Erfahrener Entwickler von Enterprise-Anwendungen (Java / JEE) und Android-Apps

Name: Axel Müller
Jahrgang: 1971
Berufserfahrung: 19 Jahre
Verfügbarkeit: Verfügbar ab 1. Dec 2017
Stundensatz: 85 €
Nationalität: Deutsch

KURZVORSTELLUNG

Wie man aus meiner Projekthistorie erkennen kann, entwickle ich Software aus Leidenschaft. Oft kommt mir dabei mein betriebswirtschaftlicher Background, meine lösungsorientierte Herangehensweise und meine Motivation durch Arbeit im Team zugute.

ICH BIETE

IT, Entwicklung
Softwareentwicklung / -programmierung
Java (allg.)
Eclipse
Oracle Database
Datenbanken
Android Entwicklung
Web
JavaScript
Spring
AngularJS
J2EE (Java EE)
UNIX
Plattformen / Betriebssysteme
SQL

FÄHIGKEITEN UND KENNTNISSE

*** Hardware ***
PC (auch Mehrprozessor-Systeme), Sun U10 und E450, Raspberry Pi
RFID-Reader Intermec IF-5, RFID-Reader Samsys MP9320, Digital I/O-Karte Sealevel

*** Betriebssysteme ***
Unix (Linux, Solaris, ResinOS), Windows

*** Datenbanken ***
DB2, mySQL, Oracle, PostgreSQL

*** Server ***
Apache Web Server, Borland Application Server, Glassfish, JBoss, Jetty, Oracle Application Server, Tomcat, WebLogic, WebSphere Application Server (WAS)

*** IDE ***
IntelliJ, Rational Software Architect (RSA), Rational Software Developer, Eclipse, WebSphere Application Developer Studio (WSAD), WebSphere Device Developer (WSDD), WebSorm

*** Applikationen ***
Ant, BSCS, Bugzilla, Cactus, Clover, Cobertura, Cruise Control, CMVC, CVS, CyberScheduler, Cyrus IMAP Server, Docker, Ethereal/Wireshark, FreeS/WAN, Grinder, JUnit, JMeter, Liquibase, MagicDraw, Make, Maven, MQ, MQe, NetBeans, TogetherJ, TWiki, openArchitectureWare, OptimizeIt, Oracle, Samba, Sendmail, SnipSnap, SOAP-UI, Subversion, Visual Age C++, VirtualBox, VMware, XEmacs

*** API/Bibliotheken ***
Android, Angular, Apache Commons, BMP, Castor XML, Cling, CMP, CMR, commons-*, Cryptix, CSS3, DBUnit, Dozer, EasyMock, EJB, Genson, HTML5, Reports, Jasmine, Javascript, Java Web Start, Java Connectoren (JCA), JCE, JMX, JNI, JodaTime, JPA, jQuery, JSON, JSP, JTidy, JUnit, Hibernate, Karma, LDAP, Log4e, Log4J, Mockito, Object Relational Bridge, Oracle-XML-Parser, OSGI, OSWorkflow, Pi4J, ProjectX, ReactiveX (RxJS), Regex for Java, REST, RMI, Quartz, Semantic UI, Servlet, SLF4J, Spring / Spring Boot, Struts, StrutsTestCase, Swing, System Management Framework (SMF), TopLink, VisiBroker, WSS4j, Xalan, Xerces, XDoclet, XML, XMLSec, XPath, XSL

SPRACHKENNTNISSE

Deutsch	Muttersprache
Englisch	Fließend

REFERENZEN

11.2015 – 0.0

Open-Source-Projekt

Projektinitiator und Entwickler

Konzeption und Entwicklung einer Applikation ("Smart Appliance Enabler") zur Integration beliebiger Geräte (Wärmepumpe, Waschmaschine, Wallbox für Elektroautos, ...) in eine Smart-Home-Steuerung. Für die Kommunikation mit der Steuerung habe ich das SEMP-Protokoll des Wechselrichtergestellers SMA implementiert, eine Kombination aus UDDI und REST. Als Laufzeitumgebung habe ich Spring Boot gewählt, das auf einem Raspberry Pi läuft. Dessen digitale Schnittstellen werden über Pi4J angesprochen, um digitale Schalter zu schalten und Impulse von digitalen Stromzählern auszuwerten. Ferner wurden Adapter für HTTP und Modbus implementiert, damit entsprechende Schalter/Stromzähler ebenfalls besteuert bzw. ausgewertet werden können.

Das Frontend wurde als "Single Page"-Webanwendung mit Angular 4 und TypeScript entwickelt, wobei als UI-Framework Semantic-UI zum Einsatz kam. Über Spring Boot werden REST-Services bereitgestellt, mit denen die Webanwendung mittels JSON kommuniziert.

Für die Serialisierung nach JSON wurde Genson verwendet, damit die Typ-Informationen nicht verlorengehen. Um ein responsive Design zu gewährleisten erfolgt die Anbindung der REST-Services asynchron über das ReactiveX-Framwork bzw. dessen Javascript-Implementierung RxJS. Für die Einbindung von Sematic-UI war teilweise der Rückgriff auf jQuery erforderlich. Zum Erstellen und Ausführen der Tests kommen die Frameworks Jasmine und Karma zum Einsatz.

Für die Durchführung der Builds und Tests - auch des Web-Fontends - wird Maven verwendet. Dabei wird das Artefakt als war-Datei und auch als Docker-Image bereitgestellt. Für Letzteres wird auf dem Raspberry Pi ResinOS als Host-OS verwendet.

Für die Entwicklung werden die IDEs IntelliJ (für den Spring Boot-Teil) und Webstorm (für Angular) verwendet.

Das Projekt habe ich als Open-Source-Projekt initiiert, das bei Github unter dem Namen "SmartApplianceEnabler" gehostet wird. Es wird von SMA als Referenzimplementierung empfohlen und der Anwender-Support läuft dementsprechend in einem Forum des Herstellers.

Hardware: PC, Raspberry Pi

OS: Linux, ResinOS

Applikationen: Docker, Git, IntelliJ, Maven, Spring Boot, WebStorm

APIs / Bibliotheken: Angular 4, Cling, Genson, HTML5, CSS3, Jasmine, JavaScript, jQuery, JodaTime, JSON, JUnit, Karma, Pi4J, Spring, ReactiveX (RxJS), REST, Semantic UI, SLF4J, TypeScript, XML, XPATH

10.2015 – 10.2016

Open-Source-Projekt

Projektinitiator und Entwickler

Zur Erweiterung meiner Kenntnisse in hardware-naher Programmierung habe ich für einen Flugplatz eine solarbetriebene Wetterstation mit Mobilfunk-Anbindung auf Basis eines Arduino gebaut und die Software dafür entwickelt. Die gemessenen Größen sind: Windgeschwindigkeit, Windrichtung, Temperatur, Luftfeuchtigkeit, Luftdruck, Niederschlagsmenge und Sonneneinstrahlung. Das Projekt wurde als Open-Source-Projekt geführt und wird bei GitHub gehostest.

Hardware: Arduino
Applikationen: Arduino IDE

9.2013 – 0.0

DB Systel

Android-Entwickler

Bei 12.000 Zugbegleitern der Deutschen Bahn dient das "Mobile Terminal (MT)" zur Kontrolle von Fahrkarten und eTickets, zum Verkaufen und Drucken von Tickets etc.. Die aktuelle Generation mobiler Terminals basiert auf Android 4.2. Die einzelnen fachlichen Funktionen sind dabei in Form separater Apps implementiert, die individuell aktualisiert werden können. Meine Aufgabe bestand zunächst im Konzipieren einer geeigneten Enwicklungsumgebung, wobei die Wahl auf ein virtualisiertes Linux mit Eclipse und Maven fiel. Im Rahmen der Software-Entwicklung war ich verantwortlich für die Implementierung der Apps "Fahrpreisnacherhebung" und "Druck" sowie teilweise auch "Verkauf". Jede App besteht aus entsprechenden Activities, Fragmenten und Services, wobei die Parametrisierung sowohl über App-spezifische SQLite-Datenbanken als auch über eingebundene Native-Bibliotheken erfolgte.

Für die Kommunikation der Apps untereinander war eine projektspezifische IPC-Implementierung vorgegeben. Auch für technische Transaktionen sowie fachliche (langlaufende) Transaktionen gab es projektspezfische APIs, die von den Apps genutzt wurden.

Parallel zur Implementierung der Apps bestand meine Aufgabe in der Erstellung eines Frameworks zur Testautomatisierung, welches wiederum auf UIAutomator basiert. Dieses wird inzwischen nicht nur im Rahmen der Entwicklung eingesetzt, sondern aufgrund der Vielzahl der Testfälle auch durch den Systemtest. Zugriffe auf die Hardware (Drucker, Belegscanner, Barcodeleser, Magnetkartenleser, NFC) erfolgten über einen Hardware-Abstraction-Layer, für den ich zwecks automatisierter Testdurchführung eine durch das Testframework steuerbare Mock-Implementierung entwickelt habe. Das Bauen der Software und die automatisierte Ausführung der Tests erfolgen mit Jenkins, der auf einer AWS-Cloud läuft. UI-Tests, die auf der Android-Hardware ausgeführt werden sollen, werden an Jenkins-Slaves delegiert, die auf einer Linux-Maschine laufen, an der mehrere MT angeschlossen sind.

OS: Linux, Windows 7

Applikationen: Eclipse, Maven, SQLite, SQL Developer, Jira, Jenkins, Confluence, HP Quality Center, Subversion, VirtualBox

APIs / Bibliotheken: Android 4.2, JSON, JUnit, Log4j, XML, XPath

10.2010 – 8.2013

DER Deutsches Reisebüro

Entwickler

Komplettsystem für touristische Prozesse

Als Reaktion auf starke Veränderungen der Strukturen und Wertschöpfungsketten der Reiseindustrie durch das Internet entwickelt das DER ein neues System zur Abbildung der gesamten touristischen Prozesse. Die Umsetzung erfolgt durch verschiedene Teams, welche jeweils für einen Teilprozess verantwortlich waren. Ich war zunächst ein Jahr im Team Booking tätig, dessen Aufgabe im Management (CRUD) von Reservierungen für eigene Produkte und auch Produkte aus Fremdsystemen bestand. Über Schnittstellen wurden die Reservierungen für nachgelagtere Prozesse (Dokumentendruck, Payment, …) verfügbar gemacht. Durch meinem Wechsel in das Team Product verlagerte sich der fachliche Fokus auf die Definition und den Lebenszyklus von touristischen Produkten und den mit ihnen assoziierten Regeln für deren Abbildung immerhin ca. 400 Entitäten erforderlich sind.

Die Entwicklung erfolgt modellgetrieben auf Basis von openArchitectureWare. Während im Bereich Booking die technische Modellierung mit UML unter Verwendung von MagicDraw erfolgt, werden im Bereich Product textuelle Modelle (XText) erstellt. Generiert wurden große Teile der Hibernate-basierten Persistenzschicht (JPA), das Grundgerüst für die EJB-basierten Services sowie der RCP-Client.

Im Rahmen der Umsetzung neuer Anforderungen und Fehlerbehebungen erstreckte sich meine Tätigkeitsbereich vom RCP-Client bis zur Datenbank. Dazu erfolgte in der Regel zunächst die Anpassung der Modelle und die Erstellung/Änderung der Geschäftslogik in den Services. Obwohl der Persitenz-Code in großen Teilen generierte wurde, waren auch hier Anpassungen vorzunehmen (z.B. Validatoren) und natürlich SQL-Scripts zur Migration der vorhandenen Daten zu erstellen und das Datenbankänderungstool Liquibase zu integrieren. Die Anpassung des RCP-Clients bestand vor allem in der Anpassung des Layouts sowie der Anbindung von Service-Aufrufen, wenn die Generierung nicht leistungsfähig genug für das gewünschte Verhalten war. Neben der Erfassung von Produkren durch den RCP-Client wurden Produkte auch aus anderen System importiert. An der Entwicklung der dafür notwendigen WebService- und Datei-basierten Schnittstellen war ich ebenfalls maßgeblich beteiligt. Die Erstellung von JUnit-Test war obligatorisch, teilweise unter Verwendung von Mocks (Mockito). Als Development- und Deploymentplatform wurde JBoss 4.3 eingesetzt. Weil die Entwicklung auf eigener Hardware erfolgte, konnte ich Linux als Betriebssystem einsetzen. Auch auf den Deploymentservern in den diversen Testumgebungen wurde Linux eingesetzt. Das zentrale Tool für Anforderungen und Fehlermeldungen und zur Abstimmung zwischen Entwicklern, Fachdesignern und Testern war Jira.

OS: Linux

Applikationen: Eclipse, openArchitectureWare, JBoss, Oracle 10g, Oracle SQL Developer, MagicDraw, Jira, Liquibase, Subversion, SmartSVN

APIs / Bibliotheken: EJB 3.0, JPA, JMS, JMX, Apache Commons, Dozer, JUnit, Hibernate, Mockito, Log4j, XML

1.2010 – 9.2010

EP Consulting Sp. z o. o.

Entwickler

Modernisierung der Warenwirtschaft

Host-basierte Programme des Warenwirtschschaftssystem eines großen Handelskonzerns sollten durch Web-Anwendungen ersetzt werden. Dazu wurden zunächst Prototypen auf Basis verschiedener Technologien (u.a. auch „Host-like“ im Browser) entwickelt, um die Akzeptanz bei den Host-gewöhnten Nutzern zu ermitteln. Letztlich erfolgte die Umsetzung auf Basis von Spring MVC. Auch die Services und die Datenbank-Zugriffe wurden mit Spring realisiert. Meine Aufgabe bestand in der Umsetzung fachlicher Teile der Anwendungen – von der Web-UI bis zum Datenbankzugriff.

Daneben habe ich Teile eines Frameworks zur Steuerung von Batch-Verarbeitungen auf dem Host entwickelt. Darauf aufbauend habe ich größere Batch-Programme von COBOL nach Java portiert. Weil die Tests auch auf dem Host ausgeführt wurden, habe ich mir entsprechendes Host-Know-How angeeignet.

Aufgrund meiner Erfahrungen aus vorangegangenen Projekten habe ich eine Continuous-Integration-Umgebung auf Basis von CruiseControl, Maven und Archiva unter Linux aufgebaut.

OS: Windows, Host, Linux

Applikationen: Eclipse, Subversion, Tomcat, Firefox, CruiseControl, Maven, Archiva

APIs / Bibliotheken: Apache Commons, Log4j, Joda Time, JavaScript, Spring

1.2010 – 2.2017 **eigenes Projekt**

Entwickler

Android-basierte Navigation für Hängegleiter

AndroFlight - die von mir entwickelte Android-Applikation für Gleitschirm-/Drachenpiloten bietet alle Funktionen eines Variometers mit Log-Funktion, ermöglicht aber durch den Zugriff auf Online-Resourcen wertvolle Informationen, die ein Offline-Variometer nicht bieten kann.

Die Benutzeroberfläche besteht aus diversen Activities, wobei die zentrale Activity Karten von Google Maps und OpenStreetMap unterstützt. Die umfangreiche Analyse und Aggregation der Flugdaten (Sensoren: GPS, Druck, Beschleunigung) sowie die Kommunikation mit Online-Resourcen sind als Service implementiert. Luftraumdaten werden nach dem automatischen Download in die SQL-Datenbank importiert und für performante Selektion im Rahmen des Renders aufbereitet. Für den Upload der Flugaufzeichnung in eine Flugdatenbank wurde ein SyncAdapter implementiert. Die Art und Weise der anzuzeigenden Informationen kann sehr flexibel über Preferences konfiguriert werden, die – wie die Anwendung insgesamt – lokalisiert (Englisch und Deutsch) angezeigt werden. Die Verwendung von Google Guice zur Dependency Injection erlaubt eine sehr gute Testbarkeit und klare Trennung der fachlichen Funktionen. Neben dem Emulator verwende ich zum Testen verschiedene Handies sowie ein Tablett. Die App ist über den Android Play Store verfübar und hat bisher ca. 25.000 Downloads.

OS: Android, Linux

Applikationen: Eclipse, Android Studio, Subversion, Git

APIs / Bibliotheken: Android SDK 1.1-2.2, commons-logging, Log4j, Goggle Guice

9.2009 – 12.2009

Dr. Eckhard und Partner GmbH

Entwickler

Portierung von COBOL-Programmen nach Java

Das Host-basierte Warenwirtschschaftssystem eines großen Handelskonzerns sollte von COBOL nach Java portiert werden. Die

Umsetzung erfolgt auf Basis einer service-orientierten Architektur, welche den Einsatz der Services in verschiedenen Umgebungen (Host, Java SE/EE) erlaubt. Meine Aufgabe bestand in der Entwicklung von Framework-Komponenten und der Portierung von COBOL-Unterprogrammen. Die Datenbank-Zugriff wurde über Spring JDBC realisiert.

OS: Windos, Host (z/OS)

Applikationen: Eclipse, Subversion, Maven, Quickbuild

APIs / Bibliotheken: Apache Common, Log4j, Joda Time, Spring

2.2009 – 8.2009

eigenes Projekt

Entwickler

Android Live Tracker

Das Android-Programm erlaubt dem Nutzer, daß Personen seines Vertrauens seine Position live (real-time) auf einem beliebigen Internet-Gerät mit Browser verfolgen können. Das Android-Programm besteht aus mehreren Activities und einem Service, der im Hintergrund die Positionsdaten vom GPS-Sensor empfängt. Die Benutzung des Programms gestaltet sich äußerst einfach, weil sich der Programm-Benutzer nirgendwo anmelden muß, da er über eine ihm zeitlich zugewiesende, generierte ID identifiziert wird. Das Android-Programm kontaktiert zunächst eine von mir entwickelte und betriebene Java-Server-Anwendung, um von dort die ID und Konfigurationsdaten zu laden. Zusätzlich kann der Benutzer einige Parameter über Preferences konfigurieren. Die Java-Server-Anwendung stellt auch das Browser-Userinterface zum Verfolgen der Position des Android-Gerätes bereit. Über den GPS-Sensor bestimmt das Android-Programm fortlaufend die aktuelle Position und übermittelt diese an die Server-Anwendung. Die Server-Anwendung übermittelt auch die Anzahl der Anzahl der momentan mit ihrem Browser "zuschauenden" Personen an das Android Programm. Die Benutzeroberfläche der Android-Anwendung ist komplett lokalisiert – derzeit deutsch und englisch.

OS: Android, Linux

Applikationen: Eclipse, Jetty, Subversion

APIs / Bibliotheken: Android SDK 1.1-2.0, Apache Commons, Log4j, JavaScript, Joda Time, JSP, OpenLayers, Servlet

5.2005 – 9.2009

Wincor-Nixdorf

Entwickler

Entwicklung eines Produktes zur Optimierung der Bargeld-Logistik auf Basis von J2EE und SOA

Das Produkt dient der mandantenfähigen Verwaltung der physischen Bargeldströme und damit zusammenhängender Kosten zwischen Tresoren verschiedener Organisationen (Banken, Wertpapiertransportunternehmen, Bundesbank) und Geldautomaten. Das Ziel besteht in der Optimierung von Zeitpunkt und Geldmenge, um eine Minimierung der Kosten zu erreichen. Ein umfangreiches Reporting bietet allen Beteiligten ein hohes Maß an Transparenz.

Das Produkt-Team bestand im Maximum aus 15 Entwicklern und 5 Testern.

Die Anwendung ist Teil einer Produktlinie, wobei jedes Produkt als eine Kombination von SOA-Services und Platform/Infrastruktur-Services realisiert ist, die auf Basis von J2EE entwickelt wurden.

Neben der Entwicklung von Teilen des Produktes war ich auch für die Neu- und Weiterentwicklung von Platform-Diensten verantwortlich und habe die Entwicklungsinfrastruktur maßgeblich mitgestaltet:

Infrastruktur

- Konzeption und Realisierung einer produktlinienkonformen, betriebssystem-unabhängigen "Continous-Integration" (CI) Entwicklungsumgebung auf Basis von CruiseControl, Maven und Rational Application Developer (RAD), wobei für die Integration diverse Anpassungen an Maven-Plugins vorgenommen werden mussten

- Konzeption und Bereitstellung einer produktlinienkonformen Test-Infrastruktur auf Basis von JUnit, Cactus und EasyMock, die besondere Unterstützung für Aspekte wie Service-Mocking, Mandantenfähigkeit oder Concurrency-Test bietet und dadurch Entwickler zur Erstellung von Tests ermutigt. Die Testabdeckung wird kontinuierlich durch Cobertura überwacht und von der Projektleitung verfolgt.

- Einrichtung eines Projekt-Wiki (SnipSnap) zum Austausch von technischen Informationen unter Entwicklern (Probleme und Lösungen, Best Practices, ...)

Unterstützung von anderen Produkt-Teams bei der Nutzung der Entwicklungs-infrastruktur

Platform

- Konzeption eines Service zur Verwaltung von Benutzern und Mandanten, wobei durch eine Realisierung von Aspekten wie Authentifizierung und Autorisierung in JAAS-konformer Weise eine Integration in WebSphere ("Custom User

Registry") und JBoss ("Login-Module") ermöglicht wird. Auch die Mandantenfähigkeit wird dabei sichergestellt. Flexible Password-Policies erlauben das Erzwingen der Einhaltung von Sicherheitsrichtlinien.

- Konzeption und Realisierung eines Service zum Propagieren von Geschäftsereignissen, sowie von weiteren Services, welche diese Geschäftsereignisse journalisieren oder sie als Report an "Stakeholder" weiterleiten (z.B. per Email).

- Für die Aufbereitung und Lokalisierung wurde ein weiterer Service konzipiert und realisiert, der ebenfalls zur Anzeige von Geschäftsereignissen im User Interface verwendet wird.

- Konzeption und Realisierung eines Service sowie zusätzlicher Tools zur Erzeugung und Verwaltung von Lizenzen sowie zur Sicherstellung der in den Lizenzen vorgegebenen Nutzungsbestimmungen

- Konzeption und Realsierung eines JCA Resource Adapters (CCI-Interface) zur Einbindung von Logik auf externen Systemen über RMI

- Konzeption und Realsierung eines Report-Service auf Basis von Jasper-Reports zur Erstellung von Reports in verschiedenen Formaten (TEXT, PDF, XML, CSV, ...) einschließlich Anpassungen der Build-Infrastruktur zum Sammeln, Compilieren und Bereitstellen der Reports von verschiedenen Services.

- Umstellung eines Produktes von J2EE 1.4 auf Java EE 5 (inkl. JPA) mit Glassfish

Konzeption und Realisierung eines MDSD-Ansatzes auf Basis von openArchitectureWare um J2EE 1.4 und Java EE 5 innerhalb der Produktlinie parallel zu unterstützen sowie eine Steigerung der Effizienz von Entwicklern durch Wegfall von Routine-Tätigkeiten zu erreichen. Die dabei erstellte textuelle DSL basiert auf der Produktlinienarchitektur und stärkt diese damit.

- Bereitstellung sämtlicher Services über RMI und als Web Service

Unterstützung von anderen Produkt-Teams bei der Nutzung von Platform-Diensten

Anwendung

- Erstellung von Analyse-Dokumenten und Dokumentation von Use-Cases inkl. Abstimmung mit Kunden Konzeption und Realisierung eines Service zur Verwaltung von Dienstleistern und deren erbringbaren und zu erbringenden Dienstleistungen

- Konzeption und Realisierung eines Service zur Verwaltung von zeitlich variablen Kosten für Dienstleistungen, Zinsen, Gebühren

- Bereitstellung sämtlicher Services über RMI und als Web Service

- Vorbereitung, Durchführung und Nachbereitung eines Lasttests im Benchmark Center von Sun Microsystem (Langen). Im Rahmen der Vorbereitung wurden parametrisierbare JMeter-Scripte sowie weitere Tools/Scripts erstellt, um die Performance der wesentlichen Geschäftstransaktionen unter Last in verschiedenen Cluster-Konfigurationen unter WebSphere und JBoss zu testen

- Unterstützung von Kunden bzw. der QA betreffend der von mir erstellten Services

OS: Windows XP, Linux, Solaris

Applikationen: Rational Software Architect (RSA), Eclipse, openArchitectureWare, WebSphere, JBoss, Glassfish, Oracle 9i/10g, DB2 8.x, Postgres, Maven, CruiseControl, Cactus, Cobertura, SnipSnap, SOAP-UI, JMeter

APIs / Bibliotheken: EJB 2.1/3.0, Java Connectoren (JCA), JMS, RMI Server, Velocity, JUnit, Apache Commons, EasyMock, DBUnit, Log4e, Log4j, JMX, XML, XSLT, dom4j, Dozer, Cryptix, Jasper Reports, Joda Time, WSS4j, XMLSec

11.2004 – 3.2005

IBM Deutschland

Entwickler

Integration und Erweiterung einer RFID-Software-Platform

Die von IBM als Produkt neu entwickelte RFID-Software-Platform soll zuerst in einem Projekt bei Metro eingeführt werden. Bei der RFID-Software-Platform handelt es sich um einen Micro-Broker (Pub-Sub-Engine) zur Steuerung von RFID-Readern, Signalen und Signalgebern, sowie um einer J2EE-Applikation zur Aggregation und Kanalisierung dieser Informationen, zur Software-Verteilung sowie zur Anbindung externer Backends.

Zunächst wurde von mir eine Build-/Testumgebung auf Basis von Apache Maven konzipiert. Dabei musste CMVC eingesetzt für Versionsverwaltung, Release-Management und Bug-Tracking verwendet werden.

Im Rahmen der Realisierung habe ich sowohl Micro-Broker-Agenten designed und implementiert, als auch Komponenten im Bereich der Geschäftslogik auf Basis von EJB. Außerdem habe ich den Web Service eines Backends in eine Komponente integriert.

Den von mir im Vorprojekt entwickelten RFID-Reader-Simulator habe ich erweitert, um variable Testszeanrien auch in Lasttests abbilden zu können.

Hardware: Samsys UHF-Reader MP9320, Signallampen, Bewegungsmelder

OS: Windows XP, Linux

Applikationen: WebSphere Device Developer (WSDD), IBM Service Management Framework (SMF), WebSphere Application Developer (WSAD), WebSphere Appplication Server (WAS), Maven, MQ, CMVC, DB2, JUnit, VMware, TogetherJ, Etherreal

APIs / Bibliotheken: OSGi, EJB, Web Service

5.2004 – 10.2004

IBM Deutschland

Entwickler

Einführung von RFID bei einem Handelskonzern

Metro sieht sich als Vorreiter bei der großflächigen Einführung von RFID im Handel. Die Integration der RFID-Hardware in die vorhandenen Software-Systeme der einzelnen Vertriebslinien wird durch IBM realisiert. Nach einer Einarbeitung in die Spezifikationen und Technologien im RFID-Umfeld habe ich zur Sicherung der Software-Qualität zunächst eine Unittest-Umgebung für OSGI-Dienste konzipiert und aufgebaut.

Für das RFID Innovation Center der Metro in Neuss habe ich für eine bestehende, OSGI-basierte Applikation aus dem Metro Future Store die Anbindung an neue Hardware realisiert. Die Anbindung der Reader-Control-Applikation an Bewegungsmelder und Signallampen erfolgte mit einer Sealevel Digital I/O-Karte, für die eine entsprechende OSGI-basierte Schnittstelle implementiert wurde. Für neue Reader-Generationen von Intermec und Samsys habe ich Anpassungen der Implementierung der jeweiligen Protokolle vorgenommen. Die Entwicklungsaktivitäten erfolgten in enger Abstimmung mit Metro und den beteiligten Hardware-Lieferanten.

Zur effizienten Durchführung und Dokumentation der Tests von RFID-Hardware habe ich eine Reader-Control-Appikation entwickelt. Dazu lassen sich Hardware-Setups und Testfälle mit Variablen und zugehörigen Wertebereichen bzw. –ausprägungen definieren. Die Ergebnisse der Testdurchführung werden durch Servlet-/JSP-basierte Reports einzeln und zueinander korreliert dargestellet.

Im Rahmen des Systemtests habe ich einen RFID-Reader-Simulator entwickelt, mit dem der parallele Betrieb einer großen Anzahl von RFID-Readern vollautomatisch getestet werden kann. Das Verhalten des Simulators kann dabei an den zu testenden Geschäftsprozess angepasst werden. Während dieser Tests wurde mehrere kritische Fehler in der zu testenden Anwendung erkannt und behoben. Auch die Performance konnte durch diese Tests gezielt optimiert werden (z.B. MQe-basiertes Messaging), was zu einer signifikanten Steigerung des Gesamtperformance führte.

Die „Going life“-Phase habe ich als technischer Kundenkontakt aktiv begleitet.

Hardware: Intermec IF-5 UHF-Reader, Samsys UHF-Reader MP9320, Sealevel Digital I/O card, Signallampen, Bewegungsmelder

OS: Windows XP, Linux

Applikationen: WebSphere Device Developer (WSDD), IBM Service Management Framework (SMF), Ant, MQe, CVS, Oracle, MySQL, JUnit, VMware, TogetherJ, Etherreal

APIs / Bibliotheken: OSGi, JMX, ObjectRelationalBridge, Castor XML

1.2004 – 4.2004

IBM Deutschland

Entwickler

Supplier Quality Control System

Im Auftrag der eigenen Wafer-Fabriken wurde ein System zur Überwachung der Lieferantenqualität (SQUIT) entwickelt. Jeder Lieferant muss Dateien mit parametrischen Daten der zur Lieferung vorgesehenen Produkte übermitteln. Diese Daten werden mit Spezifikationen verglichen bevor eine Freigabe zur Lieferung erfolgt. SQUIT besteht aus einem Web-Frontend und einem Backend. Der Schwerpunkt meiner Tätigkeit bestand in der Entwicklung der gesamten Spezifikationsprüfung. Im Zuge dieser Entwicklung habe ich das Persistenz-Framework Apache ObjectRelationalBridge eingeführt, um den Aufwand für Persistenz-Programmierung zu verringern (wird inzwischen projektweit eingesetzt). Weiterhin habe ich eine Umgebung für Continuous Integration mit CruiseControl / JUnit aufgebaut und den Unittest-Gedanken in das Team getragen. Auch an der Entwicklung einiger Bereiche des Web-GUIs mit Struts war ich beteiligt. In diesem Zusammenhang habe ich auch StrutsTestCase eingeführt, um Dialog-Sequenzen automatisiert zu testen.

OS: Windows XP, Linux

Applikationen: WebSphere Application Developer Studio, Tomcat, DB2, CVS, Cruise Control, Ant, JUnit

APIs / Bibliotheken: Struts, StrutsTestCase, Apache ObjectRelationalBridge

11.2003 – 11.2003

Otto Versand

Berater

Evaluierung eines Persistenz-Frameworks

Ziel dieses Beratungsprojektes war die Identifikation von Risiken und möglichen Maßnahmen aus dem Einsatz eines eigen-entwickelten Persistenz-Frameworks. Bestandteil der Untersuchung waren Anwender-Interviews, Code- und Dokumentationsreviews und der Vergleich mit anderen kommerziellen und nicht-kommerziellen Persistenz-Frameworks. Die Ergebnisse wurden dem IT Management als Grundlage einer Entscheidung über die Fortführung der Entwicklungs des Persistenz-Frameworks präsentiert.

Applikationen: TogetherJ

APIs / Bibliotheken: Hibernate, Apache ObjectRelationalBridge, TopLink

10.2001 – 10.2003

GBS Consulting AG

Entwickler / Team-Leiter

MidOffice-Applikation im Börsen-/Stromumfeld

Bei diesem Projekt handelte es sich um ein Kundenprojekt im Rahmen dessen einige Entwickler des Produktes MeX (Midoffice for electronic Exchanges) J2EE-Know-How aufbauen sollten. Dies war Voraussetzung für die für MeX angestrebte Technologiemigration von einer datenbank-zentrischen Client-Server-Applikation hin zu einer komponentenbasierten Anwendung auf Basis von J2EE. Die technische Führungsverantwortung für dieses Projekt mit insgesamt 6 Entwicklern lag bei mir.

Auf Basis der Analyse der existierenden Applikation und der Projektanforderungen des Kunden wurde von mir eine workflow-orientierte, mehrschichtige Architektur entworfen. Design und Implementierung erfolgte im Team, wobei zwei Entwickler von mir gecoached wurden, da sie keine Erfahrung in objektorientiertem Design und der Entwicklung mit Java/J2EE hatten. Signifikanten Aufwand erforderte die Entwicklung des Domain-Modells, da dieses in der bestehenden Anwendung nicht existierte. Für das Design der einzelnen Komponenten wurde TogetherJ verwendet, als IDE kam NetBeans zum Einsatz. Als Applikationsserver wurde zunächst WebLogic 7.0 verwendet, auf Kundenwunsch wurde jedoch auf JBoss 3.0 gewechselt. Als Workflow-Engine wurde OSWorkflow (Open Source; Teil des OpenSymphony Frameworks) verwendet, für die mehrere Ebenen von Workflows konzipiert wurden. Durch die Kopplung des (nicht von mir entwickelten) GUI mit den Workflows wurde eine hohe Parametrisierbarkeit der Anwendung erreicht. Die Workflows selbst konnten über ein von mir entwickeltes JMX-Interface beeinflusst werden. Produkt-Preise bzw. Kurse sollten von den Homepages der Kurslieferanten abgegriffen werden. Dazu habe ich die Schnittstellen so designed, dass HTML zunächst nach XHTML konvertiert wird, sodass auf die Kurse über XPath zugegriffen werden kann. Der XPath und weitere Parameter jeder Schnittstelle habe ich wiederum über JMX administrierbar gemacht. Wegen des Parallelbetriebs mit der Altapplikation mussten bestehende Datenbank-Strukturen bestehen bleiben. Die dadurch notwendige Entkopplung von Domain-Modell und Persistenz-Modell habe ich durch eine XML/XPath-basierte Abstraktion erreicht. Die Persistenz selbst wurde über CMP und CMR (EJB 2.0) gelöst, nachdem wir die Tauglichkeit dieses Ansatzes verifiziert hatten.

OS: Linux

Applikationen: JBoss, BEA WebLogic, Tomcat, Oracle, NetBeans, TogetherJ, CVS, Bugzilla, Cruise Control, Ant, TWiki, Clover, JUnit

APIs / Bibliotheken: EJB, CMP, JMX, Apache Commons, Log4J, XDoclet, OSWorkflow, Xerces (XML), Xalan (XPath), Castor XML, JTidy, JSP, Swing, Quartz, Java Web Start

12.2000 – 10.2001

Vodafone Telecommerce

Entwickler

Erweiterung/Performance-Optimierung des Price-Quote-Server

Die Funktionalität des Price Quote Server sollte in einem weiteren Release erweitert und die Performance optimiert werden. Meine Aufgabe zum einen in der Implementierung der CORBA-Schnittstelle zum Backend-System auf Basis von VisiBroker als Ersatz für die JNI-Schnittstelle. Schwerpunkt meiner Arbeit war jedoch die Optimierung der Performance einschliesslich der Schaffung der dafür notwendigen Umgebung und Tools. Dazu wurden von mir Plugins für das freie Last-Tool Grinder entwickelt, über die mehr als 1000 concurrent User simuliert wurden. Im Rahmen des von mir durchgeführten Profilings wurden Performance-Engpässe identifiziert und von den betreffenden Entwicklern beseitigt. Ich selbst habe Verarbeitung der Digitalen Signaturen (Key-Algorithmus, Key-Länge, Hash-Algorithmus) und die Verwendung des XML-Parsers optimiert. Weiterhin habe ich viel Zeit auf das Optimieren von Betriebssystem-, JavaVM- und WebLogic-Parametern verwendet. Alternativ wurde auch der Betrieb der Anwendung in einem WebLogic-Cluster, auf dem Borland Application Server und auch auf dem Oracle Application Server getestet. Bei allen Tests war auch der Support der jeweiligen Server-Anbieter involviert. Im Ergebnis konnte der Durchsatz der Gesamtanwendung unter WebLogic 5.1 auf das 60-fache gesteigert werden. Abschliessend konnte ich im Sun Benchmark Zentrum einen Performance-Test des Price Quote Servers auf einer Sun Enterprise 10000 mit 64 CPUs durchführen.

OS: Solaris, Linux

Applikationen: BEA WebLogic, Borland Application Server, Oracle Application Server, XEmacs, OptimizeIt, CVS, Bugzilla, Oracle, Grinder (Last-Tool)

APIs / Bibliotheken: EJB, BMP, JSP, Servlet, Xerces, ProjectX, XP, Oracle-XML-Parser, VisiBroker

5.2000 – 9.2000

Vodafone Telecommerce

Entwickler

Entwicklung eines Price-Quote-Server

Der Price Quote Server dient der performanten Beantwortung von Preis-Anfragen von Web-Shops. Die XML-Anfragen über HTTP(S) wurden zunächst gegen einen LDAP-Server authentifiziert und autorisiert. Anschliessend mussten sie in eine proprietäre Darstellung (M2F2) umgewandelt werden, bevor sie an das Backend-System weitergereicht werden. Die von dort erhaltenen Preis-Angebote mussten invers dazu wieder zurück nach XML konvertiert werden.

Als Applikationsserver kam WebLogic 5.1 zum Einsatz mit Oracle 8i als Datenbank.

Das Frontend-Interface zur Annahme der Price Quote Requests inklusive Authentifizierung/Autorisierung wurde von mir designt und implementiert. Ebenso die Schnittstelle zum Backend-System, welche als JNI-Interface implementiert wurde. Weiterhin habe ich die Generierung und Überprüfung der Digitalen Signaturen in Preis-Anfragen und Bestellungen implementiert. Dazu wurde die freie JCE-Implementierung Cryptix verwendet.

OS: Solaris, Linux

Applikationen: BEA WebLogic, XEmacs, CVS, Bugzilla, Oracle

APIs / Bibliotheken: EJB, BMP, JSP, Servlet, JNI, Xerces, Netscape LDAP SDK, JCE/Cryptix, IBM Regex for Java

3.2000 – 4.2000

Mannesmann Arcor

Entwickler

Entwicklung eines Prototyps für Web Order Entry

Der Prototyp sollte die Auftragserfassung für die komplexen Produkte der Telekommunikationsbranche ermöglichen, wobei das Datenbank-Design vom Kunden vorgegeben war. Anforderungsgemäss wurde der Prototyp mit denselben Technologien realisiert, wie die spätere Applikation. Durch die Verwendung von XML zur Datenstrukturierung konnten HTML-GUIs dynamisch durch einen XSLT-Prozessor erzeugt werden. Meine Aufgabe bestand in der Implementierung von Authentifizierung und der XSL-Transformation.

OS: Linux
Applikationen: BEA WebLogic, XEmacs, CVS, Oracle
APIs / Bibliotheken: EJB, BMP, Xerces (XML), Xalan (XSLT, XPath)

8.1999 – 9.1999

i2c Systems

Entwickler

Technologie-Prototyp: J2EE, XML, XSL, WML

Ziel des Prototypen war die Evaluierung von XML/XSL(T) zur Erzeugung terminial-
spezifischer User-Interfaces (Browser, Organizer, Handy).

OS: Linux
Applikationen: BEA WebLogic, XEmacs, CVS, Nokia WAP Toolkit
APIs / Bibliotheken: EJB, Servlet, Xerces, Xalan

8.1999 – 2.2000

i2c Systems

Entwickler / Berater

Aufbau der Entwicklungsumgebung

Planung und Aufbau einer dezentralen Entwicklungsumgebung, die folgende Aspekte zu berücksichtigen hatte:
Objektorientierte Softwareentwicklung in Java
Verwendung von BEA WebLogic als Applikations-Server
Verwendung von Oracle als Datenbank
Entwicklungs-PCs laufen unter Linux (lokal oder remote via VPN)
Verwendung von beliebigen Editoren (XEmacs) und GNU make
Versionsverwaltung mittels CVS (Source Code und Dokumente)
Defect-Tracking mit Bugzilla
Release-Management einschliesslich Shipment-Server
Unit-Testing mittels JUnit

OS: Linux
Applikationen: BEA WebLogic, XEmacs, CVS, Buzilla, Oracle, JUnit, GNU Make

7.1999 – 7.1999

i2c Systems

Entwickler

Technologie-Evaluierung: Applikationsserver

Durchführung einer Analyse des Entwicklungsstandes von J2EE Applikationsservern, um deren Eignung als Plattform für eigene Softwareentwicklungen zu ermitteln.
Gesichtspunkte dabei waren vor allem:
Unterstützte Plattformen
Support diverser Java APIs
Clustering
Support
Im Ergebnis dieser Analyse wurde eine Entscheidung zugunsten von BEA WebLogic getroffen.

Applikationen: BEA WebLogic, IBM WebSphere, weitere Applikationsserver

4.1999 – 2.2000

i2c Systems

Administrator

Start-Up Company: Aufbau der IT-Infrastruktur

Konzeption der IT-Infrastruktur eines Softwarehauses, wobei die Realisierung in den ersten Monaten ebenfalls durch mich erfolgte. Nach einigen Monaten wurde für die Realisierung ein System-Administrator eingestellt, wobei ich weiterhin für die Konzeption verantwortlich war.

Hardware: Serverbetriebstauglich konfigurierte PCs (auch Mehrprozessor-Systeme), Sun U10 und E450, Lucent WaveLAN Office-Router, Nortel Networks Baystack 450 Switches, Netgear Switches, HP / Epson Drucker, HP JetDirect Print-Server
Software: Linux (SuSE-Distribution, Kernel 2.0.x, 2.2.x), Solaris 7, NFS, Samba, Apache Web Server, CVS, Cyrus IMAP Server, Sendmail, CyberScheduler, Bugzilla, VPN FreeS/WAN (IPSEC)

3.1998 – 3.1999

LHS Group

Consultant Partner Sales Support

Technischer Pre-Sales-Support von Systemintegratoren (Alcatel, Ericsson, Cap Gemini u.a.) im Bereich Abrechnungssysteme für Telekommunikationsanbieter. Die Tätigkeit bestand hauptsächlich in der Durchführung von Präsentationen und Produkt-Demonstrationen bei Partnern und Kunden in Europa, Afrika und dem Mittleren Osten. Ein weiterer Schwerpunkt bestand in der Ausarbeitung von Angeboten. Im Rahmen der Planung eines neuen Produkt-Releases erfolgte eine intensive Bearbeitung des Themas IP-Billing.

OS: Windows
Applikationen: BSCS, Personal Oracle

1.1994 – 12.1995

während des Studiums

Entwickler

Email-Server / Modem-Einwahl-Server

Vollständige Programmierung eines multithreaded POP3-Servers sowie eines Mailers mit Features wie z.B. Mail-Weiterleitung, Auto-Antwort etc.. Die Konfiguration erfolgte mittels Mail-Prozessor oder über ein Web-Interface. Vollständige Programmierung eines POP3-Client für Multi-User-Accounts. Daneben habe ich einen Modemeinwahl-Servers programmiert, wobei der Schwerpunkt auf dem Session-Handling einschliesslich der Authentifizierung lag. Der Vetrieb der 3 Programme erfolgte als Shareware (insgesamt mehr als 100 Registrierungen weltweit).

OS: OS/2
Applikationen: IBM Visual Age C++, POP3, Apache Web-Server, REXX

AUSBILDUNG

1998	Technische Betriebswirtschaft Offenburg Diplom-Betriebswirt

ÜBER MICH

In meinen bisherigen Projekten war es oft so, daß ich neben den typischen Entwicklungstätigkeiten durch mein Know-How bestimmte Aspekte des Entwicklungsprozesses und der Entwicklungsumgebung maßgeblich prägen konnte, was wiederum dazu geführt hat, daß ich mein Know-How in diesen Bereichen weiter vertiefen konnte.

Build-Automatisierung/platformunabhängige Entwicklungsumgebungen

Die Gestaltung von Entwicklungsumgebungen hat erheblichen Einfluß auf die Effizienz und die Motivation der beteiligten Entwickler. Aus meiner Erfahrung kann ich sagen, daß die Automatisierung oft unzureichend ist und nicht adäquat zur Team-Größe ist. Auch der Aspekt der Platform-Unabhängigkeit, die durch Java/J2EE eigentlich gegeben ist, wird durch suboptimale Entwicklungsumgebungen unnötig geschwächt. In meinen bisherigen Projekten hatten wir spätestens in den letzten Projektphasen heterogene Build- und Testumgebungen, sodaß es sich bewährt hat, daß ich bereits in den frühen Projektphasen Wert auf eine platformunabhängige Build- und Testumgebung Wert gelegt hatte.

Test-Automatisierung (Continuous Integration)

Es erstaunt mich immer wieder, wenn in Projekten das Erstellen und automatisierte Ausführen von Tests als verzichtbarer (angeblich zu aufwendiger) Luxus dargestellt wird. Auch hier hängt die Motivation der Entwickler hauptsächlich davon ab, daß der Aufwand zur Testerstellung und -ausführung möglichs gering ist z.B. durch Anpassung von Standard-Testframeworks an die technische Architektur und Platform. Eine automatisierte Testumgebung, die unabhängig von der Entwicklungsgebung sein sollte, gibt den Entwicklern autmatisch zeitnahes Feedback über die Testergebnisse. Das führt mit steigenender Testabdeckung (die natürlich überwacht wird) zu einer deutlichen Qualitäts- und Effizienzsteigerung.

Performance-Tests / Profiling

In den meisten Enterprise-Projekten geniessen in der Anfangsphase funktionale Aspekte die höchste Priorität, weil meist unterstellt wird, daß die eigene Anwendung skaliert, weil ja man schließlich auf einer skalierenden Platform aufsetzt. Die Ernüchterung kommt dann in den späteren Projektphasen, wenn erste Tests zeigen, daß sich die Performance-Anforderungen des Kunden so nicht erfüllen lassen. Um diese Situation zu vermeiden, ist die Erstellug eines Konzeptes zur kontinuierlichen Messung der Performance relevanter Business-Transaktionen unerlässlich. Darauf aufbauend muß eine Umgebung mit entsprechenden Tools aufgebaut werden, die - idealerweise ebenfalls automatisiert - die aktuellen Performance-Zahlen liefert. Der Einsatz eines klassischen Profiling-Tools zur Analyse von Performance-Bottlencks ist meiner Erfahrung nach in Enterprise-Projekten nicht praktikabel.

Modellgetriebene Softwarentwicklung

In den letzten Jahren gab es signifikante Fortschritte im Bereich der modellgetriebenen Softwareentwicklung. Während man bisher auf UML zur Modellierung festgelegt war, kann dies heute besipielsweise auch textuell oder in XML erfolgen. Speziell textuelle Modelle machen die Erstellung von Modellen für Entwickler relativ einfach, inklusive voller Unterstützung durch die IDE (Auto-Completion, Outline, ...). Der Vorteil liegt darin, dem Entwickler Abstraktionen auf bestimmte Aspekte der Platform (z.B. Persistenzmechnismus) zu bieten und ihn von lästigen Routine-Programmierungen freizustellen, sodaß er sich auf die Erstellung von Geschäftslogik und zugehöriger Tests konzentrieren kann.

Android

Seit der Verfügbarkeit der ersten Android-Handies beschäftige ich mich im Rahmen von eigenen und auch Kundenprojekten mit Android, weil diese Platform perfekt für mobile Terminals von Enterprise-Anwendungen geeignet ist.

PERSÖNLICHE DATEN

Strasse	Nieder-Rußbacher-Str. 18
PLZ - Ort	63674 - Altenstadt, Hessen
Land	Deutschland
Telefon - Office	+49 (0)177 2935683
E-Mail	axel.mueller@avanux.de
Arbeitserlaubnis	Europäische Union