

# **Total Quality Monitoring: Oracle SOA Suite @ T-Systems**

**Dipl.-Inform. Arnd Scharpegge M.A.  
Lynx-Consulting GmbH  
Johanniskirchplatz 6, 33615 Bielefeld**

**Dipl.-Wi.-Inform. Andreas Makiola  
T-Systems International GmbH  
Lübecker Str. 2, 39124 Magdeburg**

## **Schlüsselworte:**

Oracle SOA Suite, Monitoring, Prozesse, Überwachung, Automatisierung

## **Einleitung**

Ziel dieses Vortrages ist es, das umfassende Monitoring von Anwendungen darzustellen, die die Oracle SOA Suite nutzen und von T-Systems in Magdeburg betreut werden.

Es werden Informationen zu Zielen und Vorgaben des Monitorings gegeben, die resultierende Strategie wird erläutert und, auch vor dem Hintergrund der Server- und Tool-Landschaft bei den Anwendungen, der Prozeß der Auswahl eines passenden Monitoring-Tools beleuchtet. Danach werden Details zur Analyse der zu überwachenden Prozesse angeführt, die Eignung zum Erstellen von Statistiken und Reports sowie das Umsetzen neuer Monitoring-Prozesse betrachtet. Am Ende gibt es Hinweise zu effizienten Umsetzungen in Projekten sowie einen abschließenden Vergleich der Situation vor und nach der Einführung des Total Quality Monitorings.

## **Ziele des Total Quality Monitorings**

Folgende Ziele wurden während der Konzeptionsphase zum Einsatz eines Monitoring-Tools festgelegt:

- Sicherheit beim Erkennen von Störungen
- Möglichkeit, Störungen vorher absehen zu können
- automatisiertes Reporting
- geringe Ressourcenbindung im laufenden Betrieb

Diese Ziele ergaben sich aus den Anforderungen des täglichen Betriebes, in dem wenig Zeit für das manuelle und permanente Monitoren der SOA Suites und anderer Anwendungsteile aufgewendet werden konnte.

## **Vorgaben für das Total Quality Monitoring**

Diese Ziele des Monitorings münden direkt in Vorgaben für das Monitoring-Tool:

- keine Software-Installationen auf den zu überwachenden Systemen
- kein Monitoring durch Beobachten, sondern:
  - eventgesteuertes, direktes Benachrichtigen der Mitarbeiter (z.B. Email, SMS)
  - Möglichkeit zum Versenden von Reports muß vorhanden sein
- granulare Zugriffssteuerung für Webseiten

Insgesamt soll das Tool durch die Mitarbeiter im Betrieb zu warten sein, ohne dass permanent der Einsatz von Entwicklern notwendig sein muß.

## **Monitoring Strategie**

An dieser Stelle soll kurz die Strategie skizziert werden, die aus dem zuvor Beschriebenen abgeleitet wurde:

- maschinelles Monitoren aller Systeme, die zu einer SOA Suite Installation gehören
  - SOA Suite selbst (Application Server)
  - Datenbanken
  - Schnittstellen
  - evtl. direktes Monitoren dritter Systeme, die (Teil-) Aufgaben für SOA Prozesse erledigen
- Erstellen von Routinen zum Erfassen von Zuständen, Bewerten von Parametern, etc.
- Einsatz eines Monitoring-Tools, das flexibel in die bestehende Server- und Tool-Landschaft integriert werden kann

## **Server-Landschaft**

Einer der Hauptaspekte des Einsatzes einer service-orientierten Architektur ist das Einbinden vielfältiger Systeme über ihre Schnittstellen, ohne auf ihre Besonderheiten hinsichtlich Betriebssystem, eingesetzter Software, etc. Rücksicht nehmen zu müssen. Idealerweise können alle anzusprechenden Systeme über entsprechend konfigurierte Web-Services erreicht werden.

Vor diesem Hintergrund ist es nicht wichtig, welche Ausprägungen einzelne Server bzw. Systeme haben.

Bei den von der T-Systems in Magdeburg betreuten SOA-Anwendungen handelt es sich insgesamt um eine sehr heterogene Server-Landschaft, die über eine Vielzahl von Betriebssystemen (unterschiedliche Unix-Systeme, verschiedene Windows Server-Varianten, tw. in differierenden Versionen, Patch-Releases, Service Packs, etc.) verfügt.

Diese Heterogenität ergab sich aus einer aufgabenspezifischen Beschaffung neuer Server, die für ihre jeweilige Aufgabe optimal konfiguriert wurden. Zu diesen Aufgaben zählen u.a.:

- Vorhalten von Informationen in Datenbanken
- Veröffentlichen von Services über Application Server
- Versenden und Empfangen von Emails
- Versenden und Empfangen von Faxen
- Versenden von SMSen in Mobil- und Festnetze
- Scannen von Dokumenten
- Archivieren von Dokumenten aller Art

## **Tool-Landschaft**

Neben der heterogenen Server-Landschaft der Anwendungen stand vor dem Einsatz des Total Quality Monitorings auch eine Vielzahl von Tools zur Verfügung, die zum Überwachen bestimmter Teilaspekte dienten, z.B.:

- Oracle Enterprise Manager für Datenbanken
- Oracle Enterprise Manager für die SOA Suite
- Oracle SOA Suite BPEL Console
- Oracle dms-Tool (Performance-Messungen des Application Servers)
- Tools zum Abfragen von Datenbanken
- Konsolen für Serverabfragen

Da diese aber jeweils ihre Informationen nur auf händische Abfragen zur Verfügung stellen, sind sie zwar sehr nützlich, binden aber im täglichen Einsatz viele Ressourcen.

## Auswahl des Monitoring-Systems

An ein geeignetes Monitoring-Tool wurden folgende fachliche Kriterien gestellt:

- bewährte Technologie
- Open Source
- plattformneutral
- Java-basierend
  - über die Java-API kann auf Funktionen der Oracle SOA Suite zugegriffen werden
  - Teilaufgaben können durch Einbinden vorhandener Tools einfach und schnell gelöst werden
- verbreitete Programmiersprache (kurze Einarbeitung neuer Entwickler)
- Überwachungsprojekte müssen einfach konfigurierbar sein (auch ohne Entwickler zu pflegen)

Die Entscheidung fiel am Ende für Hudson ( [www.hudson-ci.org](http://www.hudson-ci.org) ). Dieses Tool hat seinen Ursprung in der Continuous Integration (Software-Entwicklung), also dem automatisierten Pflegen und Testen von Quellcode. Überzeugt hat hier vor allem, neben dem Erfüllen aller oben genannten Kriterien, die Bedienbarkeit und einfache Adaption zum Einsatz als Monitoring-Tool.

## Analyse zu überwachender Prozesse

Dieser Abschnitt beschreibt den zentralen Aspekt des Total Quality Monitorings der Oracle SOA Suite-Anwendungen bei der T-Systems. Um es noch einmal zu betonen – Einzelaspekte sollten nicht isoliert betrachtet, überwacht und bewertet werden, sie müssen als Gesamtheit gesehen und entsprechend behandelt werden. Wenn ein vermeintlich kleiner Baustein unregelmäßig läuft, kann dies schon ein Indiz für ein anstehendes größeres Problem sein. Hier ist zum einen die Erfahrung der Mitarbeiter im Betrieb gefragt, zum anderen müssen so viele Aspekte wie möglich abgebildet und miteinander verknüpft werden.

Ein Beispiel:

Eine Messaging Queue (Queue) ist als aktiv erreichbar und befindet sich somit im normalen, störungsfreien Arbeitsmodus. Die abnehmende Anwendung aber hat ein Problem, so daß sie keine weiteren Nachrichten aus der Queue abholt. Wenn man nun die Anzahl neuer Instanzen der abnehmenden Anwendung in ein Verhältnis zum (aktuell ansteigenden) Füllstand der Queue setzt und dies zeitnah überwacht, drängt sich nicht nur schnell auf, daß es einen Fehler in der Strecke gibt, sondern kann die Ursache schnell lokalisiert werden – und somit kann die Ursachenforschung stark fokussiert beginnen.

Zu Beginn der Arbeit muß daher eine ausführliche Analyse aller Geschäftsprozesse erfolgen, die über die SOA Suite abgebildet werden. Dabei sollten nicht nur die technischen Aspekte z.B. eines BPEL-Prozesses berücksichtigt werden, sondern möglichst alle Rahmenbedingungen, die diesen Prozess betreffen:

- Überwachen der gesamten Prozeßkette
  - eingehende Schnittstellen
  - Verarbeitung im Oracle BPEL Process Manager
  - Verarbeitung von Prozeßschritten in anderen Systemen
  - Schnittstellen zu diesen Systemen
  - ausgehende Schnittstellen
- Überwachen beteiligter Systeme bzw. Schnittstellen
  - Datenbanken
  - Message Queueing Systeme
  - Email-Server
  - FTP-Server
  - Fax-Server
  - Scan-Server
  - Druck-Server bzw. Druck-Straßen
  - (Nicht-) Laufen & Anstiege kontrollieren
- Überwachen beteiligter technischer Prozesse
  - OS-Prozesse
    - Message Queueing Systeme
    - Java-Prozesse
    - OPMN-Prozesse
  - Platz für Daten
    - Festplattenplatz
    - Platz für DB-ArchiveLogs

- Überwachen fachlicher Aspekte
  - Verarbeiten ausreichender Instanzen (im Vergleich zu einem Standard-Zeitraum)
    - falls nicht: Wie kann auf einen Fehler geschlossen werden? Eine Lösung muß abgebildet werden.
  - Messen der Reaktionsgeschwindigkeiten von Nutzer-Aktionen bzw. Abläufen, z.B.:
    - Anmelden in einer Web-Anwendung
    - Durchführen prototypischer Nutzeraktionen
    - Abmelden von der Anwendung
- Automatisieren von Reaktionen
  - z.B. SOA Suite:
    - Manuelles Recovery alter Prozeßinstanzen via Java-API anstoßen
    - Manuelles Recovery alter Callbacks via Java-API anstoßen
  - Umsetzen anwendungsspezifischer Wiederanlaufmechanismen

## Statistiken & Reports

Neben der permanenten Überwachung sollte ein Monitoring-Tool ebenfalls die Möglichkeit bieten, regelmäßig Auswertungen zur Verfügung zu stellen:

- einige Beispiele:
  - zeitnahes Überwachen von Lizenzverbräuchen (z.B. Scan-Vorgänge)
  - Erstellen von Monats-, Wochen- und Tagesstatistiken für Mitarbeiter
- viele Ausgabeformate wählbar:
  - Excel
  - HTML
  - PDF
- Versand per Email oder Veröffentlichung als Webseite
- via Java sind vielfältige Verarbeitungsformen nach Entwicklervorlieben einsetzbar:
  - XSL-T
  - programmatische Entwicklung gegen Java-APIs für z.B. PDFs, Excel- oder Word-Dateien
  - Groovy-Techniken zum einfachen Erstellen von XML-Dokumenten für die Weiterverarbeitung

## **Erstellen von Hudson-Projekten**

Sehr praktisch und als in der Praxis vorteilhaft hat sich die Verknüpfung selbstgeschriebener Logik in Verbindung mit den Konfigurations-Möglichkeiten von Hudson erwiesen. Somit werden die Projekt-Vorgaben eindeutig erfüllt:

- Projekte werden als formale Hüllen erstellt
- der zugrundeliegende Code kann für weitere Projekte wiederverwendet werden
- Steuern konkreter Projekte via Konfiguration
  - Hudson-eigene Konfigurationen (Benachrichtigungen, CronJobs)
  - Konfiguration innerhalb der Projekte mittels Dateien (\*.properties, \*.xml)
- nachvollziehbare Programmierung via ausführlicher Dokumentation
  - Aufgabe und Ablauf: in Hudson
  - konkrete Umsetzung: im Code
- schnelles Ausrollen vorhandener Code-Projekte in weitere Umgebungen bzw. neue Geschäftsprozesse durch Kopieren vorhandener Projekte und reines Anpassen der Konfiguration

## **Voraussetzungen für effiziente Projektumsetzungen**

Damit ein umfassendes Monitoring möglichst schnell umgesetzt werden kann, ist es vorteilhaft, wenn nachfolgende Bedingungen erfüllt werden:

- homogene (Teil-) Serverlandschaften, z.B.:
  - Betriebssystem
  - Aufbau Dateisystem
- Umsetzung einheitlicher Vorgaben hinsichtlich
  - Software-Installation
  - Konventionen (Benamung, Aufruf von Skripten)
  - eingerichtete Cronjobs
  - eingerichtete User
- Schulung der Mitarbeiter in den Grundzügen von Java/Groovy
- Schulung im Umgang mit Hudson

Dies sind zwar ganz normale Anforderungen gerade aus Sicht eines Betriebes, aber sie sollen hier noch einmal explizit als Erfolgsfaktoren genannt werden.

## **Verbesserungen im Vergleich zu früher**

Abschließend sollen die Vor- und Nachteile des Total Quality Monitoring-Ansatzes kurz beleuchtet werden.

Vorteile:

- schnellere Umsetzung neuer Überwachungsprojekte
- umfassende Überwachung der gesamten Prozeßabläufe
- kurze Reaktionszeiten nach dem Auftreten von Fehlern
- tw. Erkennen aufziehender Fehler vor deren eigentlichem Auftreten
- effizientere Nutzung der Ressourcen im Betrieb

Nachteile:

- bisher keine zu erkennen ...

## **Kontaktadressen:**

### **Dipl.-Inform. Arnd Scharpegge M.A.**

Lynx-Consulting GmbH  
Johanniskirchplatz 6  
D-33615 Bielefeld

Telefon: +49 (0) 521-52 47 0  
Fax: +49 (0) 521-52 47 250  
E-Mail [arnd.scharpegge@lynx.de](mailto:arnd.scharpegge@lynx.de)  
Internet: [www.lynx.de](http://www.lynx.de)

### **Dipl.-Wi.-Inform. Andreas Makiola**

T-Systems International GmbH  
Lübecker Str. 2  
D-39124 Magdeburg

Telefon: +49 (0) 391-533 10 99  
Fax: +49 (0) 391-53 47 24 27  
E-Mail [andreas.makiola@t-systems.com](mailto:andreas.makiola@t-systems.com)  
Internet: [www.t-systems.de](http://www.t-systems.de)