

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals  
in den Bereichen Softwareentwicklung,  
Internettechnologie und IT-Management aktuell  
und kompetent relevantes Fachwissen über  
Technologien und Produkte zur Entwicklung  
und Anwendung moderner Informationstechnologien.

Christof Schmalenbach

# Performance- management

für serviceorientierte  
JAVA-Anwendungen

Werkzeug- und Methoden-  
unterstützung im Spannungsfeld  
von Entwicklung und Betrieb

Mit 50 Abbildungen und 16 Tabellen

 Springer

Christof Schmalenbach

IBM Deutschland GmbH  
Karl-Arnold-Platz 1a  
D-40474 Düsseldorf  
cschmale@de.ibm.com

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISSN 1439-5428

ISBN 978-3-540-36631-7 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist ein Unternehmen von Springer Science+Business Media  
[springer.de](http://springer.de)

© Springer-Verlag Berlin Heidelberg 2007

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz: ptp, Berlin

Herstellung: LE-TeX, Jelonek, Schmidt & Vöckler GbR, Leipzig

Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg

Gedruckt auf säurefreiem Papier 33/3180 YL - 5 4 3 2 1 0

*Für Kerstin und unsere Tochter Laura*

# Vorwort

Der Markt für Systemsoftware, welche die Entwicklung und den Betrieb von performanten Anwendungen zu unterstützen verspricht, ist nahezu unübersichtlich geworden. Und das nicht ohne Grund. Denn mit dem wachsenden Grad der Verteilung von Komponenten auf physisch und geographisch entfernten Servern und Netzwerken werden die bedarfsgerechte Planung von Kapazitäten und die Gewährleistung zugesagter Qualitätsmerkmale im Hinblick auf Antwortzeiten und Durchsatz zunehmend schwieriger. Dies kann bereits beobachtet werden, seit durch Java die Programmierung von Anwendungen, die über Prozessgrenzen hinweg miteinander kommunizieren, deutlich vereinfacht wurde und J2EE als eine der Schlüsseltechnologien zur Entwicklung kritischer Geschäftsanwendungen eingesetzt wird. Mit der Verbreitung serviceorientierter Technologien wird sich dieser Trend noch verstärken, da die bisher noch häufig vorzufindende enge Bindung zwischen einzelnen Applikationen durch solche Architekturen zurückgedrängt werden wird, in denen Services in vielfältiger Weise zu Anwendungen gebündelt werden können (composite applications).

Damit Entwickler und Anwendungsarchitekten bereits in frühen Entwicklungsphasen mögliche Performancerisiken erkennen oder Systemadministratoren das Laufzeitverhalten produktiver Anwendungen kontrollieren können, sind selbstverständlich auch angemessene Werkzeuge erforderlich.

Allzu oft besteht aber die Neigung, den Performanceherausforderungen durch einen inflationären Einsatz von Systemsoftware und Profilingtools begegnen zu wollen.

Häufig wird auch gerne die Aussage formuliert, dass Last- und Performancetest nur dann eine sinnvolle Qualitätssicherung darstellen, wenn sie unter dem Einsatz produktionsidentischer Hardware erfolgen. In den Projekten, in denen der Autor bislang als Performancearchitekt hat wirken können, war es aus Zeit- oder Kostengründen fast nie möglich, diese Voraussetzungen zu erfüllen. Und je weiter die Verwendung von Services über Unternehmensgrenzen hinweg in einer Vielzahl von Anwendungskonstellationen fortschreitet und Infrastrukturen virtualisiert werden, desto weniger kann von *der* Produktionsumgebung überhaupt noch gesprochen werden.

Serviceorientierte Architekturen werden nicht zum Selbstzweck betrieben. Wechselnde Anforderungen an das Business sollen durch die IT flexibel und schnell umgesetzt werden können. Mit kürzeren Produktionszyklen schwinden aber auch die Zeiten, die für qualitätssichernde Maßnahmen im Allgemeinen und das Performancemanagement im Besonderen verbleiben. Zukünftig wird es daher immer mehr darauf ankommen, auch unter vagen Annahmen und mit dem Einsatz geringer Ressourcen, Performanceprobleme vermeiden oder deren Ursachen schnell identifizieren und beheben zu können.

Die Frage, wie und mit welchen Mitteln dies gelingen kann, ist Gegenstand des vorliegenden Buches. Es wendet sich vornehmlich an Vertreter jener beiden IT-Bereiche, die auf technischer Ebene für die Performanz von Anwendungen verantwortlich sind: Auf der Entwicklungsseite die Anwendungsarchitekten und auf Betriebsseite die Systemadministratoren.

Es sind im Wesentlichen die drei folgenden Leitgedanken, die zum Verfassen des Buches motiviert haben und den Rahmen festlegen, in dem es sich bewegt:

1. Planung und Kontrolle sind auch unter restriktiven Bedingungen und Unsicherheit möglich. Unternehmen und Staaten beweisen dies immer wieder, wenn sie umfangreiche Projekte durchführen. Und noch nie hat jemand die Forderung erhoben, ein Staudamm müsse als Prototyp gebaut und unter realen Bedingungen getestet werden, bevor an einen produktiven Betrieb überhaupt zu denken sei.
2. Es gibt einfache Mittel und Verfahren, mit denen ein großer Teil der Planungs- und Kontrollaufgaben gemeistert werden kann. Einfach soll nicht im Sinne von intellektuell wenig herausfordernd, sondern als anspruchslos im Hinblick auf die notwendigen technischen Ressourcen verstanden werden. Bei diesen Mittel handelt es sich u.a. um das klassische Werkzeug schlechthin – die Mathematik.
3. Viele große Open-Source-Projekte sind im Umfeld der J2EE-Entwicklung außerordentlich populär. Hingegen gibt es zahlreiche kleinere Entwicklungsinitiativen, die wenig bekannt sind, aber durchaus nennenswerte Beiträge für das Performancemanagement leisten können. Von beiden Varianten wird das Buch reichlich Gebrauch machen.

## **Die Struktur des Buches**

Im ersten Kapitel wird anhand eines fiktiven Szenarios verdeutlicht, warum Performancemanagement oftmals so schwierig ist und dass diese

Schwierigkeiten nicht nur technologisch bedingt sind, sondern sich auch aufgrund des unterschiedlichen Selbstverständnisses verschiedener IT-Kulturen ergeben.

Das zweite Kapitel widmet sich dem Entwicklungsprozess und dem Aufbau eines Performanceteams. Die dort vertretenen Rollen und die von den Mitgliedern zu leistenden Aufgaben werden erläutert. Ausführlich wird auf die Gestaltung von Performancetests eingegangen.

Im dritten und vierten Kapitel wird eine Einführung in zwei Theorien gegeben, die für das Verständnis des Buches unabdingbar sind, die aber in der IT-Ausbildung üblicherweise allenfalls am Rande behandelt werden. In Kapitel 3 handelt es sich um quantitative Methoden und die Verwendung der Warteschlangentheorie und in Kapitel 4 um die diskrete Ereignis-Simulation.

Das fünfte Kapitel befasst sich mit den wichtigen Themen der Dokumentation von Performanceanforderungen und der vertraglich garantierten Serviceeigenschaften.

Kapitel 6 ist als Fortsetzung von Kapitel 3 zu verstehen. Hier wird am Beispiel einer Software, welche IBM zur Performance- und Kapazitätsplanung einsetzt, die praktische Verwendung der quantitativen Methoden und der aus diesen entwickelten Algorithmen erläutert.

In Kapitel 7 wird die Simulation von Geschäftsprozessen an einem fiktiven, aber praxisbezogenen Beispiel aus der Versicherungsindustrie vorgestellt. Auch Geschäftsprozesse müssen gewissen Performancekriterien genügen. Selbst wenn diese weniger technisch als fachlich definiert sind, so ist doch die Methodik an vielen Stellen vergleichbar.

Sowohl bei der Durchführung von Performancetests als auch im produktiven Betrieb besteht das Bedürfnis, die Laufzeiten bestimmter Methoden oder der von diesen geforderten Rechenleistungen zu ermitteln. Hierfür gibt es eine Reihe kommerzieller Produkte. Im Umfeld von Java nutzen diese häufig ein von der Java Virtual Machine unterstütztes API. In dieses API führt Kapitel 8 ein. Dies geschieht anhand der Entwicklung eines Performancemonitors, den der Autor mitentwickelt und bereits in mehreren Projekten eingesetzt hat.

Wer sich für den Einsatz dieses Monitors oder anderer frei verfügbarer Software zur Erhebung von Performancemetriken entscheidet, sieht sich mit dem Problem konfrontiert, geeignete Darstellungsformen für die rohen Performancedaten festzulegen und Diagramme zu erstellen. Mit dem Thema der Datenvisualisierung setzt sich Kapitel 9 auseinander. In Kapitel 10 werden alle bislang vorgestellten Techniken am Beispiel eines Webservices im Zusammenhang genutzt. Die Performance dieses Webservices wird durch quantitative Modelle prognostiziert und die Qualität dieser Prognose durch einen Lasttest überprüft. Dann wird erneut das Thema der diskreten Ereignissimulation aufgegriffen, um das reale Verhalten des Webservices

in einer wenige Sekunden dauernden Simulation nachzustellen. Auch wenn Simulationsverfahren heute nur eine bescheidene Rolle bei der Planung von komplexen IT-Infrastrukturen spielen und in diesem Buch auch nur anhand eines überschaubaren Beispiels verwendet werden können, steckt in dieser Methodik nach Überzeugung des Autors ein enormes Potential.

## Danksagungen

Nachdem mein Entschluss feststand, meine Erfahrungen und Kenntnisse, die ich in den vergangenen Jahren in der Rolle als Performancearchitekt sammeln durfte, in einem Buch darzulegen, hatte ich eine relativ konkrete Vorstellung davon, welche Themen ich darin aufnehmen wollte.

Und obgleich ich von diesem Themenkatalog nur geringfügig abgewichen bin, musste ich mit Erstaunen feststellen, dass zu vielen Inhalten, die ich zu beherrschen glaubte, doch noch ein intensiver Informationsaustausch notwendig ist. Wie ich erfahren habe, ist das Verfassen eines Buches mit vielen administrativen Tätigkeiten verbunden. Ohne entsprechende Hilfe hätte ich diese Aufgaben wohl kaum meistern können. Daher möchte ich allen Dank aussprechen, die mir in der unterschiedlichsten Art und Weise ihre Unterstützung gewährt haben.

Dem HiPODS (High Performance On Demand Solutions)-Team der IBM Software Strategy Group und hier ganz persönlich dem Kollegen Noshir Wadia für die Genehmigung, ein Kapitel über den On Demand Performance Advisor aufnehmen zu dürfen, und die Bereitschaft, dieses Kapitel einem Review zu unterziehen.

Dem Gründungsdirektor des Human Computer Interaction Laboratory, Professor Shneiderman, für die Nutzungsrechte an der Software Treemap 4.1 im Kapitel Visualisierungstechniken.

Herrn Professor Page vom Department Informatik der Universität Hamburg für die Klärung offener Fragen bei der Benutzung des von ihm und seinen Mitarbeitern betreuten Simulationsframeworks Desmo-J.

In den vergangenen drei Jahren habe ich bei der DEVK-Versicherung in Köln einige Performanceaktivitäten begleitet. Herrn Geisel vom Vorstand der DEVK möchte ich für die Bereitschaft danken, einige der dort bearbeiteten Themen im Buch aufnehmen zu dürfen. Die in Kapitel 9 aufgeworfenen Fragen im Hinblick auf Internetanwendungen unter Verwendung von Mobilfunk beruhen auf diesen Tätigkeiten. Ohne die angenehme Arbeitsatmosphäre in der von Frau Bauer verantworteten Anwendungsentwicklung und dem von Michael Porschen betreuten Architekturprojekt wären diese Untersuchungen so nicht möglich gewesen. An beide geht stellvertretend für das gesamte DEVK-Team mein herzlichster Dank.

Natürlich haben auch zahlreiche Einzelpersonen wichtige Beiträge zu dem Buch geleistet. Hier gilt mein erster Dank den Kollegen Norbert Pich, Dr. Joachim Rawolle und Gunther Schichl. Die Idee zu dem in Kapitel 7 simulierten Geschäftsprozess stammt von ihnen und sie haben mir mit Geduld meine Fragen zum versicherungsfachlichen Hintergrund beantwortet.

Mit Christian Höfig habe ich die erste Version des in Kapitel 8 vorgestellten Java-Monitors implementiert und 2004 im Dr. Dobb's Journal veröffentlicht. Mit Freude denke ich an unsere Java- und C-Entwicklung in den Berliner Biergärten zurück. Ohne seine fundierten Unix- und GCC-Kenntnisse hätte der Monitor wohl niemals „über den Tellerrand von Windows hinausgeschaut“. In diesem Zusammenhang geht auch mein Dank an Jonathan Erickson, den Chefredakteur des DDJ. Er hat meiner Bitte, den Artikel für das Buch in einem größeren Zusammenhang darstellen zu dürfen, spontan und unbürokratisch entsprochen.

Dave Jewell hat mich mit umfangreichem Informationsmaterial von IBM zum Thema Performance ausgestattet. Damit war eine bessere Beurteilung des Einsatzgebietes von Simulationstechniken im Bereich des Performancemanagements möglich.

Aus den Büchern von Daniel Menasce und denen von Neil Gunther habe ich ein sehr gutes Verständnis für formale Methoden der Kapazitätsplanung gewinnen können. Beide waren mir auch hinsichtlich rechtlicher Fragen bei der Aufnahme von Zitaten im Buch behilflich.

Herr Professor Dueck von IBM war der erste, dem ich meine Buchidee unterbreitet habe. Er hat mir meine anfängliche Scheu genommen und damit quasi den Startschuss zur Veröffentlichung gegeben.

Mit Dr. Volker Mull habe ich gemeinsam so manche Performance-schlacht geschlagen, manche gewonnen, einige auch verloren. Ich hoffe, wir halten an unserem Plan fest und verfolgen das Thema Simulation weiter.

Rat und ständiger Zuspruch meiner beiden Kollegen und Freunde Cemal Cömert und Steffen Fischer waren mir über Monate hinweg ein wichtiger Ansporn. Wenn mir nachts um zwei die Augen fast zufielen, so hatte ich doch gute Chancen, einen von ihnen im internen IBM-Chat anzutreffen und mich durch ihren Witz wieder auf die Beine bringen zu lassen. By the way: Wer von euch beiden plant eigentlich unsere nächste Eifelwanderung? Zeit hätte ich ja nun wieder ☺.

Frau Riepl von LE-TeX in Leipzig hat sehr bereitwillig alle meine laienhaften Fragen rund um die Gestaltung von Grafiken, die Seitengestaltung und das Publishing beantwortet. Auch hierfür gebührt mein herzlichster Dank.

Durch Frau Glaunsinger und Herrn Engesser vom Springer-Verlag habe ich eine hervorragende Betreuung erhalten und eine äußerst angenehme Zusammenarbeit erfahren. Ein ganz besonderer Dank gebührt ihnen auch

dafür, dass sie *meine* Performanceengpässe und die daraus resultierenden Terminverschiebungen so geduldig mitgetragen und mich in Zeiten der Frustration immer wieder aufgemuntert haben.

Das Verfassen des Buches war ein äußerst anstrengender Marathonlauf. Als mir noch auf dem letzten Kilometer die Achillessehne riss, hat mir meine Schwester Inge geholfen, auf einem Bein humpelnd die Ziellinie zu überqueren. Ich bin ihr dafür über alle Maßen dankbar.

Laura, Du hast mich mit Deinen Übersetzungshilfen bei den notwendigen Absprachen mit meinen Kollegen in den Vereinigten Staaten auf ganz praktische Weise unterstützt. Das war für mich von großer Bedeutung. Doch auch aus anderen Gründen würde es dieses Buch ohne Dich nicht geben. Ich gewinne so viel Kraft aus Deiner Kreativität, Deinem intelligenten Humor und Deiner Begeisterungsfähigkeit für die wirklich wichtigen Dinge im Leben.

Kerstin, siebzehn großartige Jahre mit Dir waren für mein Leben ein optimales Performancetuning. Du bist die Ghostwriterin dieses Buches.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.2	Ein Performancemeeting	1
1.3	Das fachliche und technische Umfeld	4
1.4	Performanceaspekte	5
1.5	Neue Herausforderungen	8
1.6	Performance als interdisziplinäre Herausforderung	10
<b>2</b>	<b>Performance und Entwicklungsprozess</b>	<b>13</b>
2.1	Motivation	13
2.2	Rollen und Aufgaben	14
2.2.1	Der Facharchitekt	14
2.2.2	Der IT-Architekt	15
2.2.3	Der Entwickler	16
2.2.4	Der Systemadministrator	17
2.3	Aufbau eines Performanceteams	18
2.3.1	Organisationsstruktur des Performanceteams	19
2.3.2	Kickoff Workshop des Performanceteams	22
2.4	Performance- und Lasttests	32
2.4.1	Der Testplan	33
2.4.2	Testvorbereitung	35
2.4.3	Testdurchführung	40
2.4.4	Analyse und Testberichte	40
2.5	Zusammenfassung	41

<b>3</b>	<b>Quantitative Methoden .....</b>	<b>43</b>
3.1	Motivation.....	43
3.2	Ressourcen und Warteschlangen .....	45
3.2.1	Begriffsfestlegungen und Definitionen.....	46
3.2.2	Auslastung und das Utilization Law .....	47
3.2.3	Das Forced Flow Law .....	50
3.2.4	Die Queueing-Formel von Little.....	50
3.2.5	Response Time Law.....	51
3.3	Mean Value Analysis.....	53
3.3.1	Das Arrival-Theorem .....	55
3.3.2	Implementierung des MVA-Algorithmus .....	56
3.3.3	Anwendungsmöglichkeiten der Mean Value Analysis .....	59
3.4	Zusammenfassung und Ausblick .....	62
<b>4</b>	<b>Simulationstechniken .....</b>	<b>65</b>
4.1	Motivation.....	65
4.1.1	Kleine Ursache – große Wirkung.....	66
4.2	Rechnergestützte Simulationsmodelle .....	67
4.2.1	Simulation als Alternative zu Testverfahren.....	68
4.2.2	Simulation als Alternative zu analytischen Verfahren.....	68
4.3	Begriffsbildung und Definitionen .....	69
4.3.1	Der Systembegriff.....	70
4.3.2	Der Modellbegriff .....	70
4.3.3	Datenstrukturen.....	71
4.4	Diskrete Ereignissimulation.....	72
4.4.1	Entitäten, Ressourcen und Attribute .....	72
4.4.2	Ereignisse, Aktivitäten und Prozesse .....	73
4.4.3	Algorithmische Grundprinzipien einer Simulation.....	76
4.5	Zusammenfassung.....	80
<b>5</b>	<b>Dokumentation der Performanceanforderungen .....</b>	<b>83</b>
5.1	Motivation.....	83
5.2	Nicht-funktionale Anforderungen.....	85
5.2.1	Die Dualität von Anforderungen.....	85
5.2.2	Die Bedeutung der nicht-funktionalen Anforderungen.....	86
5.2.3	Anforderungsdokumentation als Balanceakt .....	86
5.3	Die zentralen Performanceanforderungen .....	87
5.3.1	Mengengerüste und Volumen .....	87
5.3.2	Antwortzeiten aus Benutzersicht .....	89
5.3.3	Durchsatz .....	95
5.3.4	Verfügbarkeit .....	95
5.3.5	Skalierbarkeit .....	100
5.4	Service Level Agreements .....	102

---

5.4.1	Performancekontrollen etablieren .....	103
5.4.2	SLA und SOA .....	104
<b>6</b>	<b>Werkzeugunterstützte Kapazitätsplanung.....</b>	<b>107</b>
6.1	Zielsetzung und Motivation .....	107
6.2	Die Notwendigkeit von Modellerweiterungen.....	108
6.2.1	High Performance On Demand Solutions.....	109
6.3	On Demand Performance Advisor.....	109
6.3.1	What-If-Analysen und Anwendungsmodi .....	111
6.3.2	Verfahrensweise.....	111
6.3.3	MVA-Erweiterungen .....	116
6.4	Kapazitätsplanung einer Versicherungsanwendung .....	117
6.4.1	Antragserfassung und Vertragsverwaltung .....	117
6.4.2	Anwendungsfälle und Lastszenarien .....	118
6.4.3	Ressourcenbedarfsermittlung im Lasttest .....	119
6.4.4	Performanceziele.....	120
6.4.5	Auswertung des Modells.....	122
6.5	Zusammenfassung und Ausblick .....	125
6.5.1	Restriktionen .....	125
6.5.2	Ergänzende und alternative Werkzeuge.....	126
<b>7</b>	<b>Simulation von Geschäftsprozessen.....</b>	<b>129</b>
7.1	Motivation.....	129
7.1.1	Modelle als Orientierungsrahmen.....	129
7.1.2	Analogien nutzen .....	130
7.1.3	Performanceanforderungen identifizieren.....	130
7.2	Prozessmanagement und Simulation .....	130
7.2.1	Analyse und Modellbildung.....	131
7.2.2	Entwicklung und Deployment .....	132
7.2.3	Prozessmonitoring.....	132
7.3	WebSphere Business Modeler .....	132
7.3.1	Aktivitäten.....	133
7.3.2	Business Items .....	133
7.3.3	Ressourcen und Rollen .....	134
7.4	Fallbeispiel Schadensmanagement .....	134
7.4.1	Automatisierte Belegverarbeitung .....	135
7.4.2	Der fachliche Hintergrund .....	136
7.4.3	Die fachlichen Objekte .....	137
7.4.4	Ressourcen und Rollen .....	138
7.4.5	Aktivitäten, Services und Entscheidungspfade.....	138
7.5	Ziele der Prozessanalyse .....	141

7.6	Simulationsergebnisse.....	143
7.6.1	Ein Arbeitstag in der Schadensabteilung .....	144
7.6.2	What-If-Analysen .....	147
7.7	Zusammenfassung und Ausblick .....	151
7.7.1	Planung von Simulationsprojekten .....	152
<b>8</b>	<b>Java Monitoring und Profiling .....</b>	<b>155</b>
8.1	Zielsetzung und Motivation .....	155
8.2	Die Qual der Wahl .....	156
8.2.1	Projektübergreifendes Produktmanagement .....	157
8.2.2	Performancetests durch Werkzeugevaluierung.....	159
8.2.3	Anwendungsspezifische Messverfahren .....	160
8.2.4	Einschränkungen und Nachteile.....	161
8.3	Ein Java Monitor im Eigenbau .....	163
8.3.1	Leistungsbeschreibung des Performancemonitors .....	164
8.3.2	Profiling versus Monitoring .....	166
8.3.3	Werkzeugportfolio .....	168
8.3.4	Die Unterstützung durch Java .....	168
8.4	Java Virtual Machine Profiler Interface.....	172
8.4.1	Klassifikation der JVMPI-Ereignisse.....	173
8.4.2	Ein erster prototypischer Versuch.....	175
8.4.3	Byte-Code-Instrumentierung .....	179
8.4.4	Byte Code Engineering Library .....	182
8.4.5	Der Monitor im ersten Einsatz .....	192
8.5	Neuerungen in Java 5.....	195
8.6	Bewertungskriterien für kommerzielle Werkzeuge .....	199
8.6.1	Schlechte Antwortzeiten bei geringer Systemlast.....	200
8.6.2	Ressourcenbedarf der Werkzeuge.....	201
8.6.3	Ressourcenermittlung und Auslastungsprofile .....	202
8.6.4	End-To-End Monitoring und Transaktionstracking.....	204
8.6.5	Weitere Anforderungen .....	206
8.7	Zusammenfassung.....	207
<b>9</b>	<b>Visualisierungstechniken .....</b>	<b>209</b>
9.1	Motivation.....	209
9.2	Grundlegende Aspekte der Visualisierung .....	211
9.3	Streudiagramme .....	211
9.3.1	Einsatz von Mobilfunktechnologien .....	212
9.3.2	Laufzeiten eines Servlets .....	214
9.3.3	Speicherverhalten einer J2EE-Applikation .....	215
9.4	Balkendiagramme und Histogramme.....	217
9.4.1	GPRS-Benchmarkdaten im Histogramm .....	217
9.4.2	Klasseneinteilung über Anwendungsfälle.....	219

---

9.5	Interaktive Visualisierungstechniken.....	219
9.5.1	Eine Versicherungsanwendung im Internet .....	220
9.5.2	Ergebnisse des Tests .....	222
9.5.3	Die Analyse mit Treemaps.....	223
9.5.4	Detailanalyse und Problemlösung.....	226
9.6	Ein Chartgenerator in Perl .....	227
9.6.1	Die Struktur der Rohdaten .....	228
9.7	Zusammenfassung.....	233
<b>10</b>	<b>Beispielszenario Webservice.....</b>	<b>235</b>
10.1	Motivation.....	235
10.2	Rahmenbedingungen und Gestaltungsmöglichkeiten .....	236
10.3	Das Beispielszenario .....	237
10.4	Aufbau der Testumgebung.....	238
10.4.1	Der Webservice.....	238
10.4.2	Applikationsserver und Lasttreiber .....	240
10.5	Der Single-User-Test .....	242
10.6	Der Webservice als Warteschlangenmodell.....	243
10.6.1	Modellierung von Mehrfachressourcen .....	244
10.6.2	Analytische Performanceprognose und Validierung .....	245
10.7	Die Simulation eines Webservice .....	247
10.7.1	Die Komponenten des Modells.....	247
10.8	Das Webservice-Modell in Desmo-J .....	250
10.8.1	Die Modellklasse .....	250
10.8.2	Das Ereignis RequesterRequestServiceEvent .....	253
10.8.3	Das Ereignis WorkerThreadLookUpEvent.....	254
10.8.4	Das Ereignis WorkerThreadFinishService.....	255
10.8.5	Ergänzende Elemente der Simulation .....	256
10.9	Simulationsergebnisse.....	258
10.9.1	Häufigkeitsverteilung der Antwortzeiten.....	259
10.10	Limitierte Warteschlangen.....	262
10.11	Zusammenfassung.....	263
	<b>Sachverzeichnis .....</b>	<b>265</b>

# 1 Einführung

Es gibt wenig Softwareentwicklungsprojekte, in denen die Performance nicht zu irgendeinem Zeitpunkt eine gewisse Rolle spielt. Wenn das Projektmanagement das Thema nicht schon für die ersten Design- und Entwicklungsphasen adressiert und einplant, so wird es spätestens mit der ersten Produktivschaltung auf der Tagesordnung stehen. Und das wahrscheinlich in einer für alle Beteiligten mehr oder weniger unerfreulichen Weise.

Diese Erkenntnis ist nun wahrlich nicht neu und unzählige Studien und Monographien haben sich den damit einhergehenden Problemen mit unterschiedlichen Blickwinkeln und Schwerpunkten gewidmet. Performance ist *das* hauptsächliche Motiv für die Entwicklung des maschinellen Rechnens und damit auch immer schon ein zentrales Qualitätskriterium zur Bewertung von Anwendungen und Programmen gewesen.

Seitdem jedoch die Internettechnologien ihr ursprünglich auf den akademisch-wissenschaftlichen Bereich beschränktes Einsatzgebiet erweitert haben und zu einem der wesentlichen Erfolgsfaktoren und Treiber neuer Geschäftsmodelle und -systeme geworden sind, haben sich die Rahmenbedingungen für das Management der Performance entscheidend verändert. In immer kürzer werdenden Zyklen treten neue Technologien und Paradigmen in Erscheinung und machen es für den einzelnen IT-Manager, Architekten oder Entwickler zunehmend schwierig, einen universellen Überblick zu behalten. Zwangsläufig ergibt sich ein immer feingliedrigeres Bild von Spezialthemen und Einzeldisziplinen und die Frage, wer wann was für das Erreichen der notwendigen Anwendungs- oder Systemperformance leisten muss, lässt sich immer schwieriger beantworten. Inwieweit über Jahre gefestigte Ansichten und Meinungen hierbei eine Rolle spielen, wollen wir am Beispiel des folgenden fiktiven Fallbeispiels ergründen.

## 1.2 Ein Performancemeeting

Wöchentliches Status-Meeting einer IT-Abteilung, Beginn 10:30 im Büro des CIO. Die Teilnehmer sind:

- Frau Dr. Funke, Leiterin Informationstechnologie eines Versicherungsunternehmens und in dieser Funktion Mitglied der Geschäftsführung.
- Herr Graf, Hauptabteilungsleiter Produktion.
- Herr König, Hauptabteilungsleiter Anwendungsentwicklung.
- Frau May, Abteilungsleiterin Methoden und Verfahren / IT-Strategie.

Die übliche Tagesordnung ist abgearbeitet. Konfliktreiche Themen liegen zurzeit nicht vor. Der einzige offen gebliebene Punkt ist die Budgetfreigabe für eine Vorstudie zum Thema „Knowledge Management auf Basis neuer Webtechnologien“. Es ist 11:50, man möchte gemeinsam das Mittagessen einnehmen und nutzt die verbleibenden 10 Minuten bis zur Öffnung der Kantine für ein lockeres Gespräch.

*Frau Dr. Funke:* „Gestern Nachmittag habe ich im Anschluss an die Sitzung des Lenkungsausschusses noch einen Kaffee mit Herrn Becker aus dem Bereich KFZ getrunken. Die testen zur Zeit zusammen mit unseren entsprechenden Vertragspartnern die Anwendung zum Leihwagenmanagement. Wie ich verstanden habe, läuft das ganz gut an. Es werden kaum kritische Fehler gemeldet, und wenn es mal hakt, wird in aller Regel relativ zügig ein Fix geliefert. Meine Gratulation, Herr König. Wenn das so weitergeht, scheinen wir ja seit langer Zeit mal wieder in Time und Budget zu liefern.“

*Herr König:* „Ja. Und was mich noch mehr freut, ist die Tatsache, dass bei den Sachbearbeitern die – ich drücke es mal vorsichtig aus – anfängliche Skepsis gegenüber dem Arbeiten mit dem Webbrowser gänzlich verschwunden ist. Und da unser Architekt seine dogmatische Ablehnung des Einsatzes von Java Applets für die Visualisierung der Statistiken aufgegeben hat, konnten nun einige Wogen geglättet werden.“

*Frau Dr. Funke (lachend):* „Das waren schon sehr hartleibige Diskussionen. Vielleicht kann der KFZ-Bereich hier eine Multiplikatorrolle einnehmen. Damit würden wir auf dem Weg der Anwendungskonsolidierung natürlich ein gutes Stück vorankommen. Einen kleinen Wermutstropfen habe ich aber dennoch. Herr Becker meint, einige Sachbearbeiter wären etwas unzufrieden mit den Antwortzeiten der Anwendung gewesen. Haben wir hier vielleicht irgendwo einen Engpass, Herr Graf?“

*Herr Graf (leicht irritiert):* „Ich weiß jetzt nicht, warum Sie gerade mich das fragen. Soweit ich informiert bin, wird die Testanwendung nicht auf unserer Infrastruktur betrieben. Da haben sich doch die Mitarbeiter von Herrn König und auch von Ihnen, Frau May, selbst was zusammengezimmert, wenn ich das mal so lax sagen darf.“

*Herr König und Frau May (unisono):* „Na, na.“

*Frau May:* „Ja, an so flexible und agile Herangehensweisen werden Sie sich zukünftig noch gewöhnen müssen.“

*Frau Dr. Funke:* „Frau May, meine Herren, ich darf doch bitten. Mal im Ernst: haben wir dort eine Leiche im Keller? Herr Graf, Sie stecken doch dort mitten in den SLA-Verhandlungen und stehen damit auch ein wenig in der Pflicht.“

*Herr Graf:* „Da haben Sie natürlich Recht, Frau Dr. Funke. Der Fachbereich geht in der Endausbaustufe von einer Anzahl Anwender im unteren dreistelligen Bereich aus. Mir machen die paar Hansel mit ihrer Leihwagensteuerung auch keine Sorgen. Da hängen wir meinetwegen noch eine CPU auf die Webserver und dann bekommen wir schon die im SLA-Dokument geforderte „flüssige und ergonomische Dialogverarbeitung“. Aber nach 30 Jahren Betriebszugehörigkeit darf ich mich wohl mit Fug und Recht als einen alten IT-Hasen bezeichnen und als solcher möchte ich doch gerne mal ein paar grundsätzliche Anmerkungen machen.

Wir betreiben, wie Sie alle wissen, mit unserer Infrastruktur seit Jahren auch unter den immer enger werdenden finanziellen Möglichkeiten eine ganze Menge von unterschiedlichen Anwendungen mit teilweise beträchtlichem Transaktionsvolumen. Wenn wir gelegentlich – die Betonung liegt auf gelegentlich – mal hier und da ein Performanceproblem hatten, so haben wir das zügig und, wie ich betonen möchte, in bestem Einvernehmen mit Ihrem Vorgänger – Herr König – lösen können.

Mit dem Einzug der von unseren Methodikern so sehr gefeierten Java-Technologie scheinen wir uns aber ein Maß an Komplexität ins Haus geholt zu haben, die uns für stark frequentierte Anwendungen im Endkundenbereich noch so manche Sorgen machen wird. Da laden sich die Entwickler ohne Sinn und Verstand ein Tool nach dem anderen aus dem Internet, um ihre technische Neugier zu befriedigen, und wir können dann sehen, wie wir dieses Konglomerat verwaltet bekommen. Das macht nicht wirklich Freude. Und mit Verlaub: Sie nutzen für Ihre Anwendung u.a. die Adressverwaltung. Diese wurde in Cobol – Frau May, ich betone: *in Cobol* – zu einer Zeit entwickelt, als Sie noch die Schulbank gedrückt haben und man von Java höchstens in exotischen Reiseführern lesen konnte. Und da die Adressverwaltung noch nie mit unakzeptablem Antwortzeitverhalten aufgefallen ist, können wir diese ja schon mal als Ursache der Performanceprobleme ausschließen.“ (Verschränkt die Arme und lehnt sich zurück.)

*Frau May (zu sich selbst):* „Der wird hier noch für die Wiedereinführung der Lochkarten plädieren.“

„Herr Graf, ich kann ja Ihre Sorgen bis zu einem gewissen Grad nachvollziehen. Aber sind Sie insgesamt nicht etwas zu pessimistisch?“

*Frau Dr. Funke:* „Hätten Sie einen Lösungsvorschlag? Wie können wir dem Betrieb hier Unterstützung zukommen lassen?“

*Frau May:* „Gut, dass Sie danach fragen, Frau Dr. Funke. Ich war in der vergangenen Woche auf einer Java-Konferenz. Dort wurde Speed4J vorgestellt. Dies ist ein beeindruckender Java-Profiler. In einer Roadshow wurde eine Benchmark-Applikation zur Verschlüsselung von Daten einem Profiling unterzogen und innerhalb von 10 Minuten konnten zwei signifikant schlechte Codestellen identifiziert und optimiert werden und so die Antwortzeit um mehr als 60 % gesenkt werden. Ich habe von dem Hersteller eine zeitlich begrenzte Testlizenz erhalten.

Ich schlage daher vor, die Anwendung auf einen Ihrer Unix-Server zu deployen, Herr Graf. Ihre Administratoren müssten dann Speed4J auf dem Applikationsserver installieren. Und während der Fachbereich testet, analysieren wir parallel das Performanceverhalten. Herr König, Ihr Support wäre dann natürlich beim Tuning gefragt.“

*Herr König:* „Kein Thema. Aber wir müssen unbedingt auch einen Netzwerker ins Boot nehmen. Wir haben zwar keine dedizierten Performance-tests durchgeführt, schließlich hatten wir mit der Entwicklung der fachlichen Anforderungen nun wirklich genug zu tun. Aber letztendlich arbeiten unsere Entwickler ja ständig mit der Anwendung und von Performanceproblemen ist mir noch nichts zu Ohren gekommen. Für meinen Verantwortungsbereich habe ich da mal keine schlaflosen Nächte. Herr Graf, wären Sie im Übrigen so nett, Frau May und mir das SLA-Dokument bei Gelegenheit einmal zuzusenden?“

*Herr Graf (friedlich schmunzelnd):* Klar, geht Ihnen heute Nachmittag zu. Wie immer, Frau May, alles ganz easy und speedy. Aber ich bin dabei.“

*Frau Dr. Funke:* „Dann ist ja alles im grünen Bereich. Gehen wir zu Tisch, heute gibt es spanische Küche.“

Dann wollen auch wir ihnen gönnen, sich die Tapas schmecken zu lassen, für uns selbst aber einige Erkenntnisse aus der vorausgegangenen Diskussion ableiten.

### **1.3 Das fachliche und technische Umfeld**

Einige Aussagen sind ein Beleg dafür, dass die IT-Abteilung unter zunehmend schwierigeren Rahmenbedingungen neuen Herausforderungen begegnen und selbst eine aktive Rolle in den Transformationsprozessen des Gesamtunternehmens einnehmen muss. Technologische Innovationen eröffnen ganz neue Perspektiven bei der Gestaltung von Geschäftsprozessen.

Das erwähnte „Knowledge Management auf Basis neuer Webtechnologien“ ist dafür ein kleineres Beispiel. Die debattierte Anwendung zum Leihwagenmanagement, welche von zwei unterschiedlichen Unternehmen aus unterschiedlichen Branchen gemeinsam genutzt wird, deutet aber bereits umfangreichere Transformationen an.

Solche mit deutlichen Umbrüchen verbundenen Prozesse vollziehen sich natürlich nicht ohne Reibungsverluste. Da von diesen in letzter Konsequenz Menschen betroffen sind, zeigen sich Ängste und Widerstände, welche sich zuweilen in Ressentiments gegenüber den ermöglichenden Technologien äußern.

Herr Graf bringt seine Skepsis gegenüber jenen Technologien, welche heute die Basis von verteilten Anwendungssystemen darstellen, offen zum Ausdruck. Konsens besteht offenkundig bei der Bewertung der Rahmenbedingungen, unter denen die IT agieren muss. Herr Graf spricht von „den immer enger werdenden finanziellen Möglichkeiten“, welche ein Motiv für den von Frau Dr. Funke postulierten „Weg zur Anwendungskonsolidierung“ sein dürften.

Frau May hingegen strahlt ein optimistisches Vertrauen in neue Technologien aus, welches sich nicht nur in einer hohen Begeisterungsfähigkeit für bestimmte Produkte äußert, sondern auch in der positiven Hervorhebung bestimmter SW-Entwicklungsmethodiken („flexible und agile Herangehensweisen“) manifestiert.

Der Hauptabteilungsleiter der Anwendungsentwicklung nimmt insofern eine Mittlerrolle ein, als er eine positive Haltung zu neuen Technologien einnimmt, dabei aber zu Gunsten einer guten Kompromissfindung mit den Klienten bereit ist, pragmatische Entscheidungen zu treffen. Wie der Einsatz von Java Applets konkret bewertet werden muss, braucht uns an dieser Stelle nicht zu interessieren.

## 1.4 Performanceaspekte

Die Diskussion liefert durch das, was gesagt wird, aber auch durch das, was nicht gesagt wird, einige höchst bemerkenswerte Informationen. Verfolgen wird diese der Reihe nach:

Frau Funke ist zuversichtlich, was die Einhaltung von Terminen und Budgetgrenzen für dieses Projekt angeht, während noch keine Performance-tests in der Entwicklung stattgefunden haben und Herr König darin auch kein offensichtliches Problem erkennen kann.

Die Aussage, die Sachbearbeiter seien „etwas unzufrieden mit den Antwortzeiten der Anwendung“ ist sehr vage.

Besteht die Unzufriedenheit hinsichtlich der Antwortzeiten? Wenn ja, für alle oder nur Teile der Anwendung? Betrifft die Performanceeinschränkung überhaupt die direkte Interaktion mit dem System oder ist eher die Bearbeitung eines gesamten Prozesses bzw. Geschäftsvorfalles gemeint, an dem unter Umständen mehrere Personen beteiligt sein können (Prozesssteuerung und Workflow)?

Überraschend ist hier weniger, dass die IT-Chefin eine sehr informelle Information weitergibt. Problematisch ist aber die Bereitschaft der Diskutanten das Thema in der Tiefe zu erörtern, ohne zuvor wirklich quantifizierte und damit belastbare Informationen zu den Performanceproblemen einzuholen.

Gehen wir der Einfachheit halber von mangelnden Server-Antwortzeiten aus. Das System reagiert nach dem Verständnis des Sachbearbeiters nicht schnell genug auf Anfragen. Der Produktionsleiter sieht sich hier nicht als Ansprechpartner. Denn die Server, welche spontan für den Test bereitgestellt wurden, fallen in die Verantwortung der Anwendungsentwicklung.

Frau Funke nimmt da eine etwas andere Haltung ein. Sobald die Anwendung live geht, verwendet der Anwender implizit die Infrastruktur des Betriebes. Daher hat dieser auch für die entsprechende Performance zu sorgen. Sie macht dies mit dem dezenten Hinweis auf die laufenden SLA-Verhandlungen deutlich. Ihre Frage nach dem Vorliegen eines Engpasses bleibt unbeantwortet. Hier kann es auch ad hoc keine ausreichende Antwort geben, wir kommen aber im Verlauf des Buches hierauf noch zurück.

Wenn Herr Graf auch eine Verteidigungshaltung einnimmt, stellt er seine eigene Rolle als alleiniger technischer Verhandlungsführer der SLA nicht in Frage.

Warum haben Herr König und Frau May in ihrer Verantwortung für das Anwendungsdesign und grundsätzliche technische Strategieentscheidungen keine Aufgaben im SLA-Management? Herrn König reicht es scheinbar aus, das „SLA-Dokument einmal bei Gelegenheit zu erhalten“.

Das SLA-Dokument selbst fordert eine „flüssige und ergonomische Dialogbearbeitung“ für eine Anzahl Anwender im „unteren dreistelligen“ Bereich.

Bedeutet dies Dialogwechselzeiten von maximal 2 Sekunden für etwa 120 Benutzer in 80 % aller Fälle bei einer maximalen Abweichung von 3 Sekunden für die restlichen 20 %? Oder sind doch eher 1,2 Sekunden Antwortzeit für etwa 350 Anwender in 95% der Aufrufe gemeint?

Dies scheint nicht so wichtig zu sein. Im Zweifelsfalle kann eine CPU-Aufrüstung auf den Webservern ja die unspezifisch geforderte Performance leisten.

Oder ist es möglich, dass die Prozessoren der Webserver bereits jetzt völlig überdimensioniert sind und die Problemursache an einer ganz anderen Stelle zu suchen ist?

Und kann tatsächlich völlig ausgeschlossen werden, dass die Verwendung der bestehenden Cobol-Adressverwaltung in der Neuanwendung ein ganz anderes Performanceverhalten zeigt als bei der klassischen Nutzung?

Zum Schluss offeriert Frau May in Form eines Java Profiler und eines spontan „konzipierten“ Testverfahrens eine für alle Seiten akzeptable Lösung.

Aber eine rechenintensive (Benchmark-) Software zur Verschlüsselung unterscheidet sich doch deutlich von der im Test befindlichen Leihwagenanwendung? Eventuell stellen sich dann ja auch ganz andere Anforderungen an die zu erfassenden Messdaten und somit auch an die zu verwendenden Werkzeuge.

Es bleibt also eine Reihe offener Punkte und unbeantworteter Fragen:

- Die Performanceanforderungen der Auftraggeber (in diesem Falle des Fachbereiches) sind nicht genau spezifiziert.
- Einheitliche Methoden und Vorgehensmodelle sind nicht festgelegt.
- Insbesondere sind Performanceuntersuchungen nicht Teil des Entwicklungsprozesses.
- Es gibt keine klar festgelegten Verantwortlichkeiten und Rollen.
- Werkzeuge zur Erhebung von Performancemessdaten werden mehr zufällig und nicht auf Basis eines Kriterienkataloges ausgewählt.

Alles in allem ist das eingeschlagene Verfahren wenig zielgerichtet und reaktiv. Das schließt nicht aus, dass eine erfolgreiche Optimierung der Anwendung dennoch gelingt und mit dem von Frau May vertretenen Ansatz tatsächlich ein paar Performanceschwachstellen aufgedeckt werden.

Benötigt wird aber eine umfassende Systematik, mittels derer auch unter steigenden Anforderungen an die Anwendungen, wachsenden Benutzergruppen und Ausweitung des Serviceportfolios mögliche Performanceengpässe frühzeitig identifiziert und behoben werden können, bevor diese zu ernsthaften wirtschaftlichen Schäden für das Unternehmen führen.

Geeignete Werkzeuge sind dabei sicher ein bedeutender, aber garantiert nicht ausschließlicher Erfolgsfaktor.

Vielmehr bedarf es auf die jeweiligen Phasen des Entwicklungsprozesses abgestimmter Performanceaktivitäten, die sich in Projekt-, Personal- und Budgetplänen wiederfinden müssen und zu deren Bearbeitung zuwei-

len auch teamübergreifende Abstimmungen sowie Einzelmaßnahmen notwendig werden.

Wie für andere Teilgebiete des Software Engineering muss auch die Organisation der Performance in anwendungs- und projektübergreifende Steuerungsmechanismen eingebettet sein. Dazu gehören u.a.:

- Risikomanagement
- Unternehmensweite Architekturfestlegungen
- Budgetplanungen
- Ausbildungs- und Schulungspläne
- Vorgehensmodelle und Projektmanagement

Betriebs- und Ablauforganisation nehmen damit eine zentrale Rolle ein. Um zu unterstreichen, dass Performance eben nicht nur ein technologisches Themengebiet ist, für das Architekten, Infrastrukturspezialisten und Entwickler zuständig sind, wird in diesem Buch der Terminus Performancemanagement anstelle des geläufigeren Begriffes Performance Engineering verwendet.

## 1.5 Neue Herausforderungen

Seit den frühen 60er-Jahren des vergangenen Jahrhunderts wird IT intensiv zur Automatisierung oder Unterstützung von Geschäftsprozessen genutzt.

Daher ist die Frage berechtigt, ob Performancemanagement im Zeitalter der Internets und der Etablierung serviceorientierter Architekturen tatsächlich einen qualitativ anderen Stellenwert einnimmt oder einnehmen muss, als das bei mehr traditionellen IT-Architekturen der Fall ist.

Tatsächlich hat es schon immer Anwendungen gegeben, die aufgrund eines mangelhaften Durchsatzes oder wegen schlechter Antwortzeiten nur mit massiver Aufrüstung der Hardware oder durch aufwändige Tuningmaßnahmen in Produktion genommen werden konnten. Und schon immer wurden dadurch im Extremfall enorme betriebswirtschaftliche Schäden verursacht.

Im Laufe der Zeit haben sich jedoch mit den Technologien auch die Ursachen für Performanceprobleme verändert. Der wachsenden Komplexität der fachlichen Anforderungen konnte mit den klassischen Verfahren der Softwareentwicklung, Programmiersprachen und Werkzeugen nicht mehr in angemessener Qualität und innerhalb überschaubarer Zeit- und Kostenrahmen begegnet werden. In der Konsequenz haben sich eine Reihe von Methodiken und Technologien herausgebildet, mit Hilfe derer ein gewaltiger Produktivitätsschub bei der Erstellung von Software erzielt werden

konnte. Einige bedeutende Meilensteine auf diesem Weg waren und sind ohne Anspruch auf Vollständigkeit

- die Objektorientierung mit ihren Möglichkeiten zur Modularisierung und Wiederverwendung;
- die Unified Modelling Language zur standardisierten Beschreibung eines Softwaresystems;
- Linux als freies Betriebssystem;
- Java, wegen der Entkopplung von hardware- und plattformspezifischen Besonderheiten durch die Virtual Machine und den einfachen und einheitlichen Zugängen, die diese Sprache zu Themen wie der Netzwerkprogrammierung, des Multithreading usw. bietet. Die Java 2 Enterprise Edition erweitert dieses Portfolio um einheitliche Schnittstellen zur Transaktionsverwaltung, Ressourcenzugriffen, Sicherheit usw.;
- XML, unter anderem zum standardisierten Datenaustausch über das Web;
- Eclipse als Softwareentwicklungsumgebung, zunehmend aber auch als Basis für grafische Anwendungssysteme (Rich Client Platform);
- Webservices, als programmatische Kommunikationsschnittstelle zwischen unterschiedlichsten Systemen und als ein wesentlicher Technologietreiber der Serviceorientierung;
- Open Source und Open Source Communities als Zulieferer für hochqualitative Komponenten und Frameworks.

Die mit diesen Technologien einhergehende Abstraktion fördert die Entwicklung verteilter Anwendungen und bietet eine hohe Flexibilität für die Interaktion zwischen Komponenten und für die Komposition von Services zu vollständigen Anwendungssystemen. Jede Abstraktionsschicht führt aber auch zu zusätzlichem Ressourcenbedarf und die nahezu unendliche Anzahl von Möglichkeiten, die Architekten zur Kombination von Anwendungskomponenten, Systemsoftware und Infrastrukturen zur Verfügung stehen, erhöht das Risiko von versteckten Performanceproblemen.

Die ursprüngliche Anwendungsentwicklung erfolgte unter Verwendung maschinennaher Programmiersprachen. Der Programmierer hatte dadurch naturgemäß die Zielplattform, auf der sein Anwendungscode ausgeführt wurde, deutlicher vor Augen. Die nachfolgenden Programmiersprachen haben zwar schon von der zugrundeliegenden Hardware abstrahiert. Dennoch ist bei klassischen Client/Server-Anwendungen dem Programmierer weitestgehend das Bewusstsein darüber erhalten geblieben, an welchen Stellen seines Codes Zugriffe auf entfernte Systeme (meistens Datenbanken) stattfinden oder nicht. Eine Trennung von Anwendungs- und Systemcode war nicht vollständig gegeben, und wenn sich das in mancherlei Hinsicht als eine Quelle von Schwierigkeiten dargestellt hat, so waren doch

mögliche Performancerisiken eindeutiger und früher zu erkennen als das heute der Fall ist.

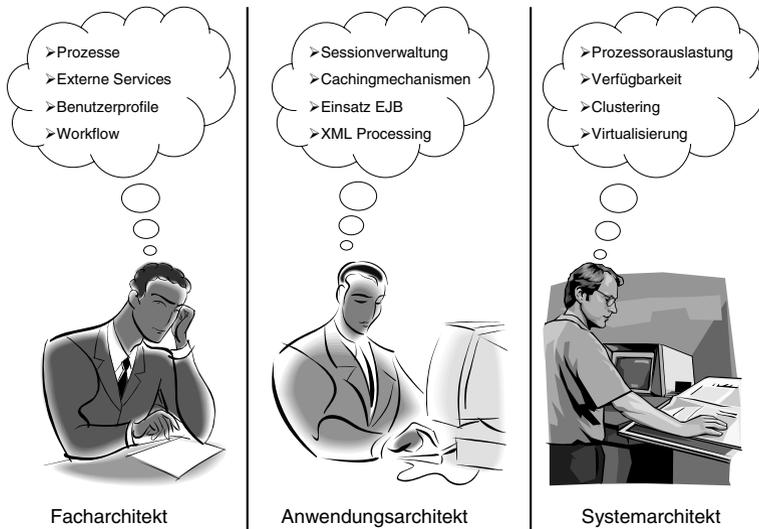
## 1.6 Performance als interdisziplinäre Herausforderung

Mit welcher Intensität ein Entwicklungsprojekt das Performancemanagement vorantreibt oder vorantreiben sollte, hängt vom Projektumfang (Anzahl der Komponenten, Anzahl der Projektmitarbeiter, eingesetzte Technologien) und der Bedeutung der durch das Projekt realisierten fachlichen Funktionen und unterstützten Geschäftsprozesse ab. Diese beiden Faktoren müssen nicht zwangsläufig miteinander korrelieren. So können manche wichtige fachliche Anforderungen gelegentlich mit relativ einfachen Technologien umgesetzt werden und umgekehrt.

Vielfach wird das Thema Performance durch einen *Performancearchitekten* verantwortet, wobei diese Rolle mitunter auch in Personalunion durch einen IT-Architekten wahrgenommen wird, dem noch andere Aufgaben im Projekt obliegen.

Ungeachtet dessen, ist Performancemanagement per se eine fortlaufende Aufgabe, zu deren Bewältigung zu bestimmten Zeiten Spezialisten für gewisse Verfahren und Technologien involviert werden müssen. Dies kann im Rahmen der Performanceanalyse, des Performancemonitoring oder auch der Performanceoptimierung und des Tunings notwendig sein.

Dass diese Formen der Zusammenarbeit zur Erreichung von Teilzielen benötigt werden, ist in der Regel auch unstrittig. Schwierig gestaltet sich jedoch häufig die Entwicklung von Gesamtkonzepten und ganzheitlichen Vorgehensweisen im Sinne eines strategischen Performancemanagements. Nach Überzeugung des Autors sind die unterschiedlichen Perspektiven zur Performance, die Fach-, Anwendungs- und Systemarchitekten aufgrund ihres jeweiligen Rollenverständnisses einnehmen, hierfür die hauptsächliche Ursache. Die Abbildung 1.1 illustriert das Problem. Jeder der drei Architekten widmet seine Aufmerksamkeit zwar wichtigen und richtigen Fragestellungen – es fehlt jedoch die übergeordnete und koordinierende Sicht und das Management der vielschichtigen Abhängigkeiten zwischen den zahlreichen Einzelaspekten. Wird dieses Defizit überwunden, bestehen gute Chancen das Performancemanagement nachhaltig zu verbessern und über Projektgrenzen hinaus zu etablieren.



**Abb. 1.1.** Unterschiedliche Architekturrollen und Performanceperspektiven

Eine an gemeinsamen Performancezielen ausgerichtete Zusammenarbeit der drei Architekturgruppen erfordert einerseits einen adäquaten organisatorischen Gestaltungsrahmen. Sie macht andererseits den Einsatz von Methoden, Modellen und Werkzeugen notwendig, welche eine langfristige Gültigkeit haben und nicht durch die zuweilen kurzlebigen Einzeltechnologien immer wieder in Frage gestellt werden.

## Literatur

- Cömert C, Fischer S, Fröhlich D, Lekkas P, Schmalenbach C (2004) Performance Management von Enterprise-Java-Anwendungen Teil 1, Objektspektrum 1/2004
- Cömert C, Fischer S, Fröhlich D, Lekkas P, Schmalenbach C (2004) Performance Management von Enterprise-Java-Anwendungen Teil 2, Objektspektrum 2/2004

## 2 Performance und Entwicklungsprozess

### 2.1 Motivation

Die Performanceeigenschaften eines IT-Systems werden von einer Vielzahl unterschiedlichster Faktoren beeinflusst. Nur wer sich der vielfältigen und komplexen Zusammenhänge bewusst wird, hat letztendlich die Chance, die notwendigen Performancequalitäten einer vollständigen Anwendung oder auch eines einzelnen Services zu erkennen und die notwendigen Maßnahmen zu ergreifen, um diesen Qualitätsmaßstäben gerecht zu werden.

So evident dieser Sachverhalt auch erscheinen mag, so muss doch konstatiert werden, dass er in der Praxis oftmals nur unzureichend berücksichtigt wird. Denn die Entwicklung und der Betrieb von Anwendungssystemen beruhen aus guten Gründen im hohen Maße auf Spezialisierung und Arbeitsteilung. Daraus resultieren höchst unterschiedliche Blickwinkel und Perspektiven der Akteure und in der Konsequenz wird bereits die Herausbildung gemeinsamer Zielvorstellungen schwierig. Und solange Fach- und IT-Architekten, Entwickler und Systemadministratoren nicht über einheitliche Vorstellungen von den Anforderungen und den daraus abzuleitenden Vorgaben an Architektur und Design verfügen, bleiben zu viele Einflussgrößen dem Zufall überlassen und die Performance wird mehr oder weniger zu einem Glücksspiel.

Interdisziplinäre Arbeitsweisen sind also notwendig, um den Herausforderungen des Performancemanagements adäquat begegnen zu können. Und dies bedeutet mehr als nur den Austausch von Informationen zwischen den Vertretern der verschiedenen Disziplinen. Vielmehr müssen die unterschiedlichen Teilaspekte miteinander verzahnt und in Beziehung gesetzt werden.

## 2.2 Rollen und Aufgaben

Bevor wir uns jedoch den Fragen widmen können, wie eine solche Interdisziplinarität erreicht werden kann und welche Themen denn überhaupt einer übergreifenden Bearbeitung bedürfen, gilt es zunächst ein Verständnis darüber zu gewinnen, welche Perspektiven auf die Performance die Vertreter der verschiedenen Rollen einnehmen.

### 2.2.1 Der Facharchitekt

Dem Facharchitekten obliegt die Aufgabe, die Produktivität der von ihm betreuten Geschäftsbereiche zu steigern. Dazu muss er klare Vorstellungen darüber besitzen, wie mittels moderner Technologien neue Geschäftsmodelle etabliert werden können. Das momentan deutlich erkennbare Bestreben vieler Unternehmen, nach Jahren der Kostenreduktion nunmehr wieder über innovative Lösungen nachzudenken, fordert von dem Facharchitekten als Mittler zwischen Business und IT die Kenntnis darüber, wo die bestehenden Systeme zu einem Hemmnis für die Wahrnehmung potentieller Marktchancen geworden sind und durch eine serviceorientierte Neuausrichtung der Unternehmensarchitektur ergänzt oder ersetzt werden müssen. Selbstverständlich spielt dabei Performance eine bedeutende Rolle, schließlich geht es ja um die Beschleunigung von Betriebsabläufen, die Vermeidung von Leerlaufzeiten, die Erhöhung der Auslastung von Organisationen und Systemen. Allerdings abstrahieren die damit verbundenen Kenngrößen und Metriken weitestgehend von den technologischen Gegebenheiten und Rahmenbedingungen. Sie werden formuliert in der Terminologie und der Sprache des zugrundeliegenden Business. Der Facharchitekt darf seine Rolle natürlich nicht als eine technologieneutrale verstehen. Dennoch wird und kann er sich in der Regel sicher nicht sehr detailliert mit Fragen beschäftigen, welche die einzusetzende Infrastruktur und Systemkomponenten betreffen. Grundlegende technische Architekturentscheidungen müssen unter Beteiligung des Facharchitekten getroffen werden. Ohne ihn kann beispielsweise nicht festgelegt werden, ob die Anwendungsarchitektur auf Basis der J2EE-Plattform konzipiert wird, welche Funktionen als Service implementiert werden, oder ob als Benutzerschnittstelle ein Webbrowser eingesetzt wird. Denn solche Entscheidungen sind nur vordergründig rein technischer Natur; in der Regel haben sie direkte Auswirkungen auf die Fachlichkeit und müssen hinsichtlich ihrer Vor- und Nachteile, der mit ihnen verbundenen Kosten, möglicher Implikationen für das Gesamtunternehmen usw. bewertet werden.