportings

Bitte beachten Sie, dass unsere Ausführungen keine Vorlage für einen »juristisch wasserdichten« Vertrag sind. Vielmehr beschränken sie sich auf fachlich sinnvolle Inhalte.

Beginnen Sie im Service Level Agreement mit der Beschreibung der Betriebszeiten. Als Beispiel könnten Sie z.B. drei Betriebszeiten definieren: Betriebszeit A, B und C. Definieren Sie für jede Betriebszeit:

#### **Definition der Betriebszeiten**

- ▶ **Geltungsdauer**

Beispiel: Montag bis Freitag, 8.00 Uhr bis 17.00 Uhr für Betriebszeit A.

- ▶ **Verfügbarkeit des Servicepersonals**

Beispiel: In Betriebszeit A steht ein Helpdesk mit ausreichender Kapazität zur Bewältigung der Anfragen von Endbenutzern zur Verfügung. Zusätzlich stehen alle für den Betrieb und für eine eventuelle Problemlösung notwendigen Experten zur Verfügung. In Betriebszeit B und C steht nur ein Notdienst zur Verfügung. Experten sind in Betriebszeit B über eine Rufbereitschaft erreichbar.

► **Verfügbarkeit, geplante und ungeplante Ausfallzeiten**

Beispiel: In Betriebszeit A sind geplante Ausfallzeiten nicht möglich, ungeplante Ausfallzeiten maximal 2 Stunden am Tag, maximal 4 Stunden im Monat. In Betriebszeit B sind geplante Ausfallzeiten nach Rücksprache mit dem Geschäftsprozessinhaber möglich, ungeplante Ausfallzeiten maximal 4 Stunden am Tag, maximal 12 Stunden im Monat. In Betriebszeit C sind Ausfallzeiten jederzeit möglich. (Ausnahmen sind Sondersituationen wie Software-Upgrades, die eventuell erweiterte Ausfallzeiten mit sich bringen können).

Alle geplanten und ungeplanten Ausfallzeiten sind mit Angabe der Ursache des Ausfalls in den Service Level Report aufzunehmen.

**Backup und Recovery**

Das Service Level Agreement sollte die Verantwortlichkeiten für Sicherungen und Wiederherstellungen von Datenbanken und falls nötig von Dateisystemen festhalten (Backup und Recovery). Legen Sie also den Umfang der durchzuführenden Sicherungen fest. Definieren Sie eine Prozedur für die ordnungsgemäße Wiederherstellung von Datenbanken und Dateisystemen im Fehlerfall. Die dazu maximal notwendige Zeit ergibt sich aus der maximal erlaubten Ausfallzeit für ungeplante Ausfälle.

**Performance**

Serviceanbieter geben häufig Garantien für mittlere Dialogantwortzeiten. Als »magische Grenze« für eine gute Performance wird in der Praxis oft eine mittlere Dialogantwortzeit von einer Sekunde angesehen. Ein derart pauschales Kriterium wird jedoch der Vielfalt der unterschiedlichen Anforderungen an mySAP.com-Komponenten nicht gerecht.

Stattdessen sollten Vereinbarungen über die Überwachung von SAP-Dialogtransaktionen getroffen werden. Die Antwortzeiten von SAP-Dialogtransaktionen können sowohl mit dem zentralen Überwachungmonitor als auch mit dem Workload-Monitor analysiert werden. Gehen Sie wie folgt vor, um zu einem sinnvollen Service Level Agreement zu kommen:

1. Wählen Sie ca. 10 bis 20 kritische Transaktionen aus, deren Performance überwacht werden soll. Ob eine Transaktion »kritisch« ist, entscheiden Sie anhand folgender Kriterien:

- Müssen Sie bei schlechter Performance einer Transaktion mit unmittelbaren wirtschaftlichen Schäden rechnen (Vertragsstrafen, entgangene Aufträge etc.)?
- Müssen Sie bei schlechter Performance einer Transaktion mit schweren Imageschäden rechnen? Ist die Transaktion direkt Ihren Kunden bzw. Partnern zugänglich (z. B. über das Internet) oder wird die Transaktion im direkten Umgang mit Kunden bzw. Partnern benutzt (z. B. im Telefonverkauf)?

- ▶ Gehört die Transaktion zu den am häufigsten ausgeführten (nachzu- prüfen im Transaktionsprofil des Workload-Monitors)?

Bitte beachten Sie, dass es nicht möglich sein wird, in einem Service Level Agreement eine lückenlose Performanceüberwachung von Geschäftsprozessen zu vereinbaren. Dennoch sollten die ausgewählten Transaktionen eine möglichst repräsentative »Stichprobe« ihrer wichtigsten Anwendungen bilden.

2. Messen Sie die mittlere Antwortzeit der Transaktion im produktiven Betrieb über eine gewissen Zeit hinweg (im Transaktionsprofil des Workload-Monitors). Stellen Sie parallel fest, ob die Anwender mit der subjektiv erfahrenen Antwortzeit zufrieden sind. Ist dies der Fall, so legen Sie die gemessene Antwortzeit plus einer Marge von z.B. 50% als Schwellenwert für eine gute Performance fest.
3. Vereinbaren Sie, dass über die Antwortzeiten für die ausgewählten Transaktionen wöchentlich im Service Level Reporting berichtet wird.
4. Binden Sie die ausgewählten Transaktionen in den CCMS-Überwachungsmonitor ein (siehe Abschnitt 1.2.4, »Kontinuierliche Überwachung«).
5. Vereinbaren Sie, dass im Falle des Überschreitens des festgelegten Schwellenwertes eine Analyse von allen an der Transaktion beteiligten Personen durchgeführt werden soll. Ein gemeinsamer Aktionsplan soll daraus erstellt und die Maßnahmen von den dort benannten Personen durchgeführt werden, um eine stabile Performance wiederherzustellen.

Fehler im laufenden Betrieb erreichen auf zwei Wegen die Aufmerksamkeit der System- bzw. Applikationsbetreuer: Entweder indem ein Benutzer diesen meldet oder durch aktives Monitoring.

#### Fehlerbehebung

Legen Sie im Service Level Agreement die Prioritäten für Fehler und die Reaktionszeiten für die Fehlerbehebung fest.

In der Kommunikation zwischen SAP und Kunden werden vier Prioritäten verwendet (SAP-Hinweis 67739):

- ▶ Sehr hoch (Beeinträchtigung, die zum Stillstand des ganzen Systems oder eines geschäftskritischen Prozesses führt)
- ▶ Hoch (erhebliche Beeinträchtigung eines geschäftskritischen Prozesses)
- ▶ Mittel (Beeinträchtigung eines Prozesses)
- ▶ Niedrig (kleine Fehler, ergänzende Fragen)

Legen Sie außerdem die initialen Reaktionszeiten fest, d.h., ab wann mit der Fehleranalyse begonnen wird, sowie die Gesamtbearbeitungsdauer, d.h. die Zeit, ab wann ein Fehler behoben sein sollte.

Beschreiben Sie des Weiteren, welche Fehlerzustände vom Monitoringteam zu überwachen sind. Dazu sollten gehören:

- ▶ Abgebrochene Verbuchungen
- ▶ Abgebrochene Hintergrundprozesse
- ▶ Abgebrochene Schnittstellenprozesse  
(transaktionale RFC, queued RFC, Idoc, ALE)

Definieren Sie, wie die Fehlerbehebung auszusehen hat. Im folgenden Beispiel zeigen wir auf, wie der Fehlerbehandlungsprozess für abgebrochene Verbuchungen aussehen sollte: (Abgebrochene oder nicht ausgeführte Verbuchungen führen dazu, dass von den Benutzern angelegte oder geänderte Dokumente nicht endgültig in den zugehörigen Applikationstabellen gesichert werden und daher für den Benutzer nicht existent sind. Die tägliche Kontrolle der Verbuchungsaufträge ist daher eine wichtige Aufgabe des SAP-Systemadministrators. Wird abgebrochenen Verbuchungen nicht sofort nachgegangen, so besteht nach mehreren Tagen kaum noch eine Chance, der Ursache des Fehlers auf den Grund zu kommen.) Das Monitoringteam überwacht abgebrochene Verbuchungen in allen Produktivsystemen. Als initiale Reaktionszeit (das ist in diesem Fall die Zeit, die zwischen dem Auftreten des Fehlers und dem Bemerkten des Fehlers durch das Monitoringteam vergeht) wird für die Betriebszeit A eine Stunde festgelegt. Das Monitoringteam leitet die Fehlerinformation an den zuständigen Fachbetreuer weiter. Dazu liegt dem Monitoringteam eine Liste der Transaktionscodes und der zuständigen Betreuer vor. Der Fachbetreuer kontaktiert den Anwender, bei dem der Fehler aufgetreten ist, behebt die abgebrochene Verbuchung im System und klärt eventuell weitere Schritte in der Fachabteilung ab, um zu verhindern, dass vergleichbare Fehlersituationen wieder auftreten. Dazu wird eine maximale Bearbeitungszeit festgelegt, die wiederum von der Wichtigkeit der Transaktion für den Geschäftsbetrieb abhängen kann. Als Richtwert nehmen wir 8 Stunden Arbeitszeit (d.h. Zeit innerhalb der Betriebszeit A) für geschäftskritische Transaktionen an.

Legen Sie schließlich fest, dass die Anzahl der abgebrochenen Verbuchungen pro Zeitraum im Service Level Report festzuhalten ist.

**Eskalationsverfahren**

Ist zu einem bestimmten Zeitpunkt abzusehen, dass ein Problem nicht im Rahmen der vereinbarten Serviceziele zu beheben ist, so muss das Problem »eskaliert« werden, d.h., eine Stelle mit weitreichender Verantwortung muss entscheiden, wie mit dem Problem zu verfahren ist.

Beschreiben Sie im Service Level Agreement, wann und wer im Eskalationsfalle zu benachrichtigen ist. Definieren Sie, wenn nötig, mehrere Eskalationsstufen. Das *Eskalationsverfahren* sollte alle Hierarchiestufen des Kundenunternehmens und des Serviceanbieters umfassen. Legen Sie fest, dass eine Eskalation im Service Level Report erscheinen muss.

Nehmen Sie in den *Service Level Report* – neben den bereits benannten Kenngrößen – weitere Größen, die für SAP-Systeme charakteristisch sind, auf. Beispiele:

**Service Level Reporting**

- ▶ Anzahl der angemeldeten Benutzer und Anzahl der Transaktions-schritte
- ▶ Mittlere Antwortzeit im Dialog- und im Updatetask
- ▶ Mittlere Antwortzeit in der Stunde mit der höchsten Systembelastung
- ▶ CPU- und Hauptspeicherauslastung in der Stunde mit der höchsten Systembelastung
- ▶ Datenbankgröße und -wachstum

Aus diesen Informationen lassen sich oft mittel- und langfristige Trends ablesen, die ein rechtzeitiges Eingreifen möglich machen (z. B. Archivierungsmaßnahmen bei hohem Datenbankwachstum).

Halten Sie fest, wie oft der Service Level Report zu erstellen ist (Vorschlag: wöchentlich) und an welche Personen er verteilt wird.

Das Service Level Management im *SAP Solution Manager*, den wir im letzten Abschnitt dieses Kapitels beschreiben, umfasst die zuvor beschriebenen Aspekte. Es besteht aus einem Setup, in dem Sie die verschiedenen Parameter Ihres Service Level Agreements (Betriebszeiten, kritische zu überwachende Transaktionen etc. ) eintragen. Basierend auf diesen Daten wird dann vom SAP Solution Manager wöchentlich ein Service Level Report erstellt. Das Service Level Reporting im SAP Solution Manager umfasst folgende Funktionen:

**SAP Solution Manager**

- ▶ Automatisiertes Setup, z. B. zur Aufnahme kritischer Transaktionen
- ▶ Darstellung ausgewählter Performanceindikatoren, Fehlermeldungen und Optimierungsvorschläge für alle Komponenten einer mySAP.com-Lösung, basierend auf den Ergebnissen des SAP EarlyWatch Alert Services
- ▶ Darstellung von Performanceindikatoren, Fehlermeldungen und Optimierungsvorschlägen gruppiert nach Systemen und nach Geschäftsbereichen

- ▶ Grafische Darstellung von Zeitreihen ausgewählter Performanceindikatoren (Trendanalyse), z. B. zur Hardwareauslastung
- ▶ Automatisch generierte Handlungsanweisungen und Verweis auf weitere Services der SAP

### 1.2.4 Kontinuierliche Überwachung

Das CCMS (Computing Center Management System) enthält einen Monitor, mit dem Sie alle Aspekte Ihrer mySAP.com-Lösung überwachen können. Zusammen mit dem SAP Solution Manager bildet es eine vollständige Lösung für eine kontinuierliche Überwachung. Der Monitor bietet Ihnen:

- ▶ komplette, detaillierte Überwachung (Performanceindikatoren) von mySAP.com-Softwarekomponenten, Rechnern, Datenbanken und von Fremdkomponenten
- ▶ Statuskennzeichen (Alerts: grün, gelb, rot) für Performanceindikatoren, wenn Schwellwerte über- oder unterschritten werden
- ▶ Alert-Verfolgung und -Verwaltung
- ▶ beliebige Erweiterbarkeit (auch für Nicht-SAP-Softwarekomponenten) durch offene Struktur
- ▶ zusammen mit dem SAP Solution Manager eine grafische Oberfläche zur Überwachung und Fehlerverfolgung

Der zentrale Überwachungsmonitor steht mit SAP-Basis 4.0 zur Verfügung. Für die zu überwachenden Systeme ist die Minimalversion SAP-Basis 3.0 nötig.

In diesem Abschnitt finden Sie nach einer Einführung in den Überwachungsmonitor konkrete Vorschläge, wie Sie den »Baum« im Überwachungsmonitor an Ihre mySAP.com-Lösung anpassen, wie Sie mit dem Überwachungsmonitor ein Eskalationsverfahren, z. B. eine Benachrichtigung per E-Mail, Pager etc., einrichten und wie Sie den Überwachungsmonitor an die grafische Oberfläche des Solution Managers anschließen. Zusammen bilden diese Maßnahmen eine vollständige Überwachungslösung.

#### Performanceindikatoren

Der Zustand einer IT-Lösung wird durch so genannte *Performanceindikatoren* charakterisiert. Performanceindikatoren können sein:

- ▶ Zahlenwerte (Counter), z. B. mittlere Antwortzeiten, Durchsatzkennzahlen, Auslastungsgrade von Prozessen oder Füllgrade von Speicherbereichen

- ▶ Textinformationen, z. B. Fehlermeldungen (eine Fehlermeldung »Dokumentbearbeitung von Dokument X wurde abgebrochen« wäre also ein Indikator für eine schlechte Performance eines Systems). Performance meint in diesem Sinne nicht nur die reine Laufzeitperformance, sondern bewertet auch die Verfügbarkeit und Fehlersituationen.

Die Aufgabe der Lösungsüberwachung ist , Performanceindikatoren zu erfassen, zu speichern, zu aggregieren und zu bewerten. Der zentrale Überwachungsmonitor der SAP integriert Performanceindikatoren aus Duzenden von »Expertenmonitoren« und führt diese in Monitoringbäumen zusammen. Diese Expertenmonitore dienen dem Überwachungsmonitor als Data Supplier. Der Vorteil des zentralen Überwachungsmonitors ist, dass Sie die Expertenmonitore nur in besonderen Ausnahmesituationen verwenden müssen, nicht aber für die normale Überwachung von Komponenten. Im zentralen Überwachungsmonitor sind die zu den Indikatoren gehörenden Expertenmonitore als so genannte Analysemethoden hinterlegt. Sie können aus dem Überwachungsmonitor direkt in den entsprechenden Expertenmonitor springen (auch mit Anmeldung an einem entfernten System).

Der Überwachungsmonitor generiert automatisch Alerts, wenn ein Indikator Schwellenwerte über- oder unterschreitet. Für Schwellenwerte werden Standards ausgeliefert, die individuell angepasst werden können.

Nicht alle Performanceindikatoren haben dieselbe Bedeutung, wenn es darum geht, zu bewerten, ob Ihre mySAP.com-Lösung performant läuft. Einige Indikatoren werden sich mit wachsender Erfahrung als besonders wichtig herausstellen. Diese bezeichnen wir als Key Performance Indikatoren (KPIs). Ein Ziel beim Aufbau eines Überwachungskonzeptes ist daher, die für Sie relevanten KPIs aus der Vielzahl der Informationen herauszufiltern. Einige Anleitungen wollen wir dazu geben.

### **Mit dem Überwachungsmonitor arbeiten**

Den Überwachungsmonitor rufen Sie wie folgt auf:

1. **Werkzeuge CCMS • Steuerung/Monitoring • Alert-Monitor**, oder verwenden Sie den Transaktionscode RZ20. Das System zeigt die CCMS-Monitorsammlungen an.
2. Expandieren Sie eine der Monitorsammlungen, indem Sie den Cursor auf die »CCMS-Monitorsammlungen« positionieren und dann **Bearbeiten • Baum expandieren** wählen.