
Linux als Entwicklungsumgebung für Web-Anwendungen

Martin Raeppe, Avinci
martin.raeppe@avinci.de

Frühjahrsfachgespräch 2003
German Unix User Group

Dieses Dokument dient als Begleitinformation zum gleichnamigen Vortrag auf dem Frühjahrsgespräch 2003 der German Unix User Group.

1 Einführung

Linux hat sich in den vergangenen Jahren immer mehr als zuverlässige Basis für Produktionsumgebungen im Umfeld kritischer Unternehmensanwendungen durchgesetzt. Mit dem Einzug von e-Business wurde dieser Trend nur noch bestärkt. Im Gegensatz dazu war und wird der Desktop nach wie vor von anderen Betriebssystemen dominiert, allen voran Windows. Auch im Bereich der Anwendungsentwicklung werden häufig noch die Argumente

- Verfügbarkeit von Entwicklungswerkzeugen und
- Komfort und hoher Integrationsgrad der unterschiedlichen Werkzeuge

genannt, wenn man nach den Gründen gegen den Einsatz von Linux auf der fragt. Daß diese Argumente inzwischen entkräftet werden können und klare Vorteile wie geringere Investitions- und Betriebskosten für den Einsatz von Linux gerade im Bereich der Anwendungsentwicklung sprechen, soll dieses Dokument in Verbindung mit dem Vortrag am konkreten Beispiel eines großen e-Business Projektes zeigen.

1.1 Das Projekt

Die im Rahmen eines über zwei Jahre andauernden Projekts mit durchschnittlich fünf Teammitgliedern entstandene Software wurde für die Hecaron AG, einem Beschaffungsdienstleister für den technischen Einkauf im Maschinen- und Anlagenbau, erstellt. Das Produkt, ein elektronischer Marktplatz, der im Internet unter www.hecaron.de erreichbar ist, bietet Einkäufern und Anbietern die Möglichkeit, Ausschreibungen für komplexe Zeichnungsteile online verhandeln zu können. Die Plattform wird inzwischen von über 2000 registrierte Teilnehmer genutzt und bildet den gesamten Beschaffungsprozeß (Lieferantensourcing, Ausschreibung, Ange-

botsverhandlung, Einkaufscontrolling, Auftragssteuerung und Lieferantenbewertung) ab. Darüber hinaus bietet sie die Möglichkeit, geschlossene Lieferantennetzwerke für große Einkaufsorganisationen einzurichten und zu verwalten. Weiterhin wurde eine separate Integrationsplattform entwickelt, mit der die Unternehmensanwendungen der Marktplatzteilnehmer (z. B. ERP-Systeme, Office-Anwendungen) mit der Plattform gekoppelt werden können.

Zu den Basistechnologien des Marktplatzes zählen die Java 2 Enterprise Edition und die eXtensible Markup Language (XML). Zudem wurde intensiver Gebrauch von Komponenten und Klassenbibliotheken aus dem Open-Source Umfeld gemacht, ohne die die komplette Eigenentwicklung der Software in diesem Zeitraum nicht möglich gewesen wäre. Zu nennen sind hier z. B. Cryptix JCE (Datenverschlüsselung und kryptografische Prüfsummenberechnung), GNU RegExp (Reguläre Ausdrücke), Apache POI (Im- und Export von MS Office Formaten) und PoolMan (Pooling von Datenbankverbindungen).



ABBILDUNG 1.

B2B-Marktplatz hecaron.de

2 Komponenten einer Web-Entwicklungsumgebung

Zunächst sollen kurz die Hauptkomponenten einer Entwicklungsumgebung aufgeführt werden, bevor auf konkrete Produkte, die im Rahmen des Projektes eingesetzt wurden, genauer eingegangen wird.

2.1 Integrierte Entwicklungsumgebung (Integrated Development Environment, IDE)

Die IDE stellt das Hauptwerkzeug der täglichen Entwicklungsarbeit dar. Zu den Funktionalitäten einer integrierten Entwicklungsumgebung gehören

- Organisation des Programmcodes, zumeist im Rahmen einer Projektverwaltung. Weiterhin werden den Projekten zusätzliche Ressourcen wie z. B. Klassenbibliotheken, Konfigurationsdateien, etc.

-
- Assistenten für wiederkehrende Routinarbeiten (z. B. Anlegen eines neuen Projektes, Erstellen neuer Klassen etc.), um den Entwicklungsprozeß zu beschleunigen. Dies bezieht sich auch auf komplexe Aufgabenbereiche wie die Restrukturierung und Optimierung des existierenden Codes (Refactoring)
 - ein integrierter Debugger

Hochpreisige Anbieter positionieren ihre Produkte inzwischen als sog. Round-Trip-Engineering Tools, die zusätzlich (UML-)Modellierungswerkzeuge bereitstellen. Damit werden in einer geschlossenen Umgebung Änderungen auf der Modellierungsebene direkt in den Programmcode übertragen und umgekehrt. Weitere, ebenfalls nicht der Kernfunktionalität zuzurechnende Bereiche sind das Profiling (Analyse des Laufzeitverhaltens) oder eine automatisierte Qualitätsüberprüfung des Quellcodes (z. B. nach Konformität gemäß konfigurierbarer Richtlinien, Coding Guidelines).

2.2 Sourcecode-Verwaltung

Unter dem Begriff Sourcecode-Verwaltung wird in erster Linie die Versionierung des Programmcodes (und weiterer Ressourcen) auf Projektebene verstanden. Hierbei muß insbesondere die gleichzeitige Bearbeitung einer Ressource durch mehrere Mitarbeiter eines Entwicklungsteams gesteuert werden. Konzepte wie

- das Sperren und Freigeben von Ressourcen
- die automatische Zusammenführung (Merging) von parallel durchgeführten Änderungen an einer Ressource sowie
- die Konfliktauflösung bei sich überschneidenden Änderungen

stehen hier im Vordergrund. Weiterführende Konzepte wie die Verwaltung parallel laufender Entwicklungspfade (engl. Branch), i.d.R. zur Trennung einer Version für die Weiterentwicklung und für das Bugfixing, sind ebenfalls für eine professionelle Umgebung von Bedeutung.

2.3 Deployment

Der Begriff Deployment bezeichnet den automatisierten Kompilierungs- und Erstellungsprozess einer (Web-)Anwendung aus den Programmquellen und allen zusätzlichen Ressourcen. Dabei besteht i.d.R. die Anforderung, das Ergebnis (das Build) an unterschiedliche Systemumgebungen (z. B. Entwicklung, Test oder Produktion) mit den entsprechenden Parametern (z. B. Netzwerkadressen, URLs, Verzeichnispfade) anzupassen.

3 Java IDEs unter Linux

Wie bereits erwähnt hat sich das Bild der unter Linux verfügbaren Entwicklungsumgebungen, insbesondere für die Programmiersprache Java, in den vergangenen zwei Jahren stark zum Positiven verändert. Abbildung 2 zeigt einige der bekannte-

sten IDEs kommerzieller Hersteller und aus dem Open Source Bereich im Vergleich.

	Eclipse 2	Visual Age Prof. 4.0	WebSphere App. Dev. 4.0	Sun Studio ONE	JBuilder
Flexibilität JDK	↑	↓	↑	↑	↑
Editor	⇒	⇒	↔	⇒	↔
Debugger	↔	↑	↔	⇒	⇒
Unterstützung Versionsmgmt.	↑	↔	↑	↓	↑
Deployment	↑	↓	↑	↔	↔
Wizards / Zusatzmodule	⇒	↑	↑	↔	↑
Performance	⇒*	↔	⇒	↔	↔
Stabilität	⇒	↑	⇒	⇒	↔
Roundtrip-Engineering	↑	↓	↓	↑	↔
Tool-Integration	↔	⇒	↔	⇒	↑
Kosten	↑	↓	↓	↑	↔

* Linux-Version mit Motif-Oberfläche

ABBILDUNG 2.

Java IDEs unter Linux im Vergleich

Eine pauschale Empfehlung für oder gegen ein bestimmtes Tool kann nicht gegeben werden. Vielmehr spielen Faktoren wie

- allgemeine Unternehmensrichtlinien für die Softwareentwicklung und Toolsauswahl
- projektspezifische Vorgaben für Basistechnologien (z. B. Version des Java SDK)
- Status des Projektes (Neu- oder Weiterentwicklung)
- vorhandenes Tool-Know-How im Entwicklungsteam

eine wichtige Rolle bei der Auswahl. Fest steht, daß Linux eine erstzunehmende Alternative gegenüber Windows aufgrund der breiten Verfügbarkeit von professionellen IDEs darstellt. Im folgenden soll einer der bekanntesten und in jüngster Zeit häufig diskutierter Vertreter, das Open-Source Projekt Eclipse, näher betrachtet werden.

3.1 Eclipse - Das Projekt

Eclipse (www.eclipse.org) wurde als Open Source Projekt von mehreren namhaften Vertretern der IT-Branche mit dem Ziel initiiert, eine offene, hersteller- und betriebssystemneutrale Integrationsplattform für Softwareentwicklungswerkzeuge zur Verfügung zu stellen. Primäres Ziel ist dabei einen umfassenden Ansatz für die Unterstützung des gesamten Softwarelebenszyklus zu schaffen, auf dessen Grundlage eine möglichst breite Basis von Toolherstellern aufsetzen und neue, miteinander integrierbare Produkte entwickeln können. Inzwischen sind die frei verfügbare Eclipse Java IDE sowie viele kommerziell vermarktete Lösungen auf Basis der

Eclipse Plattform, allen voran WebSphere Application Developer von IBM, erhältlich.

Derzeit federführend in der sog. Eclipse Projektmanagementkommission (PMC) ist das schweizer IBM-Tochterunternehmen Object Technology International (OTI), das auch schon maßgeblich an der Entwicklung von IBMs Visual Age Respository Technologie beteiligt war. Die Neutralität wird durch weitere Partner wie z. B. Borland gewährleistet, deren Einfluß und Verantwortung gleichrangig nach dem Umfang und Wert der erbrachten Leistungen steigt oder fällt.

3.2 Eclipse - Die Java IDE

Die Eclipse Java-IDE steht über die Homepage des Eclipse-Projektes kostenlos zum Download zur Verfügung. Dabei steht entweder die aktuelle Entwicklungsversion oder ein sog. Stable Build, also eine als stabil eingestufte Minor- oder Major-Release, zur Verfügung. Der Update einer bestehenden Installation erfolgt einfach durch Entpacken des Tar-Archivs im existierenden Programmverzeichnis und ist somit einfach und unkompliziert.

Grundsätzlich deckt die Eclipse Java IDE alle Funktionalitäten ab, die klassische Programmierumgebungen heute bieten (vgl. Abschnitt 2.1). Über ein intelligentes Plugin-Konzept werden inzwischen zahlreiche funktionale Erweiterungen auf allen Ebenen angeboten, die teilweise lizenzpflichtig, teilweise auch kostenlos erhältlich sind. Abbildung 3 zeigt einen Überblick der verfügbaren Funktionalität bezogen auf die Basisinstallation ohne zusätzliche Plugins.

- Runtime & IDE
 - wahlweise verschiedene JDKs (1.2 – 1.4), definierbar auf Projektebene
 - Erweiterbarkeit der IDE-Funktionalität durch Plugin-Architektur
- Editor
 - Syntax Highlighting
 - Semantische Auto-Vervollständigung des Codes
 - Kreuzreferenzfunktionen
 - Einblendung von Java Doc
 - Refactoring
 - automatische Code-Formatierung
- Debugger
 - Hot Code Replace (ab Sun Java SDK 1.4 mit neuer Debugger-API)
 - Enabling / Disabling von Breakpoints inkl. programmierbarer Stoppbedingung
 - keine Profiler-Funktionen (z. B. Zählen von Instanzen)
- Sourcecode-Verwaltung
 - intern über eigene Historie
 - extern über integrierte Anbindung an CVS od. kommerzielle Versionierungssysteme (z. B. Rational, Merant)
- Deployment
 - extern über integrierte Anbindung an Jakarta Ant

ABBILDUNG 3.

Funktionalität der Eclipse Java IDE

Im Bereich der Web-Programmierung mit Java zählen die sog. Java Server Pages (JSPs) und die Java Servlet-API zu den zentralen Basistechnologien, die auch im beschriebenen Projekt zum Einsatz kommen. Hierzu entwickelt das Apache Jakarta Projekt die Referenzimplementierung unter der Bezeichnung Tomcat. Diese Engine kommt heute in vielen Produktionsumgebungen (wie auch in dem hier beschriebenen Projekt unter Abschnitt 1.1) zum Einsatz. Ein entsprechendes Tomcat-Plugin für die Entwicklung von Servlets und JSPs unter Eclipse wird von der französischen

Softwareschmiede sysdeo kostenlos angeboten und soll im folgenden näher betrachtet werden.

3.2.1 Eclipse Tomcat-Plugin

Mit dem Plugin ist der Entwickler in der Lage, eine existierende Tomcat-Installation auf seinem System über die IDE komplett zu steuern. Da die unter Tomcat installierten Anwendungen somit in der Laufzeitumgebung der IDE ausgeführt werden, lassen sich an beliebiger Stelle Breakpoints im Programmcode setzen und die Weiterverarbeitung über den Debugger steuern.

Das funktioniert sogar auf Ebene der Java Codefragmente in den serverseitig ausgeführten JSP-Dateien, die zur Laufzeit von der Tomcat-Engine in Java Servlets übersetzt und ausgeführt werden. Aus Entwicklersicht ist dieser generierte Java-Code jedoch sehr unübersichtlich und nicht intuitiv verständlich. Daher wird z. Zt. an einer Lösung gearbeitet, synchron zum ausgeführten Servlet-Code die entsprechende Stelle in der zugehörigen JSP Datei im Debugger anzuzeigen, die den Bezug zur aktuellen Position in der JSP besser verdeutlicht. Andere IDEs wie z. B. das noch vor Eclipse auf dem Markt erhältliche Visual Age for Java Version 4 boten solche Funktionen bereits für JSP-Entwickler an.

Darüber hinaus kann der Entwickler mit dem Plugin zentrale Konfigurationsparameter (z. B. das verwendete Java Runtime Environment, die zentrale Datei für die Serverkonfiguration) komplett über die IDE und entsprechende Dialoge steuern.

4 Cross-Browser Testing unter Linux

Kopferbrechen bereitet vielen Entwicklern von Web-basierten Anwendungssystemen die Tatsache, daß die unterschiedlichen Browser, allen voran der Internet Explorer von Microsoft und der Netscape Navigator, HTML-Code teilweise sehr unterschiedlich interpretieren und anzeigen.

Noch komplizierter wird die Situation, wenn die Anwendung mit älteren Versionen der gängigsten Browser (und den damit unterstützten HTML-Versionen) kompatibel sein muß. Dies ist i.d.R. der Fall, wenn eine öffentlich über das Internet erreichbare Web-Anwendung entwickelt wird und in diesem Fall nicht von einer eingrenzenden Installationsbasis von Internet-Browsern (wie z. B. bei Intranet-Anwendungen im geschlossenen Unternehmensnetzwerk) ausgegangen werden kann. Der Einsatz von clientseitigen Scripting-Technologien (z. B. JavaScript) wirkt sich ebenfalls negativ auf die Unabhängigkeit der Anwendung von der verwendeten Browser-Software und Version aus. Es ist also unerlässlich, intensiv die Kompatibilität zu den verbreiteten Browsern zu testen.

Unter Linux besteht das Problem, daß der Internet-Explorer für dieses Betriebssystem nicht verfügbar ist, dieser Browser aber zugleich in den letzten Jahren seinen Marktanteil auf über 70% ausbauen konnte. Dahinter rangieren Browser wie Netscape (ca. 20%) und Opera (ca. 8%), die in nahezu allen Versionen unter Linux verfügbar sind. Häufig findet die Entscheidung, wie und für welchen Browser die Ausgabe auf eine Anfrage (HTTP-Request) angepaßt werden soll, über eine serverseitige Logik statt. Über das sog. User-Agent-Spoofing kann ohne großen Aufwand diese Funktionalität auch für die nicht verfügbaren Browser unter Linux getestet werden. Soll die korrekte Darstellung der zurückgelieferten Inhalte unter Linux für reine Mac- oder Windows-Browser überprüft werden, bieten sich nur die bekannten Betriebssystem-Emulationen oder virtuellen Maschinen für das Testing an.

4.1 User Agent Spoofing

Der technisch geprägte Begriff 'User Agent' bezeichnet die beim Benutzer installierte Browser-Software. Die unterschiedlichen Browser übermitteln mit jeder Anfrage an einen Web-Server eine eindeutige Kennung über sich und ihre Version (z. B. 'netscape6' für den Netscape Navigator in der Version 6). Diese Information läßt sich mit der Servlet-API serverseitig auslesen und versetzt damit die Web-Anwendung in die Lage, dynamisch die Ausgabe entsprechend der verwendeten Browser-Software für jeden Request anzupassen. Spoofing meint in diesem Zusammenhang das Vortäuschen eines bestimmten Browsers, obwohl dieser gar nicht vom Benutzer verwendet wird. Dazu wird einfach das User-Agent-Feld im HTTP-Header des Requests auf einen gewünschten Wert gesetzt.

Über diese Manipulation läßt sich das serverseitige Verhalten einfach und ohne großen Aufwand testen. Für den Netscape Navigator bietet beispielsweise die Firma XulPlanet (www.XulPlanet.com) kostenlos eine zusätzliche Leiste für Einstellungen an, über die u.a. eine Auswahl der User Agent Kennung der verbreitetsten Browser (siehe Abschnitt 4) vorgenommen werden kann.

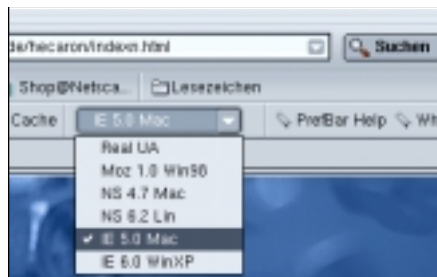


ABBILDUNG 4.

User Agent Spoofing mit der Preference Toolbar von XulPlanet

4.2 Windows Emulation und virtuelle Maschinen

Um die Ausgabe auf einem nicht unter Linux lauffähigen Browser auf dem Entwicklerarbeitsplatz zu testen ist die Installation einer Betriebssystem-Emulation (z. B. wine) oder virtuellen Maschine (z. B. VMWare), in der Windows oder Mac OS ausgeführt werden kann, unumgänglich. Sicherlich stellt dieses Verfahren einen gewissen Installations- und Pflegeaufwand dar und beeinträchtigt die Gesamtleistung des Entwicklungssystems. Produkte wie VMWare sind zudem lizenzpflichtig, haben sich jedoch im täglichen Einsatz im Rahmen des Projektes sehr bewährt.

5 Editoren für HTML & XML

Auch auf diesem Gebiet hat sich das Angebot verfügbarer WYSIWYG ('What You See is What You Get') Editoren stark verbessert. Wer mit der Befehlssyntax von vi auf Kriegsfuß steht, kann inzwischen aus einem reichhaltigen Angebot von Free-

ware-Editoren bis hin zu kommerziellen Profiwerkzeug auswählen. Drei Stellvertreter sollen im folgenden kurz vorgestellt werden.

	Bluefish	Quanta Plus	jEdit
HTML Syntax Highlightning	X	X	X
XML Syntax Highlightning	X		X
JSP Syntax Highlightning			X
PHP Syntax Highlightning	X	X	
Unterstützung für CSS		X	
Assistenten	X	X	
Vorschaufunktion	X	X	
Projektverwaltung	X	X	

ABBILDUNG 5.

HTML- und XML-Editoren unter Linux im Vergleich

5.1 Bluefish

Bluefish steht unter der Gnu Public License (GPL) und kann kostenlos unter bluefish.openoffice.org heruntergeladen werden. Der Editor basiert auf Gtk+ und bietet neben der umfangreichen Funktionalität für HTML auch Unterstützung für WML, um WAP-Seiten für Mobiltelefone zu erstellen. Weiterhin bietet er Syntax Highlighting und die komplette Befehlsreferenz für PHP3/4 und SSI, was ihn sicherlich für Java-/JSP-Entwickler weniger interessant erscheinen läßt. Eine Vielzahl von Dialogen und Assistenten runden das Bild dieses Editors als gelungene Ergänzung zu einer Entwicklungsumgebung ab.

5.2 Quanta

Quanta ist in zwei Versionen erhältlich: Kostenlos in der Variante Plus (quanta.sourceforge.net), und lizenzkostenpflichtig als Quanta Gold (www.thecompany.com). Zu den Stärken in beiden Versionen zählt sicherlich das Syntax Highlighting, das man nach Belieben individuell anpassen kann. Aber auch hier gilt: Der JSP-Entwickler hat gegenüber HTML-, PHP- oder XML-Anhängern das Nachsehen und muß auf eine hilfreiche Visualisierung seiner verwendeten Syntax verzichten. Über eine Vorschaufunktion lassen sich die aktuellen Dokumente begutachten, ohne dazu einen externen Browser starten zu müssen. Die Gold-Variante versucht u.a. mit umfangreicheren Assistenten, Unterstützung für Style Sheets und einem FTP-Client die Aufmerksamkeit der zahlenden Kundschaft auf sich zu lenken.

5.3 jEdit

Im Syntax Highlighting noch nicht vollständig überzeugend, aber im Ansatz nicht schlecht und obendrein kostenlos bietet jEdit (www.jedit.org) seine Dienste Java-, JSP- und XML-Entwicklern an. Der in Java erstellte Editor ist im Vergleich zu Bluefish und Quanta eher als Purist einzustufen, bietet aber dem serverseitig orientierten Entwickler alle notwendigen Funktionalitäten beim Editieren von Dokumenten außerhalb der IDE, ohne dazu den unnötigen Overhead großer HTML-Editoren mitzuschleppen.

6 Deployment

Die Automatisierung des Build-Prozesses (engl. Deployment) übernimmt im beschriebenen Projekt das Java-basierte Werkzeug Ant (dt. Ameise). Es ist als Teilprojekt unter dem Apache Jakarta Projekt (jakarta.apache.org) organisiert und eignet sich insbesondere für den Einsatz in Web-Projekten. Ant leistet im Rahmen des Projektes zuverlässige Dienste bei der Erstellung von Builds für die unterschiedlichen Zielsysteme (Entwicklung, Test & Produktion).

Basis jeden Build-Laufs ist die XML-Konfigurationsdatei `build.xml`, die alle notwendigen Informationen bereithält. Sie beschreibt die einzelnen Operationen eines Build-Laufs, die auch als Tasks bezeichnet werden. Es existieren viele fertige Standard-Tasks (wie z. B. (File-)Copy, Delete, Compile, FTP, Token-Replace etc.), die um benutzereigene Aufgaben mit Java erweitert werden können. Wie bei anderen Build-Werkzeugen (z. B. `make`) auch lassen sich Abhängigkeiten zwischen den einzelnen Operationen festlegen und einzelne Aufgaben innerhalb so genannter Targets zu Gruppen zusammenfassen. Für eine periodische Ausführung der Build-Tasks mit Ant bieten sich unter Linux betriebssystemnahe Lösungen wie z. B. Cron Jobs an.

Im konkreten Projekt ist Ant so konfiguriert, daß beim Aufruf über den Parameter 'system' ein bestimmtes Zielsystem definiert wird, für das der Build-Lauf angestoßen wird. Mit dieser Information wird eine sog. Property-Datei, bestehend aus Key-Value-Paaren, eingelesen, die spezifische Werte für das Zielsystem (z. B. IP-Adressen oder die URL) beinhaltet. Ant ersetzt die Platzhalter in Konfigurations- und JSP-Dateien mit diesen Werten und baut mit einem im Standard-Funktionsumfang bereits vorhandenen Task ein sog. Web-Archiv (.WAR-Datei, siehe Abbildung 6), das zur Ausführung nur noch in das Tomcat-Anwendungsverzeichnis kopiert werden muß.

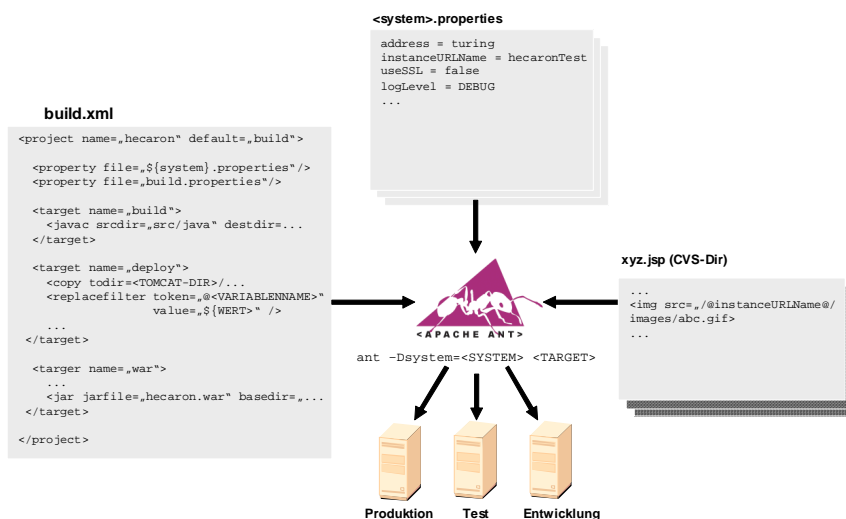


ABBILDUNG 6.

Build-Prozeß eines WAR-Archivs der Web-Anwendung mit Jakarta Ant

7 Fazit

Linux kann als ausgereifte Plattform für die Entwicklung von komplexen Web-Anwendungen auf Basis von J2EE eingesetzt werden. Insbesondere die Investitions- und Betriebskosten können unter Linux deutlich unter den Ausgaben für eine vergleichbare Windows-basierte Entwicklungsumgebung bei geschickter Auswahl der Komponenten liegen. Vorbei sind auch die Zeiten, wo professionelle Tools für Entwickler Mangelware unter Linux waren. Inzwischen sind die populärsten Open- und Closed-Source IDEs für das offene Betriebssystem im gleichen Releasestand wie für die Windows-Version erhältlich. Es gibt allerdings gerade bei der Anwendungsentwicklung für Java-basierte Web-Applikationen noch immer auf einigen Bereichen Nachholbedarf. Am wichtigsten zu nennen ist

- JSP-Debugging & Editing
- Performance / Load Testing
- Java VM-Profilng

Hier kann Linux (noch) nicht mit dem Umfang und/oder der Qualität vergleichbarer Entwicklungswerkzeuge unter Windows gleichziehen. Nicht gerade verwunderlich aber dennoch bedauerlich ist die Tatsache, daß professionelle Entwicklungstools für .NET und C# unter Linux noch sehr eingeschränkt verfügbar sind. Allerdings ist auch hier die Eclipse-Gemeinde schon aktiv geworden und hat mit dem Plugin "Improve C#" der Firma Improve Technologies (www.improve-technologies.com) erste erfolgreiche Schritte in die richtige Richtung unternommen.

Linux als Entwicklungsumgebung ist uneingeschränkt empfehlenswert für größere Projektteams, die i.d.R. auch komplexe Web-Anwendungen entwickeln wollen. Insbesondere beim gleichzeitigen Einsatz des Betriebssystems in der Produktionsumgebung hilft eine homogene Infrastruktur, unnötige Komplexität und Kosten zu vermeiden und letztlich den Entwicklungsprozeß zu beschleunigen. Ein tiefgehendes Verständnis von Linux/Unix bei den Programmierern ist dank der sehr guten grafischen Oberflächen und Tools heute nicht mehr erforderlich, um erfolgreich unter Linux Software zu entwickeln.

8 Online-Quellen

Eclipse	http://www.eclipse.org
Sysdeo Tomcat Plugin	http://www.sysdeo.com/eclipse/tomcatPlugin.html
JOnAS Plugin	http://sourceforge.net/projects/jonaseclipse/
Jakarta Ant	http://jakarta.apache.org/ant/
CVS	http://www.cvshome.org
Cervisia (CVS-Client)	http://cervisia.sourceforge.net/
Bluefish	http://bluefish.openoffice.nl
Quanta	http://quanta.sourceforge.net
IBM WebSphere Appl. Develop.	http://www-4.ibm.com/software/ad/studioappdev/
JBuilder Personal Edition	http://www.borland.com/jbuilder/personal/
C#- und weitere kostenlose Plugins	http://www.improve-technologies.com
Netscape Preference Toolbar	http://www.xulplanet.com
