

# Java-Trendbarometer

Trends und Entwicklungen in den Java-Technologien

**Dezember 2008**

## Inhaltsverzeichnis

|       |  |    |
|-------|--|----|
| 1     | Das Java-Trendbarometer .....                  | 3  |
| 2     | Verwendung der Java-Technologien .....         | 5  |
| 3     | Werkzeuge .....                                | 9  |
| 4     | Technologien und Frameworks .....              | 12 |
| 4.1   | Bewertung .....                                | 14 |
| 5     | Open Source .....                              | 15 |
| 5.1   | Open Source Produkte .....                     | 16 |
| 5.1.1 | Bedeutende Produkte .....                      | 16 |
| 5.1.2 | Unbekannte Produkte und Trends .....           | 16 |
| 6     | Software-Entwicklungsprozess .....             | 19 |
| 6.1   | Projektvorgaben zur Entwicklungsumgebung ..... | 19 |
| 6.2   | Kontinuierliche Integration .....              | 20 |
| 6.3   | Anforderungsanalyse .....                      | 21 |
| 6.4   | Entwicklerdokumentation .....                  | 22 |
| 6.5   | Qualitätssicherung .....                       | 23 |
| 7     | Innovation und Weiterbildung .....             | 24 |
| 7.1   | Bewertung .....                                | 28 |
| 8     | Fazit .....                                    | 30 |

Die Inhalte dieser Studie wurden mit Sorgfalt zusammengestellt. Gleichwohl übernimmt der Anbieter keine Haftung für die Richtigkeit und Vollständigkeit der bereitgestellten Informationen. Die Inhalte externer Links unterliegen der Haftung der jeweiligen Anbieter.

Alle genannten Firmennamen und Marken sind Eigentum ihrer Inhaber.

expeso GmbH, Weinheimer Str. 68, D-68309 Mannheim  
Telefon: +49 (621) 7249344-0  
E-Mail: [info@expeso.de](mailto:info@expeso.de), Internet: <http://www.expeso.de>  
Amtsgericht Mannheim, HRB 704028, Geschäftsführer: Markus Roth

# 1 Das Java-Trendbarometer

Java-Technologien haben für Unternehmensanwendungen eine große Bedeutung. Umso erstaunlicher ist es, dass es bisher noch keine systematische Trendstudie für die Praxis gibt. Diese Lücke schließen wir mit dem Java-Trendbarometer.

Dadurch sollen aktuelle Trends und Entwicklungen frühzeitig erkannt werden; der Fokus liegt bei der Relevanz für die praktische Arbeit in den Unternehmen.

Mit dem Java-Trendbarometer zeigt expeso IT-Verantwortlichen und Java-Experten, wohin der Trend geht, was es Neues gibt und welche Technologien schon reif für den Einsatz sind. Gerade bei den Open Source Frameworks gibt es ständig neue und interessante Entwicklungen. Darüber hinaus werden auch längst bekannte Probleme aufgezeigt, die leider immer noch nicht behoben sind.

Um aussagekräftige Ergebnisse mit einem großen Praxisbezug zu erhalten, befragten wir ausschließlich Java-Experten, die in der Praxis mit den Technologien befasst sind. So ist sichergestellt, dass sich die empfohlenen Technologien auch schon in der täglichen Projektarbeit bewährt haben.

Wir werden dieses Trendbarometer in Zukunft zwei Mal jährlich, im Frühjahr und im Herbst fortsetzen, um auch weiterhin über neue Entwicklungen berichten zu können. Vor diesem Hintergrund ist das erste Java-Trendbarometer die Basismessung, die verschiedenen Entwicklungen und Trends können erst mit den folgenden Erhebungen quantifiziert werden.

## Zusammenfassung

Das Java Trendbarometer zeigt systematisch die Verbreitung von Technologien, Frameworks und Open Source Produkten. Erstaunliche Ergebnisse ergaben sich beim Thema Software-Entwicklungsprozess (Kapitel 6). Hier sind altbekannte Probleme nach wie vor in der Praxis vorhanden – trotz entsprechender Tools und Erfahrungen. Erstaunlich sind weiterhin die Mängel in Innovation und Weiterbildung, sprich der Auseinandersetzung mit neuen Technologien (Kapitel 7).

Neben der Bestandsaufnahme zeigt das Java-Trendbarometer auch, in welchen Bereichen eine externe Unterstützung durch ausgewählte Java-Experten hilfreich ist – sei es in der Software-Entwicklung, dem Technologie-Coaching oder der Technologie-Beratung.

## Inhalt

In den Kapitel 2 bis 4 analysieren wir den Einsatz von Java-Technologien, Werkzeuge, Technologien und Frameworks in Projekten. In Kapitel 5 zeigen wir Ihnen dann, welche Open Source Tools und Produkte die Experten im Alltag einsetzen. Vielleicht gibt es da auch für Sie noch die ein oder andere „Perle“ zu entdecken. Und in den letzten zwei Kapiteln verlassen wir die Bits und Bytes und betrachten die mit der Java-Technologie eng verbundenen Themen Software-Entwicklungsprozess, Innovation und Weiterbildung.

## Registrieren Sie sich

Damit Sie die Ergebnisse unserer Studie auch in Zukunft erhalten, können Sie sich gerne bei uns registrieren. Eine formlose E-Mail an [trend@expeso.de](mailto:trend@expeso.de) genügt.

## Über expeso

Die Mannheimer expeso GmbH ist der Dienstleister für Technologie-Beratung, Technologie-Coaching und Software-Entwicklung. Der Schwerpunkt der hochkarätigen expeso-Experten sind Java und Java Enterprise Technologien für unternehmenskritische Software-Projekte. Durch die eigenen langjährigen Erfahrungen in IT-Projekten, dem Kompetenzteam Java und dem neuartigen System von Skill-Scouts ist expeso jederzeit in der Lage, Projekte kurzfristig mit qualifizierten Fachleuten zu besetzen. Dadurch können fehlende Ressourcen - etwa bei personellen Engpässen oder der Einführung von neuen Technologien - gezielt ergänzt werden.

Durch die kompetente Vorauswahl sowie den Pool von hochqualifizierten Experten sparen Unternehmen und Softwarehäuser gleich doppelt Zeit und Geld – sowohl bei der Bewerbersuche und –auswahl als auch bei der Implementierung von unternehmenskritischen Projekten.

Arbeitsbereiche von expeso sind:

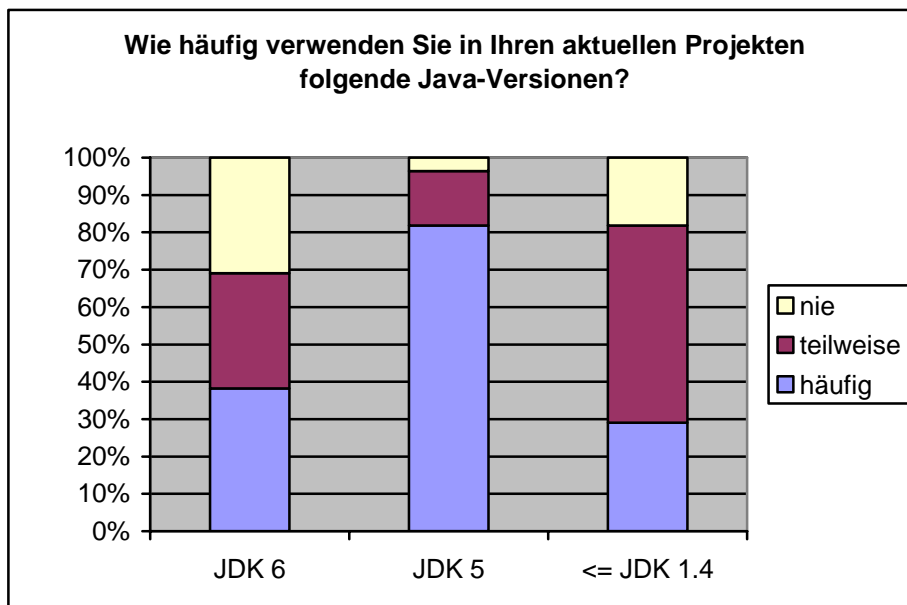
- **Projektarbeit** - Unterstützung in allen Phasen der Software-Entwicklung
- **Coaching** - Know-how Transfer beim Einsatz neuer und innovativer Technologien.
- **Technologie-Beratung** - Evaluation von neuen Technologien, Begutachtung von Architektur und Code (Reviews)

Weitere Informationen finden Sie im Internet unter [www.expeso.de](http://www.expeso.de).

## 2 Verwendung der Java-Technologien

Die **Java-Sprache** ist innovativ und wird fortwährend weiter entwickelt. In der freien Online-Enzyklopädie Wikipedia sind die Veröffentlichungen seit 1996 dokumentiert ([http://de.wikipedia.org/wiki/Java\\_\(Technik\)](http://de.wikipedia.org/wiki/Java_(Technik))).

Obwohl die heute aktuelle Version 6 nun schon seit zwei Jahren verfügbar ist, gibt es trotzdem noch eine Vielzahl von Projekten in Unternehmen, die noch nicht auf diese aktuelle Version umgestellt wurden. Abbildung 1 zeigt, wie häufig die verschiedenen Java-Versionen heute im Einsatz sind. Dabei ist zu sehen, dass die vorletzte Version (JDK 5 von 2004) immer noch die größte Verbreitung hat und häufig eingesetzt wird. Auch die noch älteren Versionen (JDK 1.4 und früher) sind bei 80% der Befragten immer noch teilweise oder sogar häufig im Einsatz. Die aktuelle Version 6 wird wohl hauptsächlich bei neuen Projekten eingesetzt.



**Abbildung 1: Aktuelle Verwendung der Java-Technologien**

Der Hauptgrund dürfe in dem nicht zu unterschätzenden Migrationsaufwand beim Umstieg auf eine neue Java-Version liegen. Eine Strategie könnte daher sein, Inkompatibilitäten zu identifizieren und den Umstieg auf eine neue Java-Version langfristig zu planen. Die Migration kann dann schrittweise erfolgen, in dem in den eigentlichen Release-Plan die notwendigen Maßnahmen nach und nach mit eingeplant werden.

So können z.B. inkompatible Drittbibliotheken schrittweise mit neueren Versionen ersetzt oder ganz ausgetauscht werden. Nach diesen Vorbereitungen erfolgt dann erst im letzten Schritt der Umstieg auf die neue Java-Version.

Inwieweit sich ein Umstieg rechnet und welche Maßnahmen notwendig sind, muss im Einzelfall überprüft werden. Bei der Ausarbeitung einer entsprechenden Migrationsstrategie bietet sich eine Technologie-Beratung

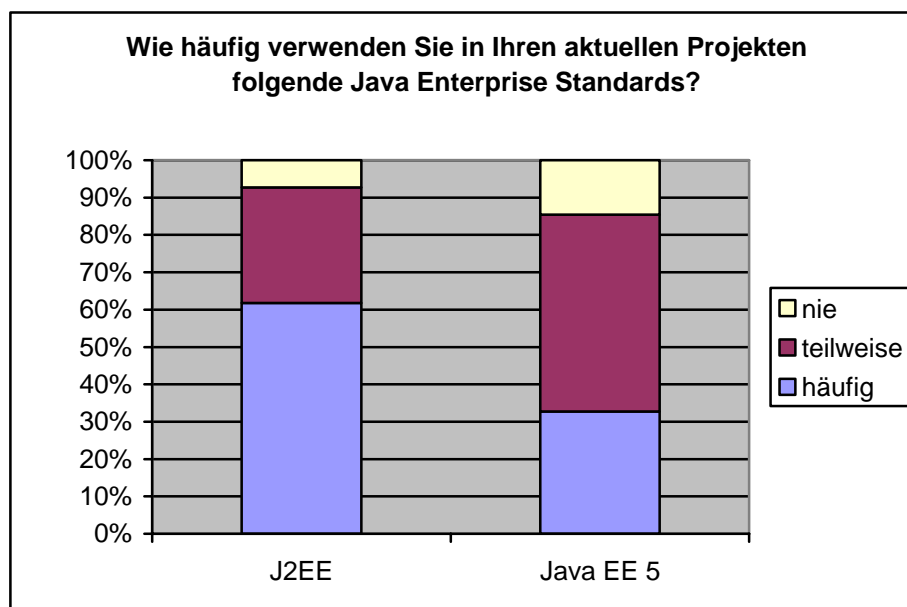
durch externen Java-Experten an, die schon Erfahrung mit den neuen Technologien haben.

Mit der **Java Platform Enterprise Edition (Java EE)** steht eine Software-Architektur zur Entwicklung verteilter, mehrschichtiger Anwendungen zur Verfügung. Dabei bezeichnet J2EE die Versionen 1.0 (Dezember 1999) bis 1.4 (November 2003).

Mit der Version Java EE 5 (Mai 2006) folgte dann ein deutlich überarbeiteter Standard, der sehr stark durch zwei Open Source Produkte beeinflusst wurde, die sich fast schon zum Defacto-Standard etabliert hatten: Das Spring Framework als leichtgewichtige Enterprise Plattform (<http://www.springframework.org>) revolutionierte die Art und Weise, mit der Enterprise Anwendungen entwickelt wurden. Und Hibernate etablierte sich als Persistenzlösung (<http://www.hibernate.org>).

Abbildung 2 zeigt, dass auch heute noch J2EE die Projektwelt dominiert. Das liegt vermutlich auch daran, dass alte Anwendungen sich nur mit erheblichem zeitlichen und finanziellen Aufwand auf die neue Version umstellen lassen. Vor einer solchen Umstellung müssen daher immer auch die Wirtschaftlichkeit (z.B. ROI durch Kosteneinsparung in der Weiterentwicklung und Wartung), Risiken und die zeitliche Verzögerung im Release-Plan der eigenen Anwendung betrachtet werden.

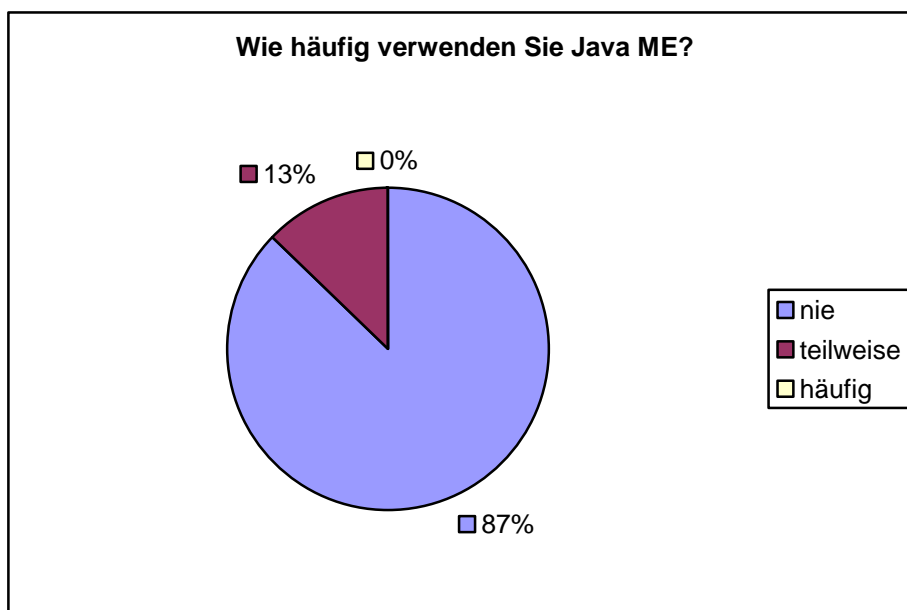
Bei einem notwendigen Umstieg sind die expeso-Experten gerne behilflich. Durch die externen Spezialisten kann ein entsprechendes Projekt schneller und effizienter realisiert werden, in dem Einarbeitungszeiten minimiert und Fehler vermieden werden.



**Abbildung 2: Aktuelle Verwendung der Java Enterprise Standards**

Eine Nebenfrage (siehe Abbildung 3) nach dem Einsatz der **Java Micro Edition (Java ME)** für Mobiltelefone und andere mobile Endgeräte zeigt, dass diese eine geringe Rolle in der Entwicklung von Anwendungen im Unternehmensumfeld spielt.

Dies dürfte sich – nicht zuletzt im Zuge der verbesserten mobilen Bandbreite und der Verfügbarkeit leistungsfähiger Endgeräte – ändern. Dies gilt gerade für die engere Einbindung mobiler Mitarbeiter in die IT-gestützten Geschäftsprozesse. Nach Einschätzung des Zukunftsinstitutes (Kelkheim) sind die entsprechenden Trends bereits erkennbar. Daher ist es gut und wichtig, sich frühzeitig auf diese Besonderheiten einzustellen.



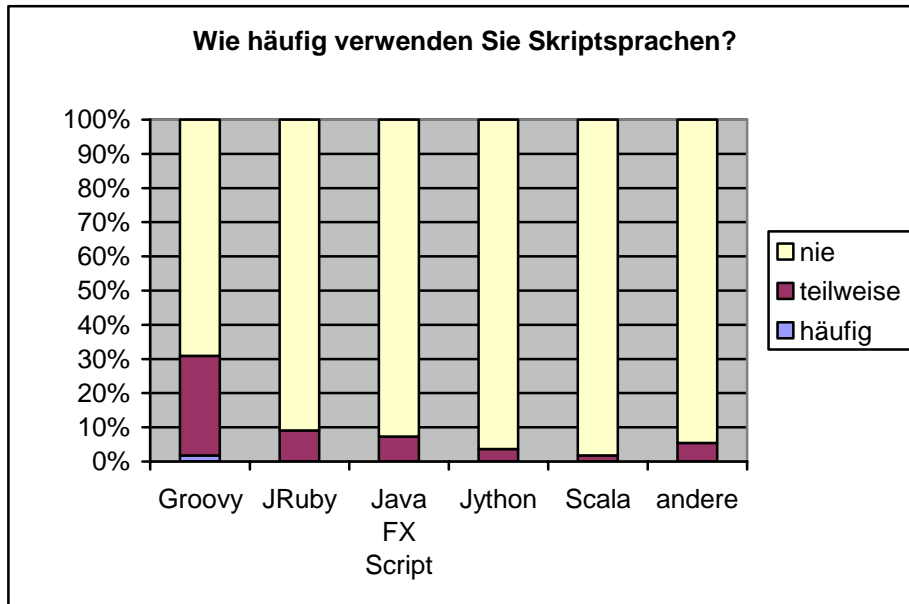
**Abbildung 3: Verwendung Java ME**

Seit 2007 treten vermehrt auch **Skriptsprachen** auf der Basis der Java Virtual Machine (JVM) in Erscheinung. Abbildung 4 zeigt, dass diese noch keine hohe Verbreitung haben. Sie werden definitiv noch nicht häufig eingesetzt. Groovy liegt in dieser Statistik klar vorne – sie wird von ca. 30% der Befragten teilweise eingesetzt. Die Entwicklung in diesem Umfeld bleibt sicher spannend. Interessant wird auch sein, in welchen Bereichen diese Skriptsprachen sich bewähren werden.

Erste eigene Erfahrungen werden durch die enge Integration der Skriptsprachen mit Java erleichtert. So können einzelne Funktionen in aktuellen Projekten mit Skriptsprachen realisiert werden ohne das gesamte Projekt zu beeinflussen. Solche Skripte können gegebenenfalls wieder recht einfach entfernt werden. Sollte sich die Skriptsprache im Projekt jedoch bewähren, kann deren Einsatz schrittweise ausdehnt werden.

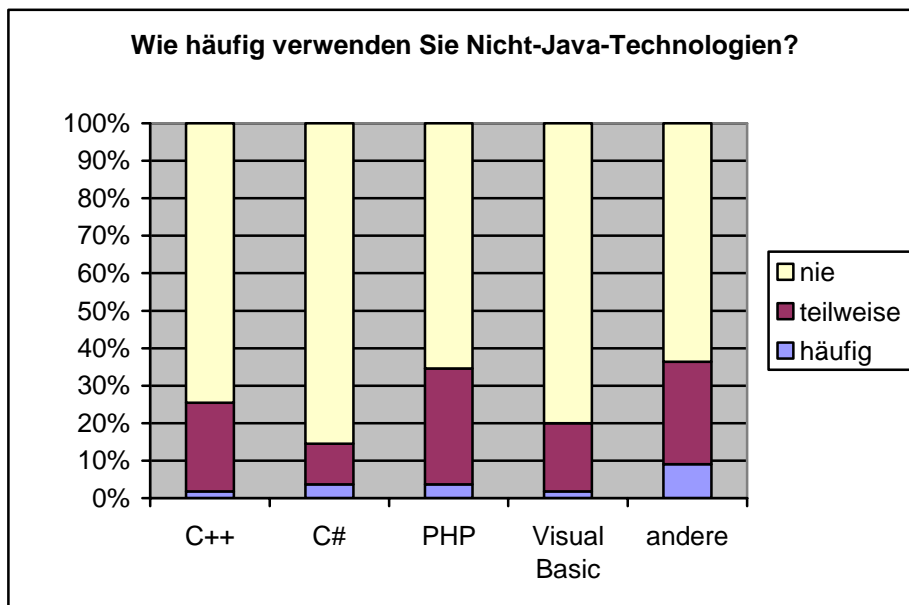
Es empfiehlt sich, geeignete Bereiche für die ersten Gehversuche zu identifizieren. So können beispielsweise mit Canoo Webtest webbasierte

Unit-Tests entwickelt werden. Auch der Einsatz in Prototypen oder als sogenannter Gluecode ist denkbar.



**Abbildung 4: Verwendung von Skriptsprachen**

Aufschlussreich war auch die Frage, wie häufig Java-Experten **Nicht-Java Technologien** einsetzen. Abbildung 5 zeigt, dass solche Technologien - wenn überhaupt - nur am Rande eingesetzt werden.



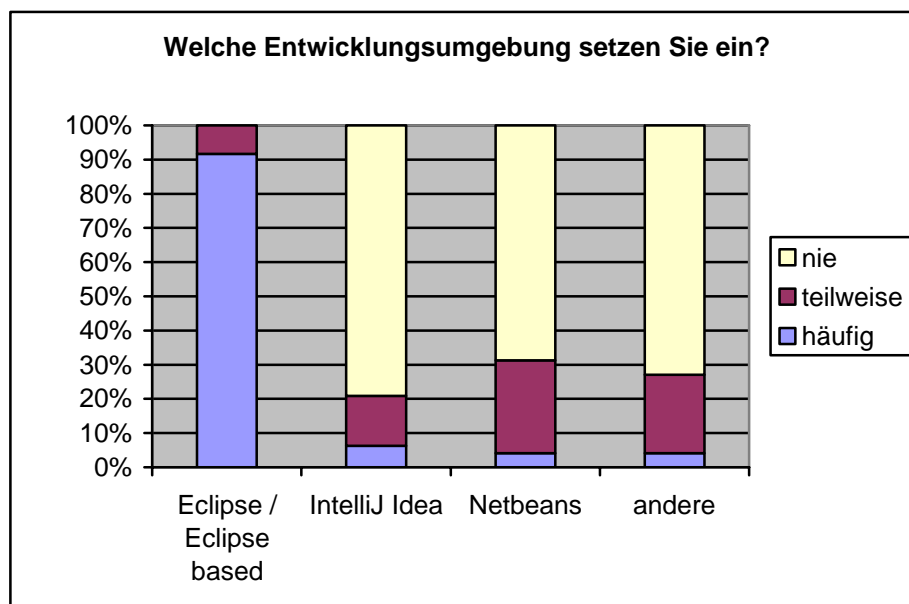
**Abbildung 5: Verwendung von Nicht-Java-Technologien**

### 3 Werkzeuge

Eng mit den Java-Technologien verbunden sind auch die Software-Entwicklungswerkzeuge und der Software-Entwicklungsprozess.

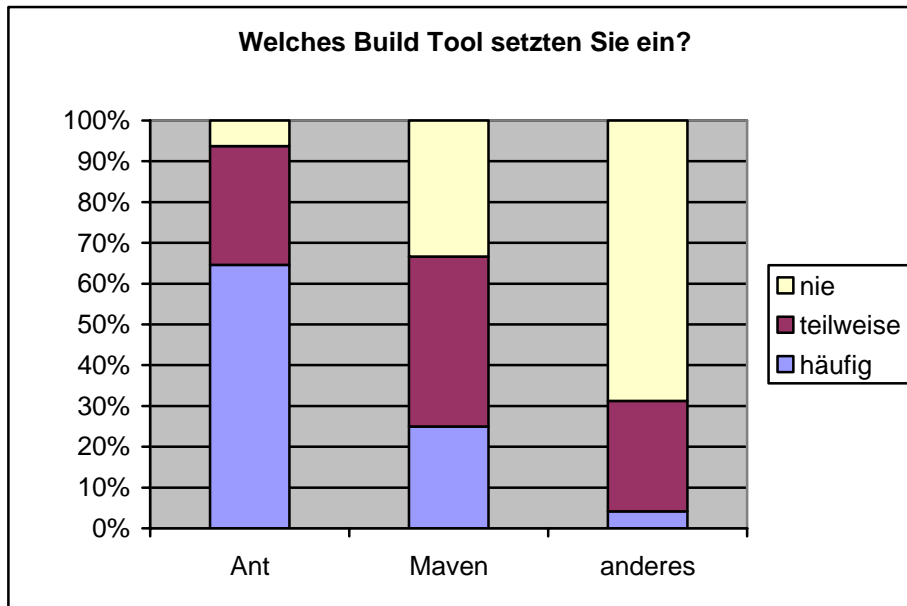
Abbildung 6 zeigt, dass praktisch alle Befragten Eclipse (oder eine auf Eclipse basierende Entwicklungsumgebungen) als **Entwicklungs-umgebung** einsetzen. Damit sind diese Werkzeuge heute der geltende Standard.

Im Vergleich zwischen IntelliJ Idea und Netbeans hat IntelliJ Idea mehr Anhänger, die es regelmäßig („häufig“) einsetzen. Netbeans hat aber die Nase vorn, wenn man die Zahl der Entwickler betrachtet, die es ab und an („häufig“ oder „teilweise“) einsetzen.



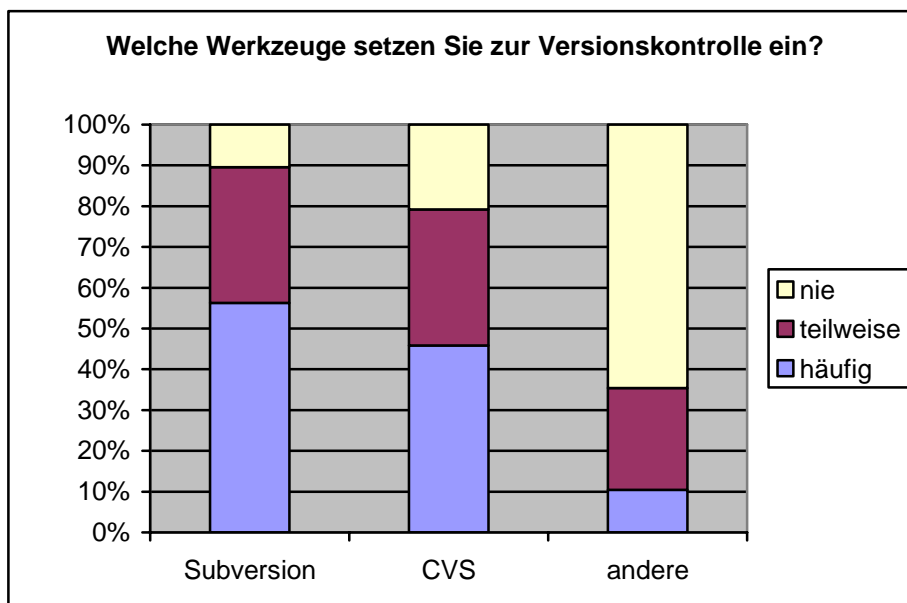
**Abbildung 6: Entwicklungsumgebungen**

Bei den **Build Tools** hat immer noch Ant die höhere Verbreitung. Das neuere Maven (in Version 1 seit Ende 2004 verfügbar) wird aber auch schon von zwei Drittel der Entwickler häufig oder zumindest teilweise eingesetzt. Damit geht der Trend zu Maven – die weitere Entwicklung werden wir in den kommenden expeso Java-Trendbarometern gezielt beobachten.



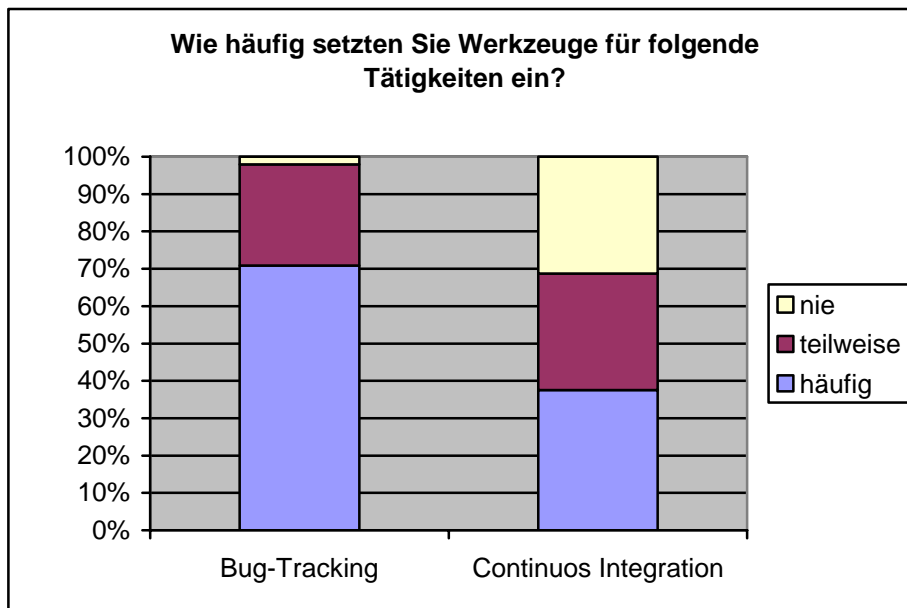
**Abbildung 7: Build Tools**

Subversion ist zwischenzeitlich das beliebteste System zur **Quellcode-Verwaltung** und hat damit das ältere CVS schon überholt. Abbildung 8 zeigt aber, dass der „Vorsprung relativ klein ist und auch CVS von fast 80% der Entwickler noch häufig bis teilweise eingesetzt wird. In der Praxis haben somit beide Systeme wohl noch in etwa gleiches Gewicht. In Zukunft dürfte aber Subversion als das neuere System mehr Gewicht erhalten.



**Abbildung 8: Source Code Verwaltung**

Werkzeuge zur **Fehlerverfolgung** werden sehr häufig eingesetzt (Abbildung 9). Vermutlich auch deshalb, weil es in Teams mit mehreren Mitarbeitern oft gar nicht anders sinnvoll zu handhaben ist. **Kontinuierliche Build und Integration Systeme** sind dagegen weniger verbreitet.



**Abbildung 9: Weitere Werkzeuge**

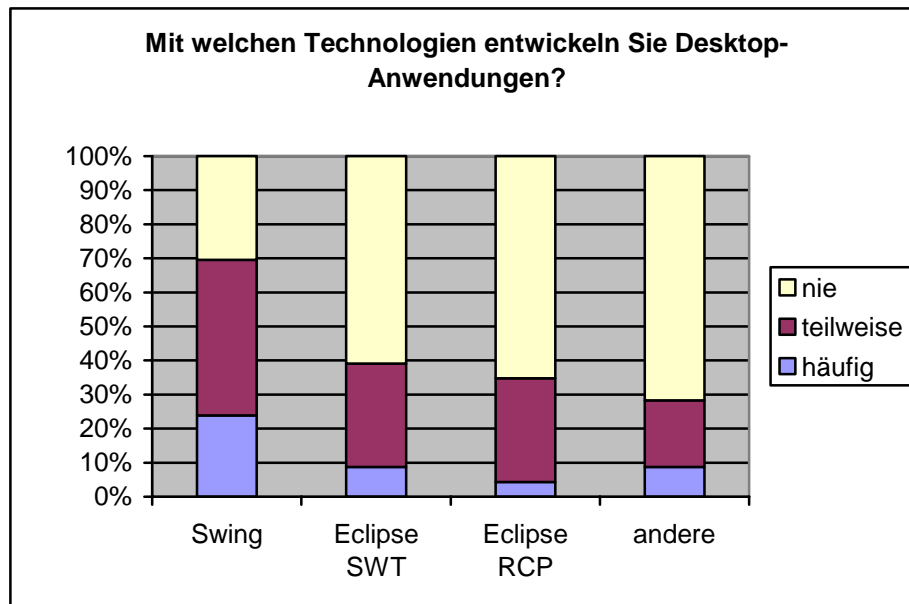
Damit zeigt es sich, dass sich die entsprechenden Tools bei der Fehlerverfolgung im Projektalltag etabliert haben. Dagegen ist der Einsatz einer Integrationsumgebung nur bei 2 von 3 Projekten im Blick. Fachlich ist eine kontinuierliche Integration jedoch empfehlenswert, um mögliche Fehler frühzeitig entdecken zu können. Um Einarbeitungszeiten zu vermeiden und die Lernkurve zu beschleunigen, sollte bei fehlender Erfahrung mit diesen Tools auf eine externe Unterstützung in Form einer Technologie- sowie Projekt-Beratung zurückgegriffen werden. Dies spart Zeit und Geld.

## 4 Technologien und Frameworks

In der Java-Welt gibt es eine Vielzahl von Frameworks, die einem Entwickler das Leben erleichtern. So setzt man z.B. Frameworks für eine Entwicklung nach dem MVC Pattern ein, abstrahiert die Persistenzschicht mit einem Objekt Relationalen Mapper (ORM) und setzt AJAX Bibliotheken für die Gestaltung komfortabler Benutzeroberflächen ein.

Aus dem Projektalltag sind solche Frameworks nicht mehr wegzudenken. Somit gehören diese Technologien essentiell zum Bereich der Software-Entwicklung mit Java.

Abbildung 10 zeigt, dass für die Entwicklung von **Desktop-Anwendungen** häufig noch Swing eingesetzt wird. Vermutlich wird bei Neuentwicklungen aber auch häufig schon eher auf eine neuere Technologie gesetzt.



**Abbildung 10: Frameworks für Desktop-Anwendungen**

Bei **Web-basierten Anwendungen** (Abbildung 11) kommen häufig Spring, Ajax und JSF zum Einsatz. Aber auch Struts und Eigenentwicklungen sind noch relativ häufig im Einsatz. Wicket und JBoss Seam sind beispielsweise nicht so stark vertreten.

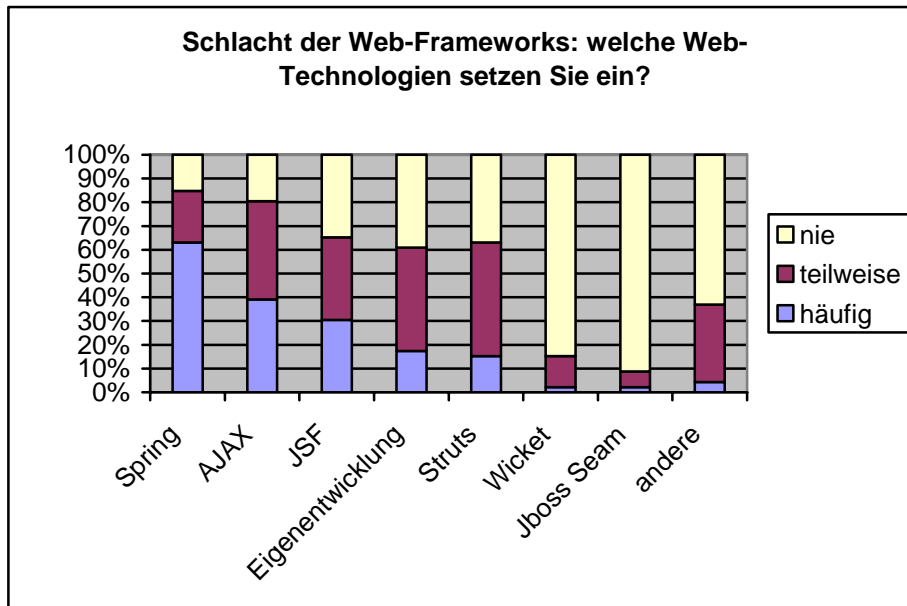


Abbildung 11: Frameworks für Web-Anwendungen

Abbildung 12 zeigt dass für die Entwicklung von **Rich Internet Applications** meist herkömmliches DHTML und Javascript zum Einsatz kommen. Andere Technologien folgen mit deutlichem Abstand.

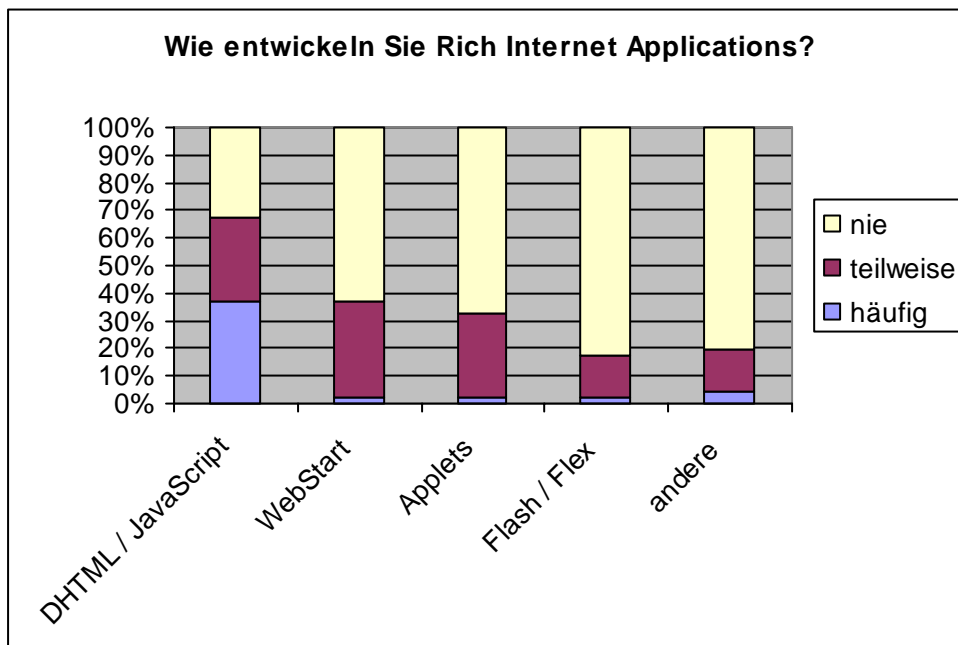
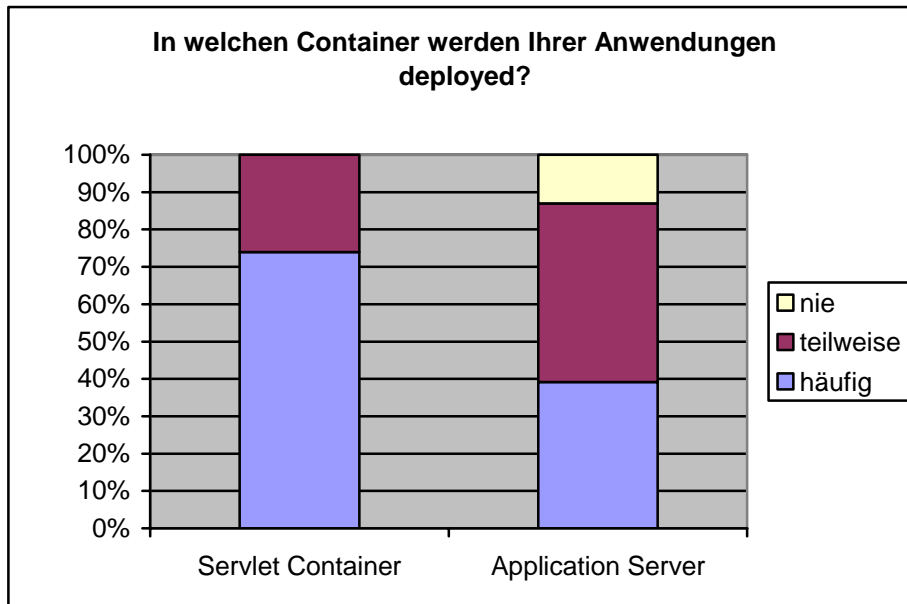


Abbildung 12: Frameworks Rich Internet Client

Als **Container** werden bei Server-Entwicklungen tendenziell eher Servlet-Container wie beispielsweise Tomcat statt vollwertiger Application Server eingesetzt. Aber auch letztere sind häufig im Einsatz (Abbildung 13).



**Abbildung 13: Verwendete Container**

Die Praxis zeigt, dass auch sehr große und geschäftskritische Systeme heute schon zuverlässig auf Servlet Containern betrieben werden. Gerade bei neuen Projekten sollte diese Möglichkeit geprüft werden – hier kann eventuell auch ein externer Fachmann seine Erfahrung mit schon realisierten Systemen einbringen.

## 4.1 Bewertung

In unserer Umfrage baten wir auch um Anmerkungen und Bewertungen zum Einsatz von Frameworks in der Software-Entwicklung.

Ein Experte sieht es als wesentlich für den mittel- und langfristigen Erfolg von Projekten an, dass der Einsatz von Frameworks im Projekt standardisiert und dokumentiert wird. Auch beim Einsatz eines Frameworks sind die Möglichkeiten, wie dieses konkret verwendet werden kann, noch sehr vielfältig. Mit einem standardisierten Vorgehen können Einarbeitungszeiten minimiert und die Wartbarkeit erhöht werden.

Andere merkten an, dass viel Unternehmen enorm in Eigenentwicklungen investiert haben und nun nicht mehr im erforderlichen Maß in deren Weiterentwicklung investieren (können). So stehen neue Möglichkeiten und Ansätze nicht zur Verfügung. Sie wünschen sich eine Trennung von Infrastruktur und unternehmensbezogenen Software-Aspekten. Erstere könnten dann mit etablierten Frameworks verwendet werden.

Und schließlich wurde auch angemerkt, dass es viele Frameworks gibt ... zu viele. Hier ist wohl jedes Unternehmen gefordert, sich für eine Auswahl zu entscheiden und keinen Wildwuchs zuzulassen.

## 5 Open Source

Im Bereich der Software-Entwicklung mit Java spielen Open Source Technologien eine sehr große Rolle. Am Einfluss der Produkte Spring Framework und Hibernate auf den Java EE 5 Standard zeigt sich, dass deren Innovationskraft sehr stark sein kann.

Die Bedeutung und Qualität der eingesetzten Open Source Produkten wird von unseren Befragten entsprechend auch als sehr hoch eingeschätzt (vgl. Abbildung 14 und Abbildung 15).

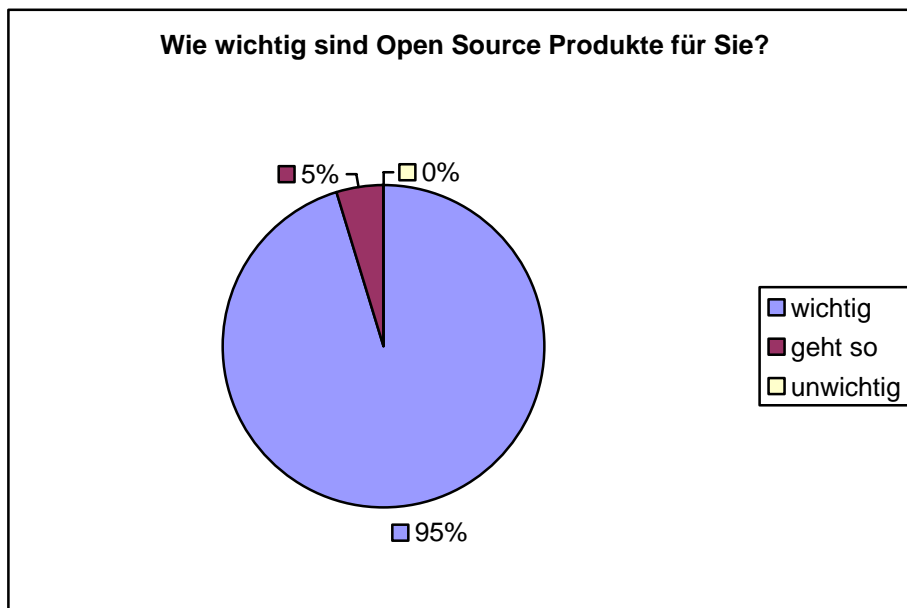


Abbildung 14: Bedeutung von Open Source Produkten

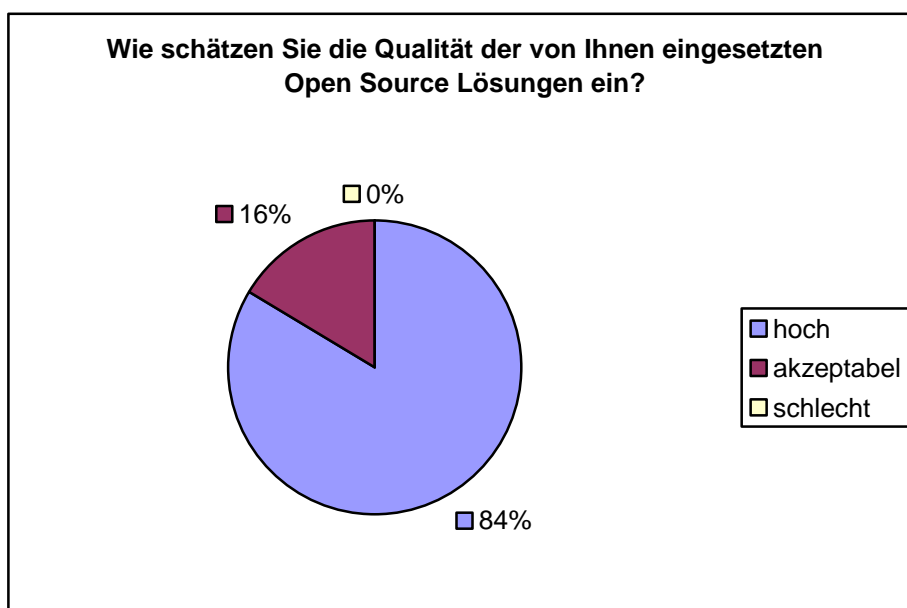


Abbildung 15: Qualität von Open Source Produkten

Open Source Produkte sind – dies belegen die Ergebnisse des Java-Trendbarometers eindeutig – inzwischen definitiv etabliert; sie haben sich bewährt und bieten gerade bei neuen Projekten interessante Möglichkeiten.

## 5.1 Open Source Produkte

### 5.1.1 Bedeutende Produkte

Uns interessierte auch, welche Open Source Produkte die Experten konkret einsetzen und als bedeutend erachten. Dabei wurde am häufigsten das **Spring Framework** genannt.

**Eclipse** wurde fast genauso häufig genannt. Dabei wurden in Einzelfällen dann auch die Teilprojekte **Equinox** (OSGI Framework), **Rich Client Platform** und **BIRT** (Business Intelligence and Reporting Tools) hervorgehoben.

Und auch die Produkte der **Apache Software Foundation** wurden oft genannt, [gleiches gilt für Tomcat](#). Häufig wurden aber auch **Commons** und **Ant** genannt. Einzelnennungen gab es zu **Lucene** (Suchmaschinentechnologie), **Log4J**, **POI** (Zugriff auf MS Office Dokumente) und **Ivy** (Dependency Management).

Ebenso wurde immer wieder **Hibernate** aus dem Hause JBoss genannt. Ab und an wurde von diesem Hersteller auch der **JBoss Application Server** und in Einzelfällen auch **JBPM** (Business Process Management) und **Seam** (Application Platform) genannt.

### 5.1.2 Unbekannte Produkte und Trends

Befragt nach Open Source Produkten, die zurzeit eher unbekannt sind oder auch demnächst wichtig werden könnten, wurden eine Vielzahl von Produkten genannt. Auffällig ist, dass Produkte in drei Bereichen besonders häufig genannt werden:

- Entwicklung komplexer Web-Frontends
- Einsatz von OSGI
- Skript-Sprachen, in der Hauptsache Groovy

Im Folgenden finden Sie eine Liste mit genannten Technologien.

Profitieren Sie von den Empfehlungen der Experten und schlagen Sie auch mal auf den verlinkten Projektseiten nach. Vielleicht entdecken Sie die eine oder andere Perle, die auch Sie für Ihre Projekte evaluieren könnten.

| Open Source Produkt   | Stichwort                                 |
|---|---|
| Apache Ant Ivy<br><a href="http://ant.apache.org/ivy/">http://ant.apache.org/ivy/</a>   | Dependency Management                     |
| Apache Camel<br><a href="http://activemq.apache.org/camel/">http://activemq.apache.org/camel/</a>   | Enterprise Integration Framework          |
| Apache Geronimo<br><a href="http://geronimo.apache.org/">http://geronimo.apache.org/</a>  | Server Framework und Application Server   |
| Apache Jackrabbit<br><a href="http://jackrabbit.apache.org/">http://jackrabbit.apache.org/</a>  | Content Repository                        |
| Apache JMeter<br><a href="http://jakarta.apache.org/jmeter/">http://jakarta.apache.org/jmeter/</a>  | Load Tests                                |
| Apache Lucene<br><a href="http://lucene.apache.org/">http://lucene.apache.org/</a>  | Suchmaschinen-Technologie                 |
| Apache MyFaces<br><a href="http://myfaces.apache.org/">http://myfaces.apache.org/</a>   | JSF Implementierung                       |
| Apache POI<br><a href="http://poi.apache.org/">http://poi.apache.org/</a>   | Zugriff auf MS Office Dokumente           |
| Apache Sling<br><a href="http://incubator.apache.org/sling">http://incubator.apache.org/sling</a>   | Web Framework and Content Management      |
| Apache Wicket<br><a href="http://wicket.apache.org/">http://wicket.apache.org/</a>  | Web Applications                          |
| ApacheDS<br><a href="http://directory.apache.org/">http://directory.apache.org/</a>   | Directory, LDAP                           |
| AspectJ<br><a href="http://www.eclipse.org/aspectj/">http://www.eclipse.org/aspectj/</a>  | Aspect Oriented Programming               |
| Bugzilla<br><a href="http://www.bugzilla.org/">http://www.bugzilla.org/</a>   | Bug-Tracking                              |
| Canoo Web Test<br><a href="http://webtest.canoo.com">http://webtest.canoo.com</a>   | Web Application Test Tool                 |
| Checkstyle<br><a href="http://checkstyle.sourceforge.net/">http://checkstyle.sourceforge.net/</a>   | Einhaltung von Coding Standards           |
| CodeCover<br><a href="http://codecover.org/">http://codecover.org/</a>  | Testtool                                  |
| Direct Web Remoting<br><a href="http://directwebremoting.org/">http://directwebremoting.org/</a>  | AJAX                                      |
| display-taglib<br><a href="http://displaytag.sourceforge.net/">http://displaytag.sourceforge.net/</a>   | Custom Tags für Tabellendarstellung       |
| dojo<br><a href="http://dojotoolkit.org/">http://dojotoolkit.org/</a>   | Rich Web Application                      |
| Eclipse BIRT<br><a href="http://www.eclipse.org/birt/">http://www.eclipse.org/birt/</a>   | Business Intelligence and Reporting Tools |
| Eclipse Equinox<br><a href="http://www.eclipse.org/equinox/">http://www.eclipse.org/equinox/</a>  | OSGI Framework                            |
| Eclipse Rich Client Platform (RCP)<br><a href="http://www.eclipse.org/home/categories/rcp.php">http://www.eclipse.org/home/categories/rcp.php</a> | Basis für Rich Client Entwicklung         |
| GlassFish<br><a href="https://glassfish.dev.java.net/">https://glassfish.dev.java.net/</a>  | Application Server                        |
| Google Web Toolkit<br><a href="http://code.google.com/webtoolkit/">http://code.google.com/webtoolkit/</a>   | AJAX                                      |
| Grails<br><a href="http://www.grails.org/">http://www.grails.org/</a>   | Groovy basierte Application Platform      |
| Groovy  | JVM Skript Sprache                        |

|  |  |
|--|--|
| <a href="http://groovy.codehaus.org/">http://groovy.codehaus.org/</a>  |  |
| Hibernate<br><a href="http://www.hibernate.org/">http://www.hibernate.org/</a>   | Persistenz                                 |
| Hudson<br><a href="https://hudson.dev.java.net/">https://hudson.dev.java.net/</a>  | Continuous Integration                     |
| Java Composite Application Platform Suite (Java CAPS)<br><a href="http://www.sun.com/software/javaenterprisesystem/javacaps">http://www.sun.com/software/javaenterprisesystem/javacaps</a> | SOA  |
| Java Server Faces<br><a href="http://java.sun.com/javaee/javaserverfaces/">http://java.sun.com/javaee/javaserverfaces/</a>   | User Interfaces                            |
| JBoss jBPM<br><a href="http://www.jboss.org/jbossjbpn/">http://www.jboss.org/jbossjbpn/</a>  | Business Process Management                |
| JBoss RichFaces<br><a href="http://www.jboss.org/jbossrichfaces/">http://www.jboss.org/jbossrichfaces/</a>   | Java Server Faces                          |
| JBoss Seam<br><a href="http://www.jboss.org/jbossseam/">http://www.jboss.org/jbossseam/</a>  | Application Platform                       |
| JDepend<br><a href="http://clarkware.com/software/JDepend.html">http://clarkware.com/software/JDepend.html</a>   | Metriken für Qualität des Software-Designs |
| JET<br><a href="http://www.eclipse.org/modeling/m2t/?project=jet">http://www.eclipse.org/modeling/m2t/?project=jet</a>   | Code Generation                            |
| JMeter<br><a href="http://jakarta.apache.org/jmeter/">http://jakarta.apache.org/jmeter/</a>  | Load Tests                                 |
| JUnit<br><a href="http://www.junit.org/">http://www.junit.org/</a>   | Unit Tests                                 |
| Liferay<br><a href="http://www.liferay.com">http://www.liferay.com</a>   | Portal Server                              |
| Maven<br><a href="http://maven.apache.org/">http://maven.apache.org/</a>   | Build Tool                                 |
| Mondrian<br><a href="http://mondrian.pentaho.org/">http://mondrian.pentaho.org/</a>  | OLAP Server                                |
| Mule<br><a href="http://mule.mulesource.org/">http://mule.mulesource.org/</a>  | ESB, SOA, Application Integration          |
| Prefuse<br><a href="http://prefuse.org/">http://prefuse.org/</a>   | Datenvisualisierung                        |
| Quartz<br><a href="http://www.opensymphony.com/quartz/">http://www.opensymphony.com/quartz/</a>  | Job Scheduling                             |
| Spring Dynamic Modules for OSGi(tm) Service Platforms<br><a href="http://www.springsource.org/osgi">http://www.springsource.org/osgi</a>   | OSGI                                       |
| Spring Framework<br><a href="http://www.springframework.org">http://www.springframework.org</a>  | Java Application Platform                  |
| Spring RCP<br><a href="http://spring-rich-c.sourceforge.net/">http://spring-rich-c.sourceforge.net/</a>  | Rich Client                                |
| Spring WebFlow<br><a href="http://www.springsource.org/webflow">http://www.springsource.org/webflow</a>  | Rich Web Application                       |
| SpringSource Application Platform<br><a href="http://www.springsource.com/products/suite/applicationplatform">http://www.springsource.com/products/suite/applicationplatform</a>           | Application Platform                       |
| Sun GlassFish Enterprise Server<br><a href="http://www.sun.com/software/products/appsrvr/">http://www.sun.com/software/products/appsrvr/</a>   | Application Server                         |
| Trac<br><a href="http://trac.edgewall.org/">http://trac.edgewall.org/</a>  | Projektmanagement                          |
| YUI<br><a href="http://developer.yahoo.com/yui/">http://developer.yahoo.com/yui/</a>   | AJAX                                       |

## 6 Software-Entwicklungsprozess

In diesem Abschnitt analysieren wir das Vorgehen in Software-Entwicklungsprojekten und wie die befragten Java-Experten den verwendeten Prozess einschätzen. Die Ergebnisse überraschen teilweise, handelt es sich doch um seit langem bekannte Grundlagen der Software-Entwicklung.

In Abbildung 16 ist dargestellt, welche **Prozesse und Vorgehensweise** typischerweise zum Einsatz kommen. Die agile Software-Entwicklung wurde dabei am häufigsten genannt. Es gibt aber offensichtlich nicht *das* typische Vorgehen im Entwicklungsprozess.

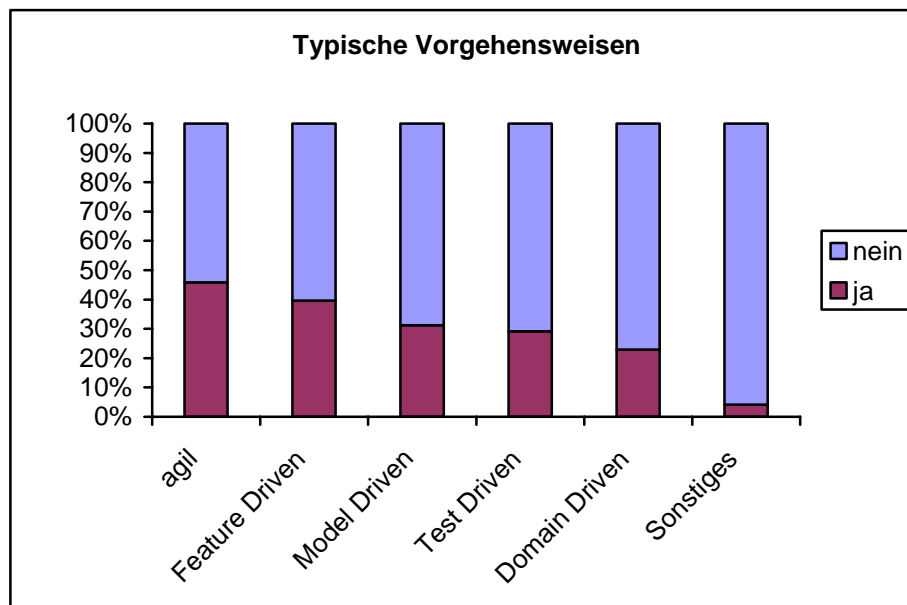


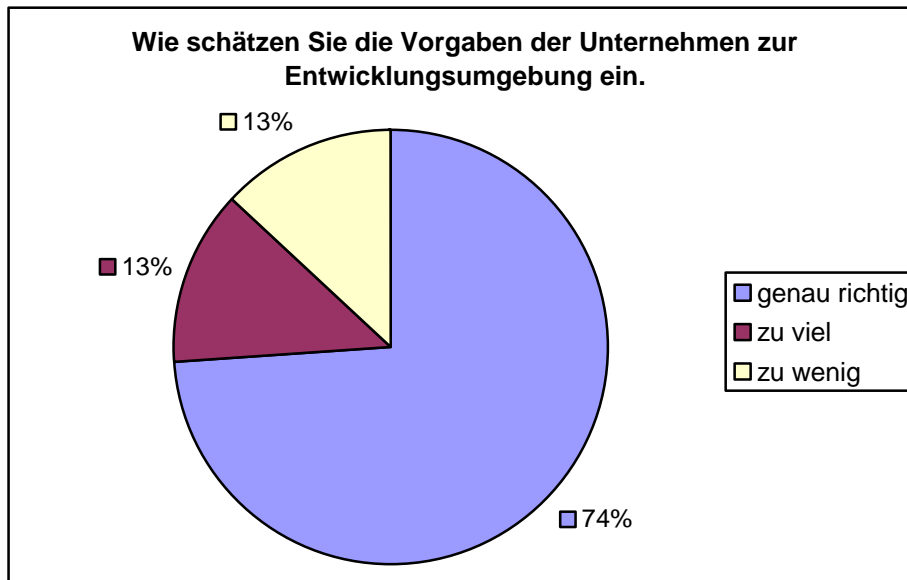
Abbildung 16: Typische Vorgehensweise im Entwicklungsprozess

Im Folgenden haben wir gezielt einige Bereiche moderner Software-Prozesse herausgenommen und analysiert. Dabei ging es nicht darum, *welche* Prozesse eingesetzt werden, sondern um die Frage, *wie zufrieden* die Java-Experten mit den Gegebenheiten in den Projekten sind.

### 6.1 Projektvorgaben zur Entwicklungsumgebung

Bei größeren Projektteams ist es von zentraler Bedeutung, ob alle Entwickler die gleiche Entwicklungsumgebung und zugehörigen Plugins verwenden. Manche Unternehmen geben in Projekten die Entwicklungsumgebung genau vor. Vielleicht wird sogar eine unternehmenseigene Distribution bzw. Update-Site errichtet. Der Vorteil liegt sicher darin, dass sich Entwickler bei der Teamarbeit auch am Arbeitsplatz des Kollegen gleich zurechtfinden. Andere Unternehmen

wiederum lassen Ihren Entwicklern mehr Spielraum, was den häufig vorhandenen persönlichen Vorlieben entsprechende Freiräume lässt.



**Abbildung 17: Vorgaben zur Entwicklungsumgebung**

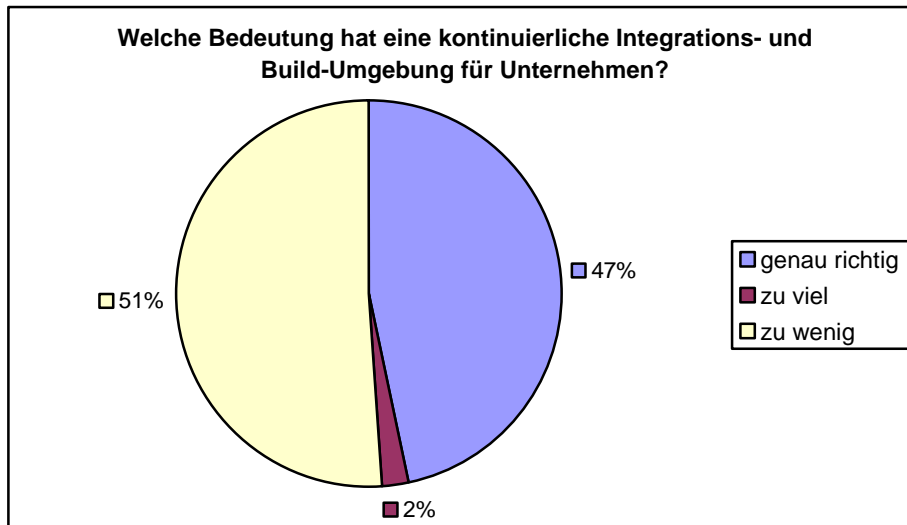
Abbildung 17 zeigt, dass drei Viertel der befragten Java-Experten mit den Vorgaben zur IDE (oder auch den gewährten Freiräumen) zufrieden sind. Das übrige Viertel teilt sich in diejenigen, die mehr bzw. weniger Vorgaben zur Entwicklungsumgebung sinnvoll fänden.

Bei drei Viertel der Projekte sind Vorgaben genau richtig. Dies ist ein tendenziell positives Ergebnis. Gleichzeitig bedeutet es, dass bei jedem Vierten Projekt die Vorgaben ungenau bzw. unvollständig sind oder zu starke Vorgaben vorhanden sind. Im Sinne eines effektiven Projektmanagements sollten Unternehmens- und IT-Leitungen ihre Projektvorgaben regelmäßig überprüfen und optimieren. Um Betriebsblindheit zu vermeiden, ist eventuell eine externe Projektberatung zu empfehlen.

## 6.2 Kontinuierliche Integration

Mit einer kontinuierlichen Build-Umgebung kann regelmäßig das Gesamtsystem kompiliert und Unit-Tests durchgeführt werden. Damit werden Integrationsprobleme so früh wie möglich entdeckt und können sofort behoben werden.

Abbildung 18 zeigt, dass 51% der befragten Java-Experten die in Projekten vorhandenen Integrationsmechanismen als nicht ausreichend empfinden. Dies sollte zu denken geben, da eine frühzeitige Fehlerentdeckung und Fehlerbehebung in der Regel günstiger ist. Dazu kommen die erzielbare höhere Qualität und Kundenzufriedenheit.

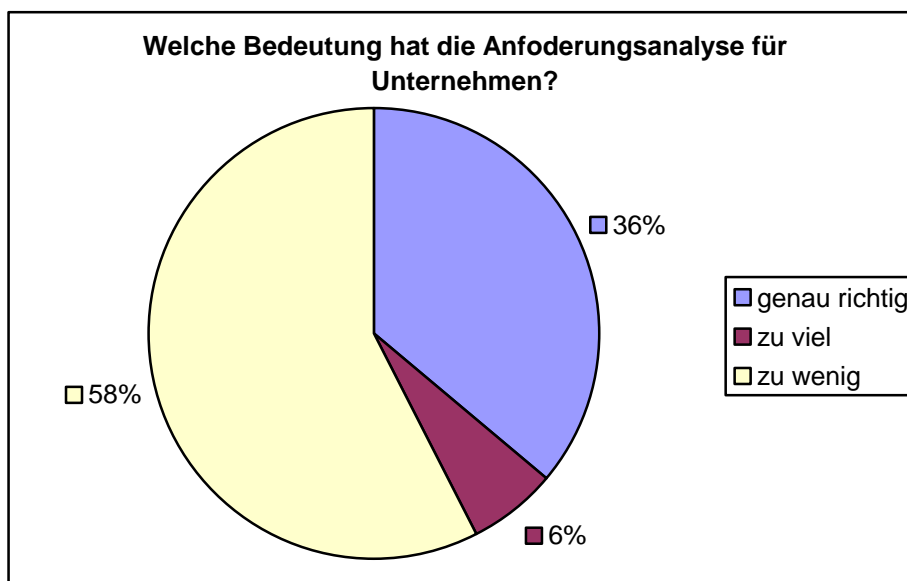


**Abbildung 18: Bedeutung der Integrations-Umgebung**

Hier besteht ein großes Rationalisierungs-Potential – sowohl finanziell als auch zeitlich.

### 6.3 Anforderungsanalyse

Die Anforderungsanalyse ist die Basis jedes Software-Entwicklungsprojekts. Nur wenn die Anforderungen des Kunden bekannt sind, kann eine optimale Lösung entwickelt werden. Andernfalls bestehen erhebliche Risiken für das Projekt, von überzogenen Terminen und Budgets bis hin zum gänzlichen Scheitern, wenn die Lösung „am Bedarf vorbei“ entwickelt wird.



**Abbildung 19: Bedeutung der Anforderungsanalyse**

Trotzdem ist der Projektalltag ein anderer. 58% der Java-Experten sind der Meinung, dass der Anforderungsanalyse zu wenig Bedeutung

zugemessen wird. Diese Zahl ist schockierend, zumal die Notwendigkeit einer optimalen Anforderungsanalyse bekannt ist.

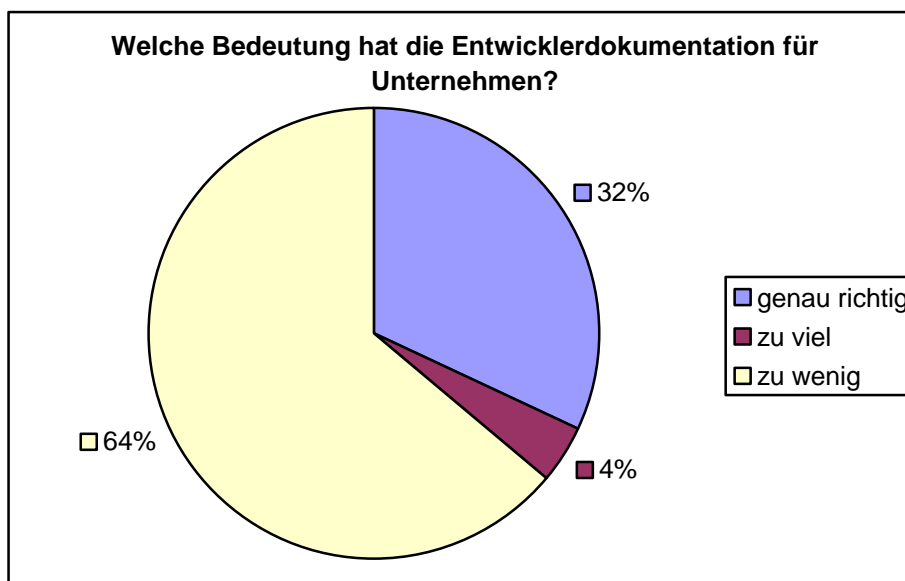
Der Software-Entwickler ist so der erste, dem bei ungenügender Spezifikation Sonderfälle oder fehlende Prozessschritte auffallen. Hier gibt es durch eine bessere Planung ein erhebliches Verbesserung- und Einsparpotential. Um den eigenen Trott im Unternehmen zu durchbrechen, kann es hilfreich sein, sich hier externe Spezialisten ins Haus zu holen.

## 6.4 Entwicklerdokumentation

Die Entwicklerdokumentation beschreibt die technologischen Aspekte einer Software. Von der groben Beschreibung des Gesamten (Adlerblick) bis hin zur quellcodenahen Dokumentation einzelner Klassen, Methoden und Algorithmen. Diese Dokumentation ist wichtig für die Wartbarkeit einer Anwendung, da sie neuen Entwicklern im Projektteam - aber nach Monaten und Jahren auch den damaligen Entwicklern selbst - einen Überblick über die Software gibt.

Eine fehlende Dokumentation führt zu erhöhten Einarbeitungsaufwand für neue Entwickler oder auch nicht-optimalen Weiterentwicklungen der Software, wenn die ursprüngliche Architektur nicht mehr vollständig verstanden wird.

Leider sind zwei Drittel der befragten Java-Experten der Meinung, dass dieser Entwicklerdokumentation zu wenig Bedeutung zugemessen wird. Kurzfristig werden so vielleicht entsprechende Projektziele erreicht (z.B. Termintreue und Budgeteinhaltung). Mittel- und langfristig ist das für die Fortentwicklung und Pflege der Anwendung jedoch nachteilig.



**Abbildung 20: Bedeutung der Entwicklerdokumentation**

Wird nur auf einen kurzfristigen Erfolg gesetzt oder werden erhebliche Mängel im Projektmanagement in Kauf genommen? Gleich, was die Gründe sind: In wirtschaftlich schwierigen Zeit kann es sich kein Unternehmen leisten, vorhersehbare Schwierigkeiten zu produzieren – zumal sie sich mit einem vergleichsweise geringen Aufwand beheben lassen.

## 6.5 Qualitätssicherung

Ähnlich sieht es auch in der Qualitätssicherung aus. 62% der Befragten sind mit der Bedeutung der Qualitätssicherung in Projekten unzufrieden. Dies ist wohl kein unerwartetes Ergebnis. Nichts desto trotz sollte es doch zu denken geben. Kosten für eine nachträgliche Fehlerbehebung und die Unzufriedenheit von Kunden sind sicher nicht einfach in Zahlen zu fassen, aber trotzdem sicher beträchtlich.

Dass man mit einer guten Qualitätssicherung auch Geld sparen kann, *weiß* eigentlich jeder gute Software-Entwickler und Projekt-Manager. Ob er auch daran *glaubt* und danach *handelt*, steht wohl auf einem anderen Blatt. Hier sind die Unternehmens- und IT-Leiter gefordert, aktiv zu werden.

Inzwischen gibt es auch qualifizierte Off-shore Anbieter, die sich auf Qualitätssicherung in der Software-Entwicklung spezialisiert haben. Von der Testfallspezifikation, über manuelle Abarbeitung von Testkatalogen bis hin zur Erstellung von Regressionstests. Neben den Kosten ist vielleicht auch die Trennung der Entwicklungs- und QS-Teams von Vorteil. So ist die Gefahr, dass QS-Mitarbeiter bei Projektverzögerungen in die Entwicklung abgezogen werden, deutlich geringer. Das Team von expeso kann hierbei behilflich sein.

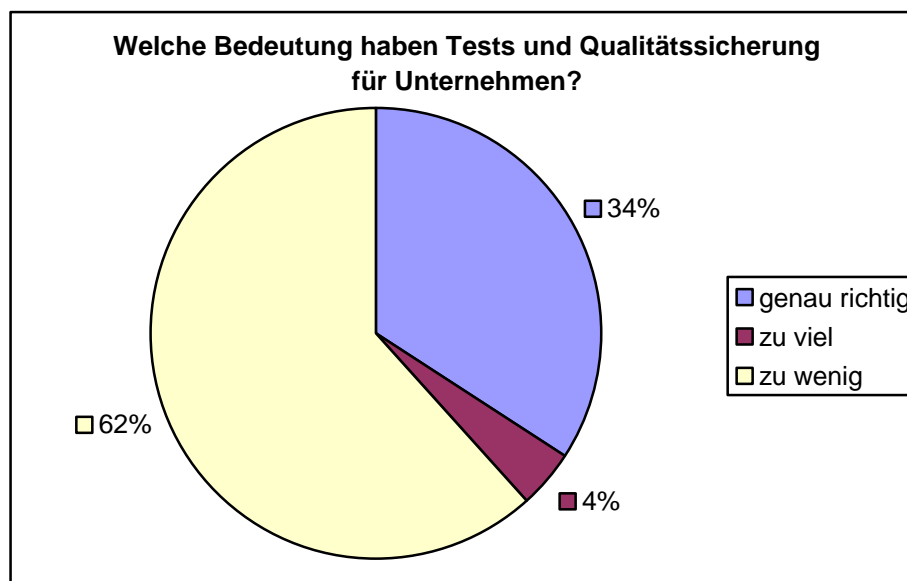


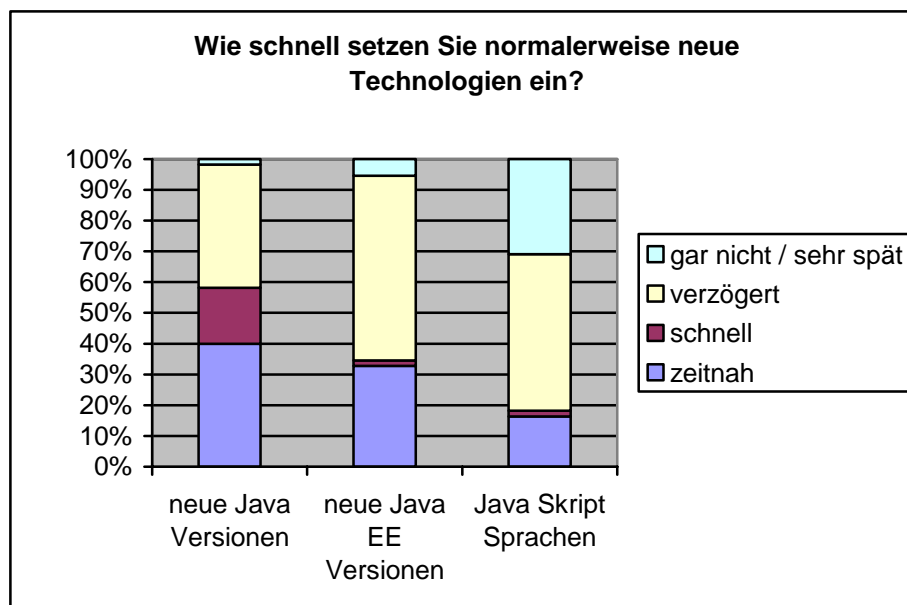
Abbildung 21: Bedeutung von Tests und Qualitätssicherung

## 7 Innovation und Weiterbildung

Die Java Technologien sind einem ständigen Innovationsprozess unterworfen. Sei es, dass der Sprachstandard selbst weiter entwickelt wird, oder aber auch neue Frameworks, Produkte und Industriestandards entstehen.

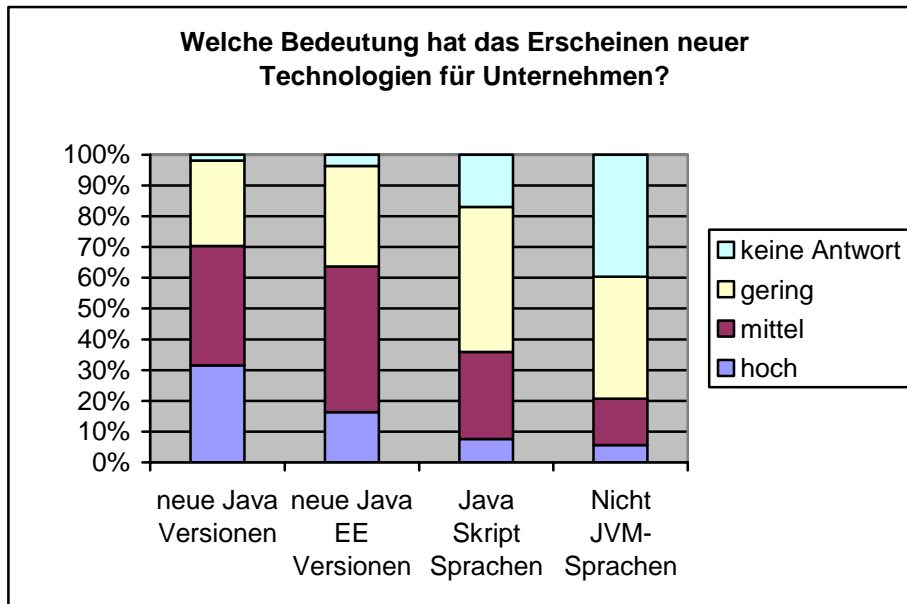
Diese ständige Innovation macht die persönliche Weiterentwicklung zur Pflicht. So müssen sich Mitarbeiter ständig weiter entwickeln, um auch morgen noch den aktuellen Arbeitsplatzanforderungen zu entsprechen. Aber auch Unternehmen und deren Produkte müssen sich an neue Technologien anpassen, um den Anforderungen ihrer Kunden gerecht zu werden. Nur so kann ein Unternehmen dauerhaft auf dem Markt bestehen. Dies gilt in besonderem Maße in wirtschaftlich schwierigen Zeiten, in denen Innovation ein wichtiger Erfolgsparameter für ein Unternehmen ist.

Abbildung 22 zeigt, dass neue Java-Versionen von ca. 60% der Befragten zeitnah bis schnell eingesetzt werden. Der Umstieg auf neue Java EE Versionen geschieht dann doch schon mit deutlicher Verzögerung und der Umstieg auf die neuen Java Skript Sprachen ist noch zurückhaltender.



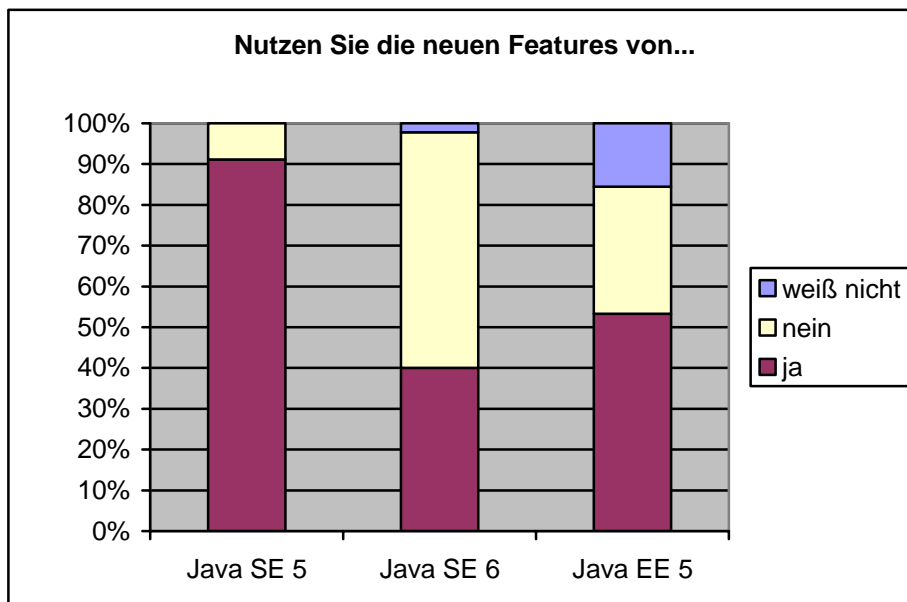
**Abbildung 22: Wechsel auf neue Technologien**

Entsprechend schätzen die Java-Experten auch die Bedeutung neuer Java-Technologien ein (Abbildung 23): eine mittlere bis hohe Bedeutung wird neuen Java und Java EE Versionen beigemessen. Mit Java Skript Sprachen und Nicht-JVM-Sprachen können deutlich weniger Experten etwas anfangen (keine Antwort) oder messen diesen Technologien geringe Bedeutung zu.



**Abbildung 23: Bedeutung neuer Technologien**

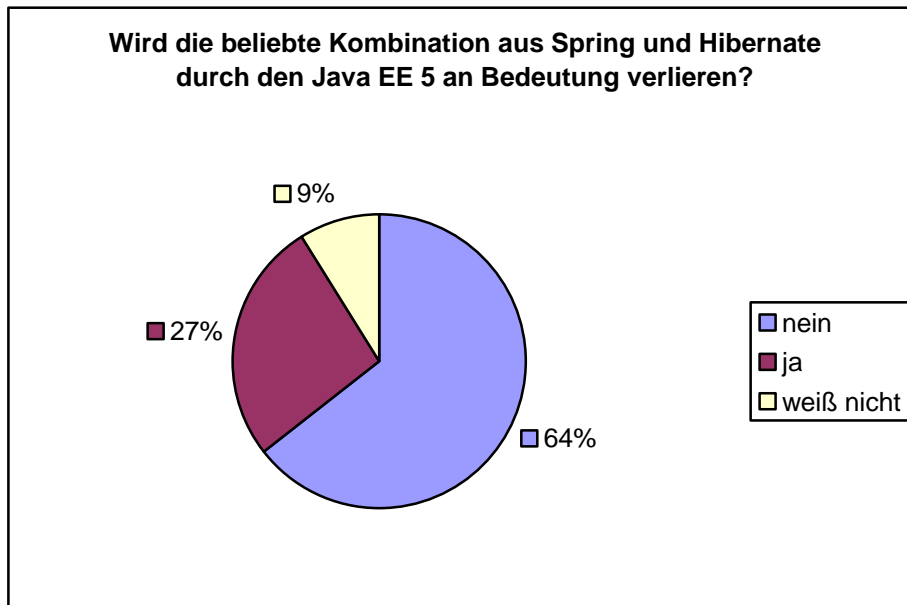
Abbildung 24 zeigt, wie häufig neue Merkmale der Java-Sprache verwendet werden. Es zeigt auch, dass die Erweiterungen der Java SE 5 deutlich mehr den Entwicklungsalltag der Befragten betreffen als die Erweiterungen der nun auch schon seit fast zwei Jahren existierenden Nachfolgeversion Java SE 6. Auch die Java EE 5 spielt nur bei etwas mehr als der Hälfte der Befragten eine Rolle.



**Abbildung 24: Nutzung neuer Sprachmerkmale**

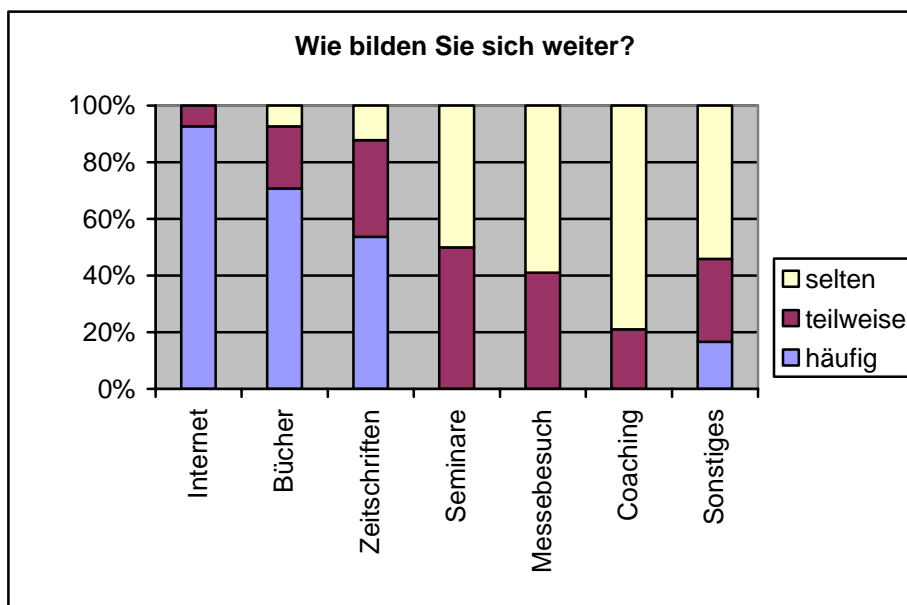
Interessant ist auch die Einschätzung der Bedeutung von Spring und Hibernate. Diese Kombination ist zwischenzeitlich häufig im Einsatz und hat auch entscheidend die Java Persistence API (JPA) und Java EE 5

geprägt. Trotzdem denken ca. zwei Drittel der Befragten (siehe Abbildung 25) nicht, dass die Bedeutung von Spring und Hibernate durch diese neuen Standards sinken wird.



**Abbildung 25: Bedeutung von Spring und Hibernate**

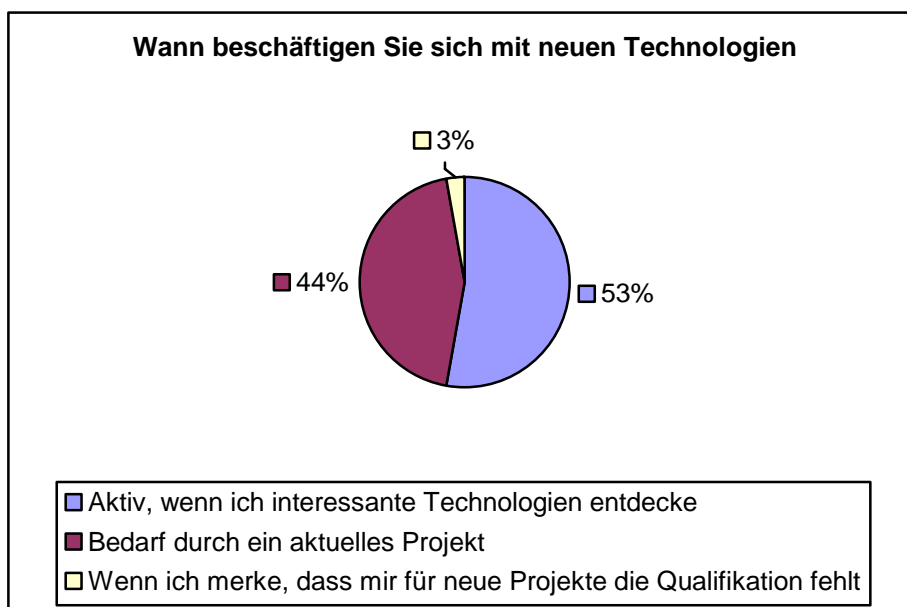
Abbildung 26 zeigt, wie sich die Befragten typischerweise weiterbilden. Einen sehr hohen Anteil hat das Selbststudium durch Internet, Bücher und Zeitschriften. Weniger als die Hälfte bilden sich aktiv durch Seminare oder Messebesuche weiter. Und nur jeder Fünfte greift auf das Know-how von Spezialisten zurück (Coaching).



**Abbildung 26: Methoden der Weiterbildung**

Diese Zahlen sind umso erstaunlicher, wenn man bedenkt, wie sehr der Unternehmenserfolg von Innovationen abhängt. So wichtig und lobenswert ein Selbststudium und eigenes Engagement sind, so beinhaltet es entscheidende Schwächen. Die volle Leistungsfähigkeit einer Technologie wird mit einem zeit- und kostenintensiven „try and error“-Vorgehen erkundet; einheitliche Standards und Kenntnisstände in einem Team fehlen. Unternehmens- und IT-Leitungen sollten daher im Sinne einer effektiven Weiterbildung verstärkt auf Technologie-Coaching setzen. Hier werden Innovationen schnell, kompetent und in voller Bandbreite gemeinsam in einem Team eingeführt und am Arbeitsplatz geschult.

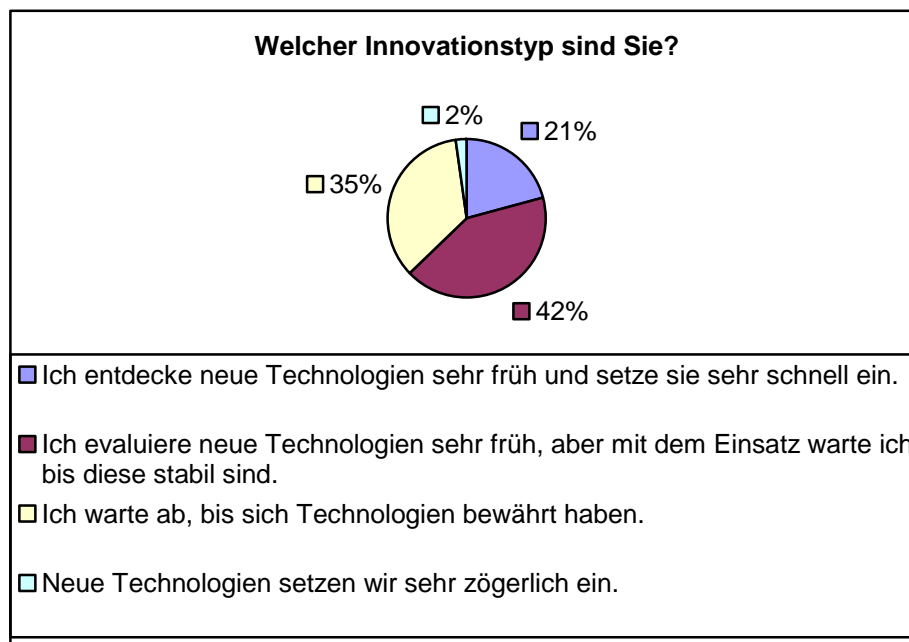
Mehr als die Hälfte (53%) der Befragten beschäftigen sich mit neuen Technologien, sobald sie diese „entdecken“. 44% lassen sich durch den Bedarf in aktuellen Projekten leiten und 3% der Befragten werden erst dann aktiv, wenn sie mit der aktuellen Qualifikation nicht mehr weiter kommen.



**Abbildung 27: Zeitpunkt der Weiterbildung**

Fast die Hälfte der Mitarbeiter beschäftigen sich somit mit neuen Technologien erst dann, wenn sie es müssen! Dies ist gerade in wirtschaftlich schwierigen Zeiten entschieden zu spät. Hierunter leidet die Innovationsfähigkeit, aber auch die Wirtschaftlichkeit. Effektivere Weiterentwicklungen und neue Technologien werden damit bei jedem zweiten Unternehmen herausgezögert, bis es nicht anders geht. Hier sind Unternehmens- und IT-Leitungen gefordert, aktiv zu werden – etwa durch entsprechende Weiterbildungsmaßnahmen.

Dieser Trend wird auch bei der Frage nach dem Innovationstyp deutlich. Hierbei geht es nicht allein um den einzelnen Mitarbeiter, sondern um den Umgang mit Innovationen in einem Unternehmen. Bei dieser Frage war uns wichtig, dass es kein Falsch und Richtig gibt, da wir von den Besonderheiten und Rahmenbedingungen für einen Technologieschwenk wissen. Abhängig von der eigenen Position kann man neue Technologien entweder sehr schnell ausprobieren, oder aber man muss vielleicht auch sehr konservativ vorgehen



**Abbildung 28: Innovationstyp**

Das Feld der Befragten teilt sich somit in etwa so auf: jeder Fünfte (21%) setzt nach eigenen Angaben neue Technologien sehr schnell ein. Zwei von Fünf (42%) der Befragten evaluieren Technologien sehr früh, setzen diese aber erst mit einem gewissen Reifegrad ein. 35% der Befragten beschäftigen sich mit Technologien erst dann, wenn sie sich bewährt haben und 2% der Befragten setzen neue Technologien sehr zögerlich ein.

## 7.1 Bewertung

Im Rahmen unserer Studie interessierten wir uns auch dafür, wie die befragten Experten neue Technologien und deren Bedeutung für Unternehmen sehen. Da dies als offene Frage formuliert war, wurde erwartungsgemäß eine Vielzahl verschiedener Aspekte genannt.

### Chancen und Risiken

In den Antworten wurden oft die Chancen und Risiken neuer Technologien betrachtet. Neue Technologien werden als **Chance** für mehr Produktivität, Arbeitserleichterung und höhere Qualität gesehen. Software kann so robuster, effizienter und effektiver entwickelt werden.

Geschäftsprozesse sind mit neuen Technologien schneller und besser an den aktuellen Markt anpassbar. Und durch die breite Unterstützung von Java EE durch die Hersteller werden immer höhere Integrationsgrade erreicht, die eine effiziente Anwendungsentwicklung ermöglichen. Damit sinkt auch die Dauer bis zum Markteintritt (time to market) und die Wettbewerbsfähigkeit steigt.

Aber man muss auch die **Risiken** betrachten. Angefangen bei unausgereiften Produkten, dem Migrations- und Testaufwand und auch Risiken durch fehlendes Know-how.

### **Konservatives Verhalten empfohlen**

Insgesamt empfehlen die Experten eher ein konservatives Verhalten für Unternehmen. Neue Technologien sollten mit Augenmaß beurteilt werden. Der Einsatz neuer Technologien als Selbstzweck ist nicht sinnvoll. Es wird empfohlen, aus der Masse der Technologien regelmäßig und sorgfältig sich „seine“ Werkzeuge auszusuchen, um sich nicht zu verzetteln und trotzdem effizient neue Lösungen erstellen zu können.

Als **Fazit** könnte das Zitat eines Teilnehmers gelten:

„Das Verstehen und Einhalten der Prinzipien, die zu wartbarer, verständlicher und robuster Software führen, sind wichtiger als technologische Mode-Erscheinungen.“

## 8 Fazit

Das Java-Trendbarometer liefert interessante Erkenntnisse. Viele ältere Versionen der Java-Sprache und des Java EE Standards sind heute noch im Projektumfeld anzutreffen. Ein Grund ist sicher der notwendige Migrationsaufwand in bestehenden Projekten.

Bei aller Vielfalt haben sich in den vergangenen Jahren doch bestimmte Technologien, Frameworks und Open Source Produkte herauskristallisiert und etabliert. Auch bei der Fehlerverfolgung im Projektalltag haben sich entsprechende Tools etabliert. Dagegen ist der Einsatz einer Integrationsumgebung nur bei 2 von 3 Projekten im Blick.

Erstaunliche Ergebnisse ergaben sich in den Bereichen Innovation und Weiterbildung, sprich der Auseinandersetzung mit neuen Technologien. Hier gibt es noch erhebliches Verbesserungspotential mit wirtschaftlichem Nutzen.

Und im Entwicklungsprozess wird es auch weiterhin Optimierungspotential geben. Von der Anforderungsanalyse über die Dokumentation bis hin zur Qualitätssicherung ist ein großer Bedarf, die Prozesse zu optimieren – gerade in wirtschaftlich schwierigen sollten Unternehmen diese Potenziale nutzen. Agile Prozesse und Testframeworks stellen interessante und hilfreiche Methoden zur Verfügung.

**Bei allen Detailergebnissen zeigt das Java-Trendbarometer auch, in welchen Bereichen eine externe Unterstützung durch ausgewählte und erfahrene Java-Experten hilfreich ist – sei es in der Software-Entwicklung, dem Technologie-Coaching oder der Technologie-Beratung. Der Aufwand hierfür dürfte sich zügig amortisieren, da in vielen IT-Projekten zum Teil erhebliche Wirtschaftlichkeitsreserven vorhanden sind. Fachlich wie betriebswirtschaftlich rechnet sich der Einsatz erfahrener Java-Experten – zum Beispiel von expeso.**

Wir hoffen, Ihnen interessante Trends und Entwicklungen gezeigt zu haben, die Sie auch für Ihre praktische Arbeit nutzen können. Wir werden dieses Trendbarometer auch zukünftig und regelmäßig erstellen. Senden Sie uns eine e-Mail an [trend@expeso.de](mailto:trend@expeso.de), und Sie erhalten auch in Zukunft die Ergebnisse zugesandt.

Haben Sie Anregungen, Tipps, Kritik oder auch Fragestellungen, die Sie gerne in zukünftigen Studien analysiert sehen möchten, senden Sie uns bitte ebenfalls eine E-Mail an [trend@expeso.de](mailto:trend@expeso.de). Wir freuen uns auf Ihr Feedback.

Herzliche Grüße



Markus Roth