



Fraunhofer Gesellschaft

Fraunhofer-Institut für  
Arbeitswirtschaft und Organisation IAO

Competence Center Electronic Business

Thomas Renner, Michael Vetter,  
Sascha Rex, Holger Kett

Eine Studie der Fraunhofer-Gesellschaft

# Open Source Software

## Einsatzpotenziale und Wirtschaftlichkeit

```

CreateObjectDemo (
main(String[] args) {
    Point origin_one = new Point(0, 0);
    Rectangle rect_one = new Rectangle(origin_one, 100, 100);
    Rectangle rect_two = new Rectangle(0, 0);

    System.out.println("rect_one width: " + rect_one.width);
    System.out.println("rect_one height: " + rect_one.height);
    System.out.println("rect_one area: " + rect_one.area());

    rect_two.origin = origin_one;

    System.out.println("rect_two origin x: " + rect_two.origin.x);
    System.out.println("rect_two origin y: " + rect_two.origin.y);

    rect_two.move(100, 0);
}

```



```

CreateObjectDemo (
main(String[] args) {
    Point origin_one = new Point(0, 0);
    Rectangle rect_one = new Rectangle(origin_one, 100, 100);
    Rectangle rect_two = new Rectangle(0, 0);

    System.out.println("rect_one width: " + rect_one.width);
    System.out.println("rect_one height: " + rect_one.height);
    System.out.println("rect_one area: " + rect_one.area());

    rect_two.origin = origin_one;

    System.out.println("rect_two origin x: " + rect_two.origin.x);
    System.out.println("rect_two origin y: " + rect_two.origin.y);

    rect_two.move(100, 0);
}

```

Thomas Renner  
Michael Vetter  
Sascha Rex  
Holger Kett

Competence Center Electronic Business

# Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit

Eine Studie der Fraunhofer-Gesellschaft

2005

# Impressum

## **Autoren**

Thomas Renner, Michael Vetter, Sascha Rex, Holger Kett  
Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart

## **Danksagung**

Die dieser Veröffentlichung zugrunde liegenden Arbeiten wurden von der Zentrale der Fraunhofer-Gesellschaft finanziert und durch IT-Leiter und IT-Verantwortliche vieler Fraunhofer-Institute unterstützt.

## **Verlag und Druck**

Fraunhofer IRB Verlag, Stuttgart

## **Vertrieb und Auslieferung**

Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO

Nobelstraße 12, 70569 Stuttgart

Telefon + 49 (0) 7 11/9 70-51 20

Telefax + 49 (0) 7 11/9 70-51 11

Andrea.Giemsch@iao.fraunhofer.de

[www.iao.fraunhofer.de/d/shop](http://www.iao.fraunhofer.de/d/shop)

und

Fraunhofer IRB Verlag

Nobelstraße 12, 70569 Stuttgart

Telefon + 49 (0) 7 11/9 70-25 00

Telefax + 49 (0) 7 11/9 70-25 08

[info@irb.fraunhofer.de](mailto:info@irb.fraunhofer.de)

[www.IRBbuch.de](http://www.IRBbuch.de)

## **Erscheinungsjahr**

2005

**ISBN** 3-8167-7008-8

© Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart

Alle Rechte vorbehalten

Dieses Werk ist einschließlich seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Fraunhofer-Instituts für Arbeitswirtschaft und Organisation IAO unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warennamen und Handelsnamen in diesem Werk berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürfen.

# Inhalt

<b>1</b>	<b>Einleitung und Zielsetzung</b>	<b>9</b>
<b>2</b>	<b>Charakteristika von Open Source Software</b>	<b>12</b>
2.1	Begriffesdefinitionen und Begriffesabgrenzungen	12
2.1.1	Begriff der Open Source Software	12
2.1.2	Public Domain Software	14
2.1.3	Freeware	14
2.1.4	Shareware	14
2.1.5	Kommerzielle Software	15
2.2	Vorteile und Nachteile von Open Source	15
2.2.1	Vorteile von Open Source Software	16
2.2.2	Nachteile von Open Source Software	17
2.3	Rechtliche Rahmenbedingungen	19
2.3.1	Open Source Software Lizenzen	19
2.3.2	Für Open Source Software relevante Rechtsbestimmungen	23
2.4	Wirtschaftlichkeit	26
2.5	Zusammenfassung und strategische Analyse	28
2.6	Gremien und Plattformen	29
<b>3</b>	<b>Open Source Software – Eine Ist-Analyse</b>	<b>31</b>
3.1	Einsatzbereiche von Software	31
3.2	Eigenschaften von Software	33
3.3	Eingesetzte Produkte und Einsatzerfahrungen	34
3.3.1	Betriebssysteme	35
3.3.2	Office Produkte	36
3.3.3	Groupware	37
3.3.4	Content Management Systeme	39
3.3.5	Dokumenten Management Systeme	40
3.3.6	Entwicklungssysteme	41
3.3.7	Datenbanken	43
3.3.8	Grafiksoftware	45
3.3.9	Sprachsoftware	46
3.3.10	ERP Systeme	47
3.3.11	Webbrowser	48
3.3.12	CRM Systeme	50
3.3.13	SRM Systeme	51
3.3.14	Wissenschaftlich-Technische Software	52
3.3.15	Cluster / Grid Software	54

3.4	Einschätzung von Open Source Software	56
3.4.1	Vorteile von Open Source Software	56
3.4.2	Nachteile von Open Source Software	57
3.4.3	Reife von Open Source Software	58
3.5	Einsatz von Open Source Software in Kundenprojekten	64
3.6	Eigenentwicklung von Open Source Software	65
3.7	Zusammenfassung	67
3.8	Ergebnisse einer Befragung bei deutschen Unternehmen	68
<b>4</b>	<b>Anwendungsfelder</b>	<b>70</b>
4.1	Einführung und Überblick	70
4.2	Betriebssysteme	71
4.2.1	Merkmale von Betriebssystemen	71
4.2.2	Gegenüberstellung von Betriebssystemen	73
4.2.3	Einsatzerfahrungen	78
4.2.4	Bewertung und Empfehlung	80
4.3	Office Pakete	82
4.3.1	Merkmale von Office Paketen	83
4.3.2	Gegenüberstellung von Office Paketen	84
4.3.3	Einsatzerfahrungen	90
4.3.4	Bewertung und Empfehlung	91
4.3.5	Datenaustausch im Office Bereich – Anforderungen und Lösungen	94
4.4	Wissenschaftlich-technische Software	99
4.4.1	Mathematische Software	100
4.4.2	Analyse- und Visualisierungssoftware	101
4.4.3	Simulationen	103
4.4.4	Konstruktionssoftware	103
4.4.5	Referenzverwaltung	104
4.4.6	Einsatzerfahrungen	105
4.5	Groupware	106
4.5.1	Merkmale von Groupware	106
4.5.2	Gegenüberstellung von Groupware	108
4.5.3	Einsatzerfahrungen	111
4.5.4	Bewertung und Empfehlung	112
4.6	Datenbanken	115
4.6.1	Merkmale von Datenbanken	115
4.6.2	Gegenüberstellung von Datenbanksystemen	116
4.6.3	Einsatzerfahrungen	118
4.6.4	Bewertung und Empfehlung	119
4.7	Entwicklungssysteme	120
4.7.1	Merkmale von Entwicklungssystemen	121
4.7.2	Gegenüberstellung von Entwicklungssystemen	124
4.7.3	Einsatzerfahrungen	131
4.7.4	Bewertung und Empfehlung	132

4.8	Grafiksoftware	134
4.8.1	Merkmale von Grafiksoftware	135
4.8.2	Gegenüberstellung von Grafiksoftware	136
4.8.3	Einsatzerfahrungen	138
4.8.4	Bewertung und Empfehlung	138
4.9	Application Server / Portalsoftware	138
4.9.1	Merkmale von Application Servern	139
4.9.2	Gegenüberstellung von Application Servern	140
4.9.3	Einsatzerfahrungen	142
4.9.4	Bewertung und Empfehlung	143
4.9.5	Portalsoftware	144
4.10	Dokumenten Management Systeme (DMS)	146
4.11	Application Service Providing (ASP)	146
4.12	Content Management Systeme	147
4.13	Cluster / Grid Software	148
4.14	Customer Relationship Management	149
4.15	Videosoftware und Rendering	150
4.16	Enterprise Ressource Planning (ERP) Software	151
4.17	Sprachverarbeitung	152
4.18	Zusammenfassung	152
<b>5</b>	<b>Wirtschaftlichkeitsbetrachtung</b>	<b>154</b>
5.1	Beschreibung des Migrationsszenarios	155
5.2	Ausgangssituation und Annahmen	157
5.3	Kostentreiber	160
5.3.1	Hardwarekosten (haushaltswirksam)	161
5.3.2	Lizenzkosten (haushaltswirksam)	161
5.3.3	Migrationsplanung (nicht haushaltswirksam)	161
5.3.4	Installation (nicht haushaltswirksam)	162
5.3.5	Übernahme Altdatenbestände (nicht haushaltswirksam)	162
5.3.6	Softwareanpassung (nicht haushaltswirksam)	162
5.3.7	Schulung (teilweise haushaltswirksam)	162
5.4	Reduzierte Produktivität (nicht haushaltswirksam)	163
5.4.1	Einführungsberatung/-unterstützung (nicht haushaltswirksam)	163
5.4.2	Interne Administration/ Programmierung (nicht haushaltswirksam)	164
5.4.3	Externer Support (nicht haushaltswirksam)	164
5.4.4	Upgrades/Neuanschaffung (haushaltswirksam)	164
5.5	Wirtschaftlichkeitsbetrachtung	164
5.5.1	Vollkostenbetrachtung	166
5.5.2	Teilkostenbetrachtung von haushaltswirksamen Kostentreibern	167
5.6	Vergleich der Ergebnisse mit Studien	167

<b>6</b>	<b>Strategische Analyse</b>	<b>169</b>
6.1	Bewertung der Vor- und Nachteile freier Software	169
6.1.1	Vorteile freier Software	169
6.1.2	Nachteile freier Software	172
6.1.3	Wertung	175
6.2	Portfolio Analyse	175
6.2.1	Strategiegebiet I: Freie Software einsetzen und vorantreiben	178
6.2.2	Strategiegebiet II: Freie Software einsetzen	179
6.2.3	Strategiegebiet III: Freie Software vorantreiben	179
6.2.4	Strategiegebiet IV: Kein Handlungsbedarf	180
6.3	Wirtschaftlichkeit	180
<b>7</b>	<b>Zusammenfassung</b>	<b>182</b>

## Abbildungen

Abbildung 1: Verbreitung von Open Source Software Lizenzen	22
Abbildung 2: Wichtigkeit von Softwarearten für die Fraunhofer-Gesellschaft	32
Abbildung 3: Relevanz von Softwareeigenschaften	34
Abbildung 4: Einsatz von Betriebssystemen	36
Abbildung 5: Einsatz von Office Produkten	37
Abbildung 6: Einsatz von Groupware	38
Abbildung 7: Einsatz von Content Management Systemen	40
Abbildung 8: Einsatz von Dokumenten Management Systemen	41
Abbildung 9: Einsatz von Entwicklungssystemen	43
Abbildung 10: Einsatz von Datenbanken	44
Abbildung 11: Einsatz von Grafiksoftware	46
Abbildung 12: Einsatz von Sprachsoftware	47
Abbildung 13: Einsatz von ERP Systemen in Projekten	48
Abbildung 14: Einsatz von Webbrowsern	49
Abbildung 15: Einsatz von CRM Systemen	50
Abbildung 16: Einsatz von SRM Systemen	52
Abbildung 17: Einsatz von Wissenschaftlich-Technischer Software	53
Abbildung 18: Einsatz von Cluster und Grid Software	55
Abbildung 19: Vorteile von Open Source Software	56
Abbildung 20: Nachteile von Open Source Software	57
Abbildung 21: Reifegrad der Funktionalität	59
Abbildung 22: Reifegrad der Benutzerfreundlichkeit	60
Abbildung 23: Reifegrad der Zuverlässigkeit	61
Abbildung 24: Reifegrad der Leistungsfähigkeit	62
Abbildung 25: Reifegrad des Supports	63
Abbildung 26: Einsatz von Open Source Software in Kundenprojekten	64
Abbildung 27: Eigenentwicklung von Open Source Software	65
Abbildung 28: Gründe für die Eigenentwicklung	66
Abbildung 29: Gründe gegen die Entwicklung von Open Source Software	66
Abbildung 30: Vorteile durch Open Source Einsatz	69
Abbildung 31: Nachteile von Open Source	69
Abbildung 32: Software-Lebenszyklus und Kostentreiber.	160
Abbildung 33: Das Relevanz-/Reifegrad-Portfolio der Open Source Software	177

## Tabellen

Tabelle 1: Vor- und Nachteile von Open Source Software	19
Tabelle 2: Kriterienraster zur Bewertung von Softwareprodukten	27
Tabelle 3: Betriebssysteme nach Lizenzen	35
Tabelle 4: Groupware Produkte nach Lizenzen	38
Tabelle 5: Content Management Systeme nach Lizenzen	39
Tabelle 6: Dokumenten Management Systeme nach Lizenzen	40
Tabelle 7: Entwicklungssysteme nach Lizenzen	42
Tabelle 8: Datenbanken nach Lizenzen	44
Tabelle 9: Grafiksoftware nach Lizenzen	45
Tabelle 10: Sprachsoftware nach Lizenzen	46
Tabelle 11: ERP Systeme nach Lizenzen	48
Tabelle 12: Web Browser nach Lizenzen	50
Tabelle 13: CRM Systeme nach Lizenzen	51
Tabelle 14: SRM Systeme nach Lizenzen	51
Tabelle 15: Wissenschaftlich-Technische Software nach Lizenzen	53
Tabelle 16: Cluster/Grid Software nach Lizenzen	54
Tabelle 17: Gegenüberstellung von Betriebssystemen	80
Tabelle 18: Gegenüberstellung von Office Paketen	92
Tabelle 19: Datenaustausch zwischen MS Office und OpenOffice.org (lesend)	95
Tabelle 20: Datenaustausch zwischen MS Office und OpenOffice.org (lesend und bearbeitend)	97
Tabelle 21: Gegenüberstellung von Groupware	113
Tabelle 22: Gegenüberstellung von Datenbanksoftware	119
Tabelle 23: Gegenüberstellung von Entwicklungssystemen	133
Tabelle 24: Gegenüberstellung von Application Servern	143
Tabelle 25: Softwareprogramme für die Migrationsszenarien	156
Tabelle 26: Auflistung der Annahmen	159

# 1 Einleitung und Zielsetzung

Open Source Software ist mittlerweile in vielen Bereichen eine realistische Alternative im Vergleich mit kommerziellen Produkten geworden. Neben Unternehmen jeder Größenordnung wird Open Source Software zunehmend auch von öffentlichen Institutionen eingesetzt. Dabei ist, wie die Beispiele der Stadtverwaltungen von Schwäbisch Hall<sup>1</sup> und München<sup>2</sup> zeigen, auch eine vollständige Migration von kommerzieller Software hin zu freien Alternativen kein Tabuthema mehr.

Das Phänomen »Open Source Software« erscheint auf den ersten Blick alle Regeln der konventionellen Wirtschaft zu brechen. So mag es beispielsweise einem Beobachter zunächst völlig unverständlich erscheinen, warum die Produzenten von komplexen und hochwertigen Softwareprodukten aus den Früchten ihrer Arbeit kein Kapital schlagen, sondern diese einfach verschenken. Es wäre jedoch zu kurz gegriffen, den Entwicklern von Open Source Software ausschließlich altruistische Motive zu unterstellen. Die Gründe für die Entwicklung von Open Source Software sind vielschichtig. Oft werden als maßgebliche Gründe der Aufbau von Know-how und Reputation genannt. Teilweise ist die Entwicklung aber auch streng nutzengetrieben – es wird Software einfach deswegen entwickelt, weil sie selbst benötigt wird<sup>3</sup>.

In letzter Zeit gingen auch immer mehr Unternehmen dazu über, Open Source Software zu fördern oder gar selbst zu entwickeln. Dahinter stehen in erster Linie betriebliche Interessen. Open Source Software ist somit ein Teil der Unternehmenspolitik geworden. Für manch ein Unternehmen ist Open Source sogar die Geschäftsgrundlage – beispielsweise in Form von Dienstleistungen im Support von Open Source Software<sup>4</sup>.

Auch für Softwareanwender werden Open Source Produkte immer interessanter. Die Qualität und Quantität von Open Source Software hat in den letzten Jahren zugenommen. In der Öffentlichkeit wird die Erfolgsgeschichte der Open Source Software sehr oft mit dem Namen »Linux« verbunden, das freie Betriebssystem ist aber bei weitem nicht das einzige erfolgreiche Open Source Produkt.

<sup>1</sup> <http://www.golem.de/0211/22883.html>

<sup>2</sup> <http://www.muenchen.de/Rathaus/dir/Imux/projekt/89257/index.html>

<sup>3</sup> Vgl. Luthiger, Benno: Alles aus Spaß? Zur Motivation von Open Source Entwicklern. In: Open Source Jahrbuch 2004. Berlin: Lehmanns Media, 2004

<sup>4</sup> Eine sehr ausführliche Darstellung der mit Open Source Software zusammenhängenden Geschäftsmodelle findet sich beispielsweise bei: Leiteritz, Raphael: Open Source Geschäftsmodelle. In: Open Source Jahrbuch 2004. Berlin: Lehmanns Media, 2004

Trotz aller Erfolge der jüngeren Zeit konnte Open Source Software die kommerzielle Software bisher nicht komplett ersetzen. In vielen Bereichen sind nach wie vor die Produkte aus den großen Softwareschmieden dominant. Auch existieren noch nicht für alle Arten von Software ausgereifte Open Source Produkte.

Open Source Software ist derzeit der Untersuchungsgegenstand zahlreicher Studien und Analysen. Viele dieser Studien haben spezielle Szenarien zum Gegenstand (z.B. die Untersuchung der Machbarkeit bzw. Wirtschaftlichkeit einer Migration auf Linux im Deutschen Bundestag). Die Ergebnisse solcher Studien sind nicht als allgemeingültig zu betrachten, da sie in der Regel spezielle Anforderungen und Rahmenparameter berücksichtigen müssen. Umgekehrt gilt aber auch für allgemeine Betrachtungen, dass diese auf ein bestimmtes Szenario nicht ohne weiteres übertragbar sind.

Die Studie stellt Einsatzpotenziale von Open Source Software in den wichtigsten Teilbereichen dar und zeigt eine Methodik, um die Wirtschaftlichkeit des Einsatzes zu berechnen. Der derzeitige Stand des Einsatzes freier Software wird beispielhaft anhand einer Erhebung dargestellt, die im Herbst 2004 in der Fraunhofer-Gesellschaft durchgeführt wurde. Des Weiteren werden für wichtige Softwarearten Vergleichsbetrachtungen (Stand Juni 2005) durchgeführt, in deren Rahmen sowohl freie als auch kommerzielle Software einander gegenübergestellt wird. Die Studie hat nicht das Ziel, eine vollständige Markterhebung für jegliche Art von Software darzustellen. Vielmehr werden generelle Tendenzen herausgearbeitet, die als Leitlinie für eigene Entscheidungen dienen können.

Im Einzelnen werden folgende *Zielsetzungen* mit dieser Studie verfolgt:

**Untersuchung, welche Open Source basierten Lösungen bereits heute verbreitet sind.**

Im November 2004 wurde eine Erhebung bei IT Verantwortlichen und Projektleitern der Fraunhofer-Gesellschaft durchgeführt die insbesondere folgende Fragestellungen beleuchtete:

- Klärung der Relevanz verschiedener Softwarearten in einer Großforschungseinrichtung.
- Stellung und Verbreitungsgrad von Open Source Software in den jeweiligen Bereichen.
- Betrachtung von Vor- und Nachteilen freier Software.
- Erhebung des Partizipationsgrades am Entwicklungsprozess von freier Software.

Die Ergebnisse dieser Erhebung werden detailliert dargestellt.

## **Darstellung von Open Source Lösungen**

Für die relevanten Anwendungsgebiete werden wichtige kommerzielle und freie Softwareprodukte gegenübergestellt und dabei wird diskutiert, welche Einsatzgebiete und -szenarien sich für den Einsatz von Open Source Produkten eher eignen und welche weniger.

## **Wirtschaftlichkeitsbetrachtungen, Kosten- / Nutzen-Analysen**

Für einige ausgewählte Einsatzgebiete werden exemplarisch Wege zur Durchführung von Wirtschaftlichkeitsbetrachtungen bzw. Kosten/Nutzen-Analysen aufgezeigt. Dies geschieht ohne Anspruch auf Vollständigkeit, da die Durchführung einer kompletten Wirtschaftlichkeitsbetrachtung den Rahmen der Studie sprengen würde. Es wird aber das grundsätzliche Vorgehen für die Erstellung von Wirtschaftlichkeitsbetrachtungen für Software dargestellt. Die Ergebnisse lassen sich daher für konkrete Szenarien leicht erweitern bzw. vervollständigen.

## **Ausarbeitung von Empfehlungen**

Es werden strategische Empfehlungen ausgearbeitet, sowohl zum Einsatz von Open Source Software als auch zu weitergehenden Fragestellungen.

Aufbauend auf den Studienergebnissen wurden zusätzlich interne Handlungsempfehlungen für den zukünftigen Einsatz von Open Source Software innerhalb der Fraunhofer-Gesellschaft erarbeitet, die in dieser Fassung der Studie nicht enthalten sind.

*Die Erstellung der Studie wurde durch Mittel der Fraunhofer-Gesellschaft ermöglicht. Die Autoren danken der Fraunhofer-Zentrale und den beteiligten Mitarbeitern für die inhaltlichen Diskussionen und die Chance, diese Studie zu erstellen. Den beteiligten Mitarbeitern aus den einzelnen Fraunhofer-Instituten gilt der Dank für die aktive Mitwirkung bei der Befragung.*

## 2 Charakteristika von Open Source Software

### 2.1 Begriffesdefinitionen und Begriffesabgrenzungen

In der aktuellen Diskussion wird der Begriff der Open Source Software oft in unzutreffender Weise gebraucht, bzw. unzulässigerweise mit anderen Softwarearten vermengt. Dieser Abschnitt soll daher zunächst den Begriff der Open Source Software klären und im Hinblick auf andere Softwarearten abgrenzen.

#### 2.1.1 Begriff der Open Source Software

Die Geschichte der Open Source Software ist noch jung. 1985 brachte Richard Stallman die Idee eines freien Betriebssystems auf UNIX Basis hervor, das GNU Projekt<sup>5</sup>. Er gründete auch die Free Software Foundation (FSF)<sup>6</sup>, eine der bedeutendsten Organisationen im Open Source Umfeld<sup>7</sup>. Der Begriff der Open Source Software ist aber noch deutlich jünger – er wurde erst im Jahr 1998 in seiner heutigen Form geprägt. Maßgeblich ist heute die Definition der Open Source Initiative (OSI)<sup>8</sup>. Diese kennzeichnet Open Source Software im Wesentlichen anhand der im Folgenden dargestellten Kriterien<sup>9</sup>.

##### **Freier Zugang zum Quellcode**

Unter dieser Forderung wird die vollständige Offenlegung des gesamten Quellcodes der Applikation verstanden. Einer Distribution von ausführbaren Binärdateien soll regelmäßig der Quellcode beiliegen. Ist dies nicht der Fall, muss der Quellcode auf geeigneten anderen Wegen (z.B. Webseite) öffentlich zugänglich gemacht werden.

##### **Freie Weitergabe der Software**

Open Source Software darf an beliebige Dritte weitergegeben werden. Einschränkungen der Autoren bezüglich der Weitergabe sind unzulässig. Die Erhebung von Lizenzgebühren ist zwar nicht ausdrücklich verboten, aber deren Durchsetzung wäre aufgrund der freien Weitergabemöglichkeit unrealistisch.

<sup>5</sup> <http://www.gnu.org>

<sup>6</sup> <http://www.fsf.org>

<sup>7</sup> Open Source Software wird teilweise auch synonym als »freie Software« bezeichnet

<sup>8</sup> <http://www.opensource.org>

<sup>9</sup> Die vollständige Darstellung der von der OSI geprägten Definition des Open Source Begriffes findet sich bei:  
<http://www.opensource.org/docs/definition.php>

## Beliebige Modifikationen der Software

Entwickler freier Software müssen beliebige Modifikationen an ihren Produkten gestatten. Die modifizierte Software muss unter den gleichen Bedingungen wie das Original weiterverbreitet werden können. Hierbei kann der Autor aber verlangen, dass die modifizierte Software als separater Patch angeboten werden muss. Damit kann die Integrität der Originalsoftware sichergestellt werden.

## Keine Einschränkungen der Nutzung

Open Source Software kann für beliebige Zwecke von jedermann genutzt werden. Eine Einschränkung ist weder im Hinblick auf bestimmte Nutzergruppen noch auf bestimmte Nutzungszwecke zulässig. So darf weder die kommerzielle Nutzung noch die Nutzung durch bestimmte Personen ausgeschlossen werden. Auch darf der Einsatz der Software nicht an den (Nicht-)Einsatz anderer Produkte oder an die Unterzeichnung eines Geheimhaltungsabkommens gebunden werden<sup>10</sup>.

Softwareprodukte, die alle diese Kriterien gleichzeitig erfüllen, sind Open Source Software im Sinne der OSI Definition – das Erfüllen eines dieser Kriterien ist keineswegs hinreichend.

Treibende Kraft hinter der Schaffung des Open Source Software Gedankens war im Übrigen keineswegs, eine kostenlose Alternative zu kommerzieller Software zu schaffen. Vielmehr maßgeblich war hingegen der Gedanke, die (Weiter-)Entwickelbarkeit von qualitativ hochwertiger Software sicherzustellen und so zu verhindern, dass dieselben Komponenten von unterschiedlichen Entwicklern neu entwickelt werden, wie dies bei Closed Source Software der Fall ist.

Die OSI formuliert das grundlegende Ziel von Open Source Software so:

»The **basic idea behind open source** is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.«<sup>11</sup>

<sup>10</sup> Insbesondere dieses Kriterium macht das von Microsoft geprägte und als »Alternative« zu Open Source eingeführte Konstrukt »Shared Source« unvereinbar mit dem Open Source Gedanken. Shared Source sieht die Unterzeichnung eines Geheimhaltungsabkommens zwingend vor.

<sup>11</sup> Vgl. <http://www.opensource.org>

### 2.1.2 Public Domain Software

Dieser Begriff umfasst Software an der der Urheber sämtliche Rechte an die Allgemeinheit abgetreten hat. Es ist somit theoretisch jedermann möglich, Änderungen an der Software vorzunehmen und diese weiterzuverbreiten. Im Gegensatz zu Open Source Software ist es aber nicht zwingend notwendig, den Quelltext zu veröffentlichen (dies ist in der Praxis eher unüblich). In diesem Fall sind Änderungen an der Software nicht möglich.

Es soll nicht unerwähnt bleiben, dass in manchen Rechtssystemen (z.B. in Deutschland) der Abtritt aller Urheberrechte an die Allgemeinheit gar nicht möglich ist. Ein Übergang in das »public domain« ist für urheberrechtlich geschützte Werke in Deutschland erst nach Ablauf der Schutzfrist von derzeit 70 Jahren vorgesehen<sup>12</sup>. Ein vorzeitiger Verzicht durch den Rechteinhaber ist nicht möglich. Rechtlich wird Public Domain Software daher wie Freeware behandelt.

### 2.1.3 Freeware

Freeware ist Software, deren Urheber der Allgemeinheit ein umfassendes kostenfreies Nutzungsrecht und Weiterverbreitungsrecht einräumt. Der Autor kann aber beliebige andere Lizenzbestimmungen aufstellen. Eine Veröffentlichung des Quelltextes ist für diese Softwareart kein notwendiges Merkmal und kann in der Praxis auch nur sehr selten beobachtet werden. Im Gegensatz zu Open Source Software wird eine Veränderung der Software von den meisten Freeware Lizenzen verboten. Häufig treten auch Einschränkungen in Bezug auf eine kommerzielle Nutzung auf. Viele Autoren von Freeware wünschen keine kommerzielle Verwertung ihrer Produkte und untersagen diese bzw. bieten die Software für kommerzielle Nutzer unter einer kommerziellen Lizenz an.

Teilweise wird diese Lizenzierungsart von kommerziellen Softwareanbietern zur Verbreitung von Demoversionen oder Beigaben genutzt. Bekannte Beispiele hierfür sind der Acrobat Reader oder der MS Internet Explorer.

### 2.1.4 Shareware

Shareware ist letztendlich keine eigene Softwareart, sondern lediglich ein besonderes Vermarktungsmodell für kommerzielle Software. Hierbei kann der Anwender die voll funktionsfähige Software kostenlos weiterverbreiten und testen. Wird die Software aber über einen bestimmten Testzeitraum hinaus genutzt (oft werden 30 Tage in den Lizenzbestimmungen genannt), sind

<sup>12</sup> Vgl. Jäger, Till; Metzger, Axel: Open Source Software. Rechtliche Rahmenbedingungen der freien Software. München: Beck, 2002. S. 5f

Lizenzgebühren fällig. Alle anderen Eigenschaften dieser Softwareart entsprechen denen kommerzieller Software.

Der Vorteil für den Vermarkter ist eine einfache Abwicklung des Vertriebs (meist über das Internet). Für den Anwender ist vorteilhaft, dass die Software vor dem endgültigen Kauf erst ausgiebig getestet werden kann, was bei kommerzieller Software ansonsten kaum möglich ist.

In der Praxis ist das Vertriebsmodell aber weitgehend gescheitert, da Anwender zur nachträglichen Entrichtung von Lizenzgebühren meist nicht bereit sind. Daher sind die meisten derart vertriebenen Produkte hinsichtlich der Funktionalität stark eingeschränkt oder mit anderen Schutzmaßnahmen versehen, so dass von Shareware im engeren Sinn kaum noch gesprochen werden kann. Bei den meisten als Shareware vertriebenen Softwareprodukten handelt es sich in der Praxis eher um Demoversionen. Die Vollversion erhält der Kunde erst nach Entrichtung der Lizenzgebühr.

### **2.1.5 Kommerzielle Software**

Kommerzielle Software wird von profitorientiert arbeitenden Unternehmen als Produkt erstellt und mit Gewinnerzielungsabsicht vermarktet. Die Bedingungen der Nutzung werden durch Urheberrecht und Lizenzbestimmungen gestaltet. Die Weiterverbreitung und Veränderung der Software ist dem Nutzer in aller Regel untersagt. Ebenso werden meistens über Lizenzbestimmungen weitergehende Einschränkungen bezüglich der Art und Weise der Nutzung vorgenommen. Die Nutzung selbst ist an die Entrichtung von Lizenzgebühren gebunden, in deren Gegenzug der Anbieter ein (mehr oder weniger beschränktes) Nutzungsrecht an der Software einräumt. Weitergehende Freiheitsrechte hat der Anwender in aller Regel nicht.

In fast allen Fällen steht dem Anwender der Quelltext von kommerzieller Software nicht zur Verfügung. Dieser wird von den Anbietern meist als Geschäftsgeheimnis betrachtet.

## **2.2 Vorteile und Nachteile von Open Source**

Mit Open Source Software wird eine Reihe von nutzenbringenden Vorteilen, aber auch Nachteilen für Entwickler und Anwender verbunden. Einige der am häufigsten genannten Aspekte sollen im Folgenden kurz dargestellt werden.

## 2.2.1 Vorteile von Open Source Software

### **Anpassbarkeit**

Open Source Software kann von jedem für individuelle Zwecke erweitert oder angepasst werden. Im Vergleich zu einer Neuentwicklung können damit umfangreiche Synergieeffekte realisiert werden. Die Vorteile von Standard-Software (bewährtes Grundprodukt) und Eigenentwicklung (Anpassung an individuelle Zwecke) können so vereinigt werden.

### **Wiederverwendbarkeit von Code**

Durch die Verwendung von Komponenten eines Open Source Produkts in eigenen Produkten kann Entwicklungszeit eingespart werden. Durch Studium von Quellcode anderer Entwickler findet auch ein Know-how Transfer innerhalb der Entwicklergemeinschaft statt.

### **Höhere Produktqualität**

Open Source Software wird (zumindest von deren Befürwortern) eine tendenziell höhere Produktqualität zugeschrieben. Dies wird mit dem im Vergleich zu kommerzieller Software grundlegend anderen Entwicklungsprozess begründet, der für Open Source Software gilt. Letztere unterliegt im Unterschied zu kommerzieller Software keinerlei Marktzwängen, insbesondere müssen keine festen Veröffentlichungstermine eingehalten werden.

### **Anbieterunabhängigkeit**

Open Source Software zwingt im Gegensatz zu manchem kommerziellen Pendant die Nutzer nicht in ein Abhängigkeitsverhältnis zu bestimmten Herstellern. Ein Diktat von (Nutzungs-)Bedingungen, wie dies bei kommerzieller Software durchaus üblich ist, muss ein Anwender von Open Source Software nicht befürchten. Vielmehr genießen die Nutzer freier Software umfassende Freiheitsrechte in Bezug auf deren Einsatz.

### **Höhere Sicherheit**

Oft wird Open Source Software ein höheres Maß an Sicherheit zugesprochen als Closed Source Produkten. Da jeder Einsicht in den Quellcode nehmen kann, werden Sicherheitslücken und –probleme schneller entdeckt, als dies bei kommerzieller Software der Fall ist. Die Möglichkeit zum Einbau von Hintertüren besteht bei freier Software ebenfalls nicht, da der entsprechende Code im Quelltext schnell auffallen würde.

### **Offene Standards**

Die von Open Source Produkten verwendeten Dateiformate und Datenaustauschstandards sind über die Verfügbarkeit der Quelltexte per Definition offen gelegt.

Dies hat eine höhere Kompatibilität und Interoperabilität von Open Source Produkten mit anderer Software zur Folge, da entsprechende Schnittstellen mit geringerem Aufwand erstellt werden können.

### **Keine Lizenzkosten**

Der Wegfall von Lizenzkosten wird oft als der zentrale Vorteil von Open Source Software überhaupt betrachtet. Man muss hierbei aber einschränkend hinzufügen, dass die Lizenzkosten oft nur einen relativ geringen Anteil des Total Cost of Ownership von Software darstellen. Nichtsdestotrotz führt der Wegfall der Lizenzgebühren in vielen Fällen zu einer erhöhten Wirtschaftlichkeit von Open Source Software gegenüber kommerzieller Software.

## **2.2.2 Nachteile von Open Source Software**

### **Keine Gewährleistungsrechte**

Die Nutzer von Open Source Software können im Regelfall weder Gewährleistungs- noch Haftungsansprüche gegen die Entwickler der Software geltend machen. Die gängigen Open Source Lizenzen übernehmen auch keinerlei Garantie für die Funktionstüchtigkeit der Software, so dass der Anwender das volle Risiko beim Einsatz trägt.

Aber auch Anbieter kommerzieller Softwareprodukte versuchen zunehmend, über entsprechende Klauseln einen weitgehenden Ausschluss von Haftungs- oder Gewährleistungsansprüchen vorzunehmen. In der Praxis spielen Gewährleistungsansprüche bei Software daher ohnehin kaum eine Rolle.

### **Kein Support durch Entwickler**

Von Seiten der Entwickler freier Software wird in den seltensten Fällen auch Support für ihre Produkte angeboten. Die Anwender müssen daher im Bedarfsfall auf die Dienstleistungen Dritter zurückgreifen. Immerhin kann festgestellt werden, dass auf dem Markt zunehmend Servicedienstleistungen für Open Source Produkte angeboten werden. Teilweise betrachten die Entwickler freier Software das Angebot von passenden Dienstleistungen für ihre Software sogar als Geschäftsmodell.

### **Höherer Schulungsaufwand**

In vielen Bereichen sind kommerzielle Softwareprodukte (noch) verbreiteter als ihre Open Source Alternativen (z.B. Office-Produkte oder Betriebssysteme). Das führt dazu, dass sowohl unternehmensintern als auch auf dem Arbeitsmarkt eher Mitarbeiter mit Know-how für diese kommerziellen Produkte gefunden werden. Sollen freie Alternativen zu diesen Produkten eingesetzt werden, müssen Mitarbeiter oftmals zunächst geschult bzw. neu eingestellt werden. Teilweise macht sich dieser Umstand auch in einem vorübergehenden Verlust von Produktivität während der Einarbeitungszeit bemerkbar.

### **Ungewisse Weiterentwicklung**

Open Source Projekte können von ihren Entwicklern jederzeit aufgegeben werden, da diese regelmäßig keinen Verpflichtungen bezüglich Wartung und Pflege ihrer Software unterliegen. Dies ist im Sinne einer wünschenswerten längerfristigen Planbarkeit des Softwareeinsatzes als Nachteil zu bezeichnen. Es soll aber auch hier nicht unerwähnt bleiben, dass es einen umfassenden Schutz vor einer Produkteinstellung auch für kommerzielle Software letztendlich nicht gibt. Open Source Software bietet aufgrund des verfügbaren Quellcodes wenigstens die Möglichkeit, die Software im Notfall auch selbst warten zu können.

### **Benötigte Applikationen sind nicht verfügbar**

Die mangelnde Verfügbarkeit benötigter Applikationen ist zwar kein Nachteil von Open Source Software an sich, es ist aber dennoch ein vielfach zu beobachtender Grund für den Nicht-Einsatz von Open Source Software. Teilweise löst dieser Aspekt auch Seiteneffekte auf andere Open Source Software aus. So sind beispielsweise deutlich weniger Applikationen für Linux verfügbar, als dies für Windows der Fall ist, was wiederum viele Anwender dazu bewegt, Linux nicht einzusetzen.

### **Mangelhafte Interoperabilität mit kommerzieller Software**

Die Anbieter von kommerzieller Software haben in den seltensten Fällen Interesse an einer Interoperabilität ihrer Software mit Open Source Produkten. Aus diesem Grund legen viele Softwareanbieter ihre Dateiformate und Schnittstellen nicht offen. Dies kann dazu führen, dass in bestimmten Szenarien Open Source Software nicht eingesetzt werden kann. So ist zum Beispiel ein Datenaustausch zwischen Microsoft Office und Open Source Office Produkten noch immer schwierig, was aufgrund der hohen Verbreitung der Microsoft Office Dateiformate zu Problemen führen kann.

In Tabelle 1 sind die Vor- und Nachteile von Open Source Software noch einmal zusammenfassend gegenübergestellt.

Vorteile	Nachteile
Anpassbarkeit	Keine Gewährleistungsrechte
Wiederverwendbarkeit von Code	(Oft) kein Support durch Entwickler
Höhere Produktqualität	Höherer Schulungsaufwand
Anbieterunabhängigkeit	Ungewisse Weiterentwicklung
Höhere Sicherheit	Applikationen teilweise nicht erhältlich
Offene Standards	Teilweise mangelnde Interoperabilität mit kommerzieller Software
Keine Lizenzkosten	

Tabelle 1: Vor- und Nachteile von Open Source Software

## 2.3 Rechtliche Rahmenbedingungen

Es ist ein weit verbreiteter Irrtum, dass sich Open Source Software in einem rechtsfreien Raum bewegt. Vielmehr spielt eine ganze Reihe von gesetzlichen Vorschriften eine Rolle für Entwickler und Anwender von Open Source Software. Die wesentlichen relevanten Regelungen entstammen den Bereichen

- Urheberrecht,
- Patentrecht,
- Haftungsrecht,
- Markenrecht und
- Wettbewerbsrecht.

Dieser Abschnitt gibt einen kurzen Überblick über die wesentlichen rechtlichen Rahmenbedingungen von freier Software<sup>13</sup>.

### 2.3.1 Open Source Software Lizenzen

Ein besonders oft anzutreffendes Missverständnis ist die Annahme, dass Open Source Software frei von Lizenzbedingungen ist. Das ist nicht der Fall – im Gegenteil: Das Urheberrecht gestattet die Vervielfältigung und Verbreitung von Software ausschließlich dem Urheber. Im Falle von Open Source Software muss

<sup>13</sup> Eine sehr ausführliche Darstellung der in Zusammenhang mit Open Source Software auftretenden Rechtsfragen findet sich in: Jäger, Till; Metzger, Axel: Open Source Software. Rechtliche Rahmenbedingungen der freien Software. München: Beck, 2002.

also auf dieses Recht explizit verzichtet werden, da jede Weitergabe der Software ansonsten illegal wäre. Dies wird in der Regel über speziell auf freie Software zugeschnittene Softwarelizenzen bewerkstelligt, von denen eine Auswahl im Folgenden kurz vorgestellt werden soll.

### **Die GNU General Public License (GPL)**

Die bekannteste Open Source Software Lizenz ist zweifellos die GNU General Public License (GPL)<sup>14</sup>. Sie wurde von Richard Stallman als Grundlage für das GNU Projekt entworfen. Die GPL erfüllt alle oben genannten Kriterien für Open Source Software. So wird die Veröffentlichung des Quelltextes für alle dieser Lizenz unterliegende Softwareprodukte zugesichert.

Darüber hinaus ist die Kerneigenschaft der GPL, dass Modifikationen an einer der GPL unterliegenden Software im Falle der Weiterverbreitung wiederum der GPL unterliegen müssen. Es ist also nicht möglich, Teile von GPL Software in kommerziellen (Closed Source) Produkten zu verwenden. Das »geistige Eigentum« der Open Source Entwickler wird so vor einer Kommerzialisierung geschützt.

Teilweise führt diese Eigenschaft der Lizenz in der Praxis dann zu Schwierigkeiten, wenn Produkte unter die GPL gestellt werden sollen, die Komponenten von Drittanbietern enthalten (z.B. Bibliotheken, die nicht einer mit der GPL kompatiblen Lizenz unterliegen<sup>15</sup>). Diese Komponenten können ohne Einverständnis des Urhebers nicht ohne weiteres umlizenzieren werden. Für einen Einsatz in GPL Software wäre dies aber nötig. Die von interessierter Seite manchmal angeführte Behauptung, dass der Einsatz von GPL lizenzierter Software dazu führt, dass sämtliche ebenfalls eingesetzte Software auch der GPL unterliegen muss, ist allerdings völlig unzutreffend.

Es soll nicht unerwähnt bleiben, dass die Einschränkungen der GPL nur dann greifen, wenn abgeleitete Software auch tatsächlich weiterverbreitet wird. Für den internen Einsatz ist es sehr wohl möglich, nicht der GPL unterliegende Komponenten mit GPL Software zu verbinden oder Änderungen an GPL Software vorzunehmen, ohne den Quellcode zu veröffentlichen. Die GPL kennt keine Pflicht zur Veröffentlichung oder Weitergabe von abgeleiteten Produkten.

Im Hinblick auf die aktuelle Diskussion um Softwarepatente wurde auch eine Klausel in die GPL eingefügt, die es verbietet, patentgeschützte Software unter die GPL zu stellen. So soll eine Aushöhlung der GPL durch Softwarepatente vermieden werden.

<sup>14</sup> Der vollständige Text der GPL findet sich bei <http://www.gnu.org/copyleft/gpl.html>

<sup>15</sup> Eine Aufstellung mit der GPL kompatibler Lizenzen findet sich bei <http://www.gnu.org/philosophy/license-list.html>

Derzeit findet auch die Überarbeitung der aktuellen Version 2.0 der GPL statt. Die kommende Version 3.0 ist aber noch nicht veröffentlicht.

### **Die Lesser GNU General Public License (LGPL)**

Eine abgeschwächte Variante der GPL, die Lesser GNU General Public License (LGPL)<sup>16</sup> kennt die Einschränkung der GPL nicht, dass abgeleitete Produkte wiederum derselben Lizenz unterliegen müssen. Eine Verwendung von LGPL lizenzierten Komponenten ist somit auch in kommerziellen Produkten möglich. Diese Lizenz wird daher insbesondere für Softwarebibliotheken genutzt, die gleichermaßen Open Source Entwicklern und kommerziellen Softwareanbietern zur Verfügung stehen sollen. Dennoch gilt, dass eine Veränderung des Basisprodukts (etwa der Bibliothek selbst) wiederum der LGPL unterliegen muss, sofern sie weiterverbreitet wird.

### **Die Berkeley Software Distribution License (BSD Lizenz)**

Die BSD Lizenz<sup>17</sup> wurde von der Universität Berkeley entwickelt. Diese Lizenz sieht eine Veröffentlichung des Quellcodes nicht zwingend vor, so dass es auch möglich ist, nur Binärdateien zu veröffentlichen. Auch müssen abgeleitete Produkte nicht zwingend wieder derselben Lizenz unterliegen, was eine Verwendung von BSD Software in kommerziellen Produkten möglich macht. Nicht einmal eine Änderung des Basisprodukts bedingt zwingend die Weiterverwendung der BSD Lizenz. Somit können Abwandlungen von BSD lizenzierter Software durchaus in proprietäre Software überführt werden. Insgesamt kennt die BSD Lizenz kaum eine Einschränkung der Nutzung von dieser Lizenz unterliegender Software.

Eine ältere Version der BSD Lizenz enthielt eine Klausel bezüglich der Pflichtanbringung eines Werbeverweises in abgeleiteten Produkten. Diese alte BSD Lizenz wird von den Erstellern der GPL als inkompatibel mit der GPL betrachtet, so dass derartig lizenzierte Komponenten nicht in GPL Software verwendet werden können.

<sup>16</sup> Der vollständige Text der LGPL findet sich bei <http://www.gnu.org/copyleft/lesser.html>

<sup>17</sup> Ein Template für eine BSD Lizenz kann bei der OSI eingesehen werden: <http://www.opensource.org/licenses/bsd-license.php>

### Zusammenfassung

Aus Abbildung 1 ist ersichtlich, dass die drei vorgestellten Lizenzen den überwiegenden Teil der existierenden Open Source Software abdecken. Alleine die der GPL unterliegende Software stellt fast drei Viertel aller Open Source Software Produkte. Es gibt jedoch eine fast unüberschaubare Anzahl an weiteren Lizenzen, da sehr oft für Produkte eigens neue Lizenzen entwickelt werden, anstatt bereits vorhandene zu nutzen. Die Open Source Initiative (OSI) stellt auf ihrer Webseite eine Liste der mit der Open Source Definition vereinbaren Lizenzen bereit<sup>19</sup>.

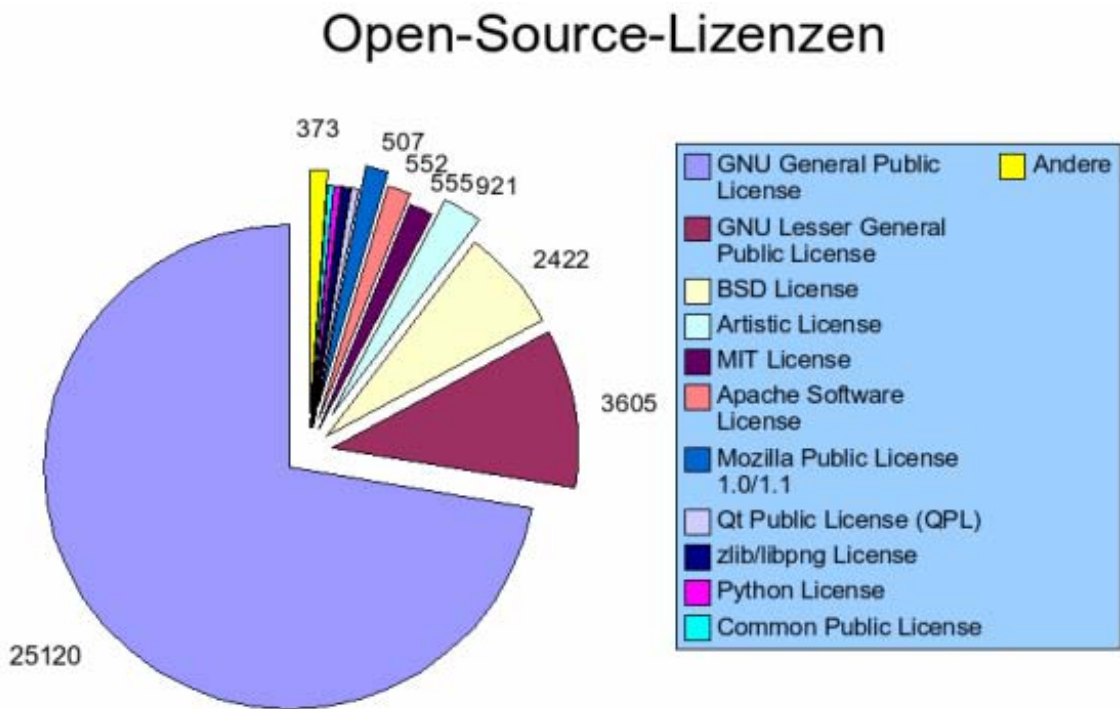


Abbildung 1: Verbreitung von Open Source Software Lizenzen (Quelle: BerliOS<sup>18</sup>)

<sup>18</sup> <http://openfacts.berlios.de/index.phtml?title=Open-Source-Lizenzen>

<sup>19</sup> <http://www.opensource.org/licenses/index.php>

## 2.3.2 Für Open Source Software relevante Rechtsbestimmungen

### Urheberrecht

Das Urheberrecht gesteht dem Autor einer Software das ausschließliche Recht zur Weiterverbreitung, Veränderung und zur Anfertigung von Kopien zu. Es macht dabei keinen Unterschied ob die Software im Quellcode oder nur als ausführbare Binärdatei vorliegt. Die urheberrechtlichen Bestimmungen finden somit auch für Open Source Produkte Anwendung. Der Schutz durch das Urheberrecht ist automatisch. Eine gesonderte Beantragung der Schutzrechte durch den Autor ist nicht notwendig.

### Gewährleistungs- und Haftungsrecht

Alle gängigen Open Source Lizenzen enthalten einen vollständigen Ausschluss von Gewährleistungs- und Haftungsansprüchen. Der Anwender nutzt Open Source Software also grundsätzlich auf eigene Gefahr.

Interessant ist die Frage, inwiefern dieser Haftungsausschluss zulässig bzw. mit deutschem Recht vereinbar ist (denn die meisten Open Source Lizenzen wurden vor dem Hintergrund des amerikanischen Rechtssystems ausgestaltet). Nach deutschem Recht sind die Open Source Lizenzen als Allgemeine Geschäftsbedingungen aufzufassen<sup>20</sup>. Für die Geltung von AGB ist es wichtig, dass der Nutzer von Open Source Software von den AGB in zumutbarer Weise Kenntnis erlangen konnte – dies kann wohl mit der Beifügung der Lizenz in das Softwarepaket als erfüllt gelten. Insbesondere, wenn Änderungen an der Software vorgenommen werden oder diese weitergegeben wird, kann unterstellt werden, dass der Nutzer die Lizenz zur Kenntnis genommen und auch akzeptiert hat. Denn ansonsten wären derartige Handlungen aufgrund von urheberrechtlichen Einschränkungen unzulässig. Man kann also davon ausgehen, dass Open Source Lizenzen durchaus wirksame AGB sind.

Der vollständige Gewährleistungsausschluss ist allerdings mit dem AGB Recht nicht vereinbar und daher unwirksam. Trotzdem sind in der Praxis keine Gewährleistungs- oder Haftungsansprüche gegen die Urheber von Open Source Software möglich. Dies ist dadurch begründet, dass die (kostenfreie!) Nutzung von Open Source Software unter das Schenkungsrecht fällt, das wegen der Uneigennützigkeit der Schenkung die Gewährleistungsansprüche auf Arglist beschränkt. Des Weiteren sind Haftungsansprüche auf Vorsatz und grobe Fahrlässigkeit beschränkt. Diese sind in der Praxis äußerst selten nachweisbar.

<sup>20</sup> Vgl. Jaeger, Till; Metzger, Axel: Open Source Software – Rechtliche Rahmenbedingungen der freien Software. München: Beck Verlag, 2002. S. 146f

## Markenrecht

Grundsätzlich ist der Grundgedanke des Schutzes von Marken und Handelszeichen sehr wohl mit dem Open Source Gedanken vereinbar. Die Rechteinhaber von Open Source Produkten können daher Namen oder Logos ihrer Produkte markenrechtlich absichern. Dies hat keinen Einfluss auf die Qualifikation einer Software als freie Software.

Problematisch ist, dass der Namensgebungsprozess bei vielen Open Source Projekten keine markenrechtliche Überprüfung beinhaltet und es daher leicht zu Konflikten mit bestehenden Marken kommen kann. Teilweise lehnen sich Open Source Projekte absichtlich an Namen von funktionsähnlichen kommerziellen Produkten an. Dies bietet den Rechteinhabern teilweise willkommene Angriffsflächen und erhöht somit die Gefahr von Markenstreitigkeiten sogar noch.

## Patentrecht

Weiterhin spielt zunehmend das Patentrecht eine wichtige Rolle im Umfeld von Open Source Software. Obwohl Software derzeit in Europa eigentlich nicht patentierbar ist, werden in der Praxis von den Patentämtern Schutzrechte zunehmend auch auf Softwareverfahren gewährt.

In der Europäischen Union wird derzeit über eine neue Richtlinie diskutiert, die zukünftig die Frage nach der Patentierbarkeit von Software klären soll. Die EU-Kommission strebt dabei eine Regelung an, die Softwarepatente auf breiter Basis zulassen würde. Begründet wird die Patentierbarkeit von Software insbesondere mit dem Schutz des Erfinders vor wirtschaftlichen Nachteilen durch Nachahmer. Investitionen in Forschung sollen zukünftig besser als bisher geschützt und damit gefördert werden. Auch soll damit in diesem Bereich ein Mehr an Rechtssicherheit entstehen, da in den vergangenen Jahren in Europa bereits ca. 30 000 Softwarepatente<sup>21</sup> gewährt wurden, dafür aber keine eindeutige Rechtsgrundlage existiert.

Der Richtlinienentwurf ist heftig umstritten - Gegner von Softwarepatenten halten dem Entwurf der EU Kommission entgegen, dass es keineswegs gewährleistet ist, dass die Zulassung von Softwarepatenten zu mehr Innovation führen würde. Die Mehrzahl der Softwarepatente wird von Großunternehmen gehalten. Kleinere und mittelständische Anbieter können sich die oft sehr teuren und langwierigen Patentanmeldungen kaum leisten. Es besteht folglich die Gefahr, dass Großunternehmen die kleineren Konkurrenten mit Hilfe ihres Patentportfolios aus dem Markt zu drängen versuchen, insbesondere, wenn

<sup>21</sup> Vgl. <http://swpat.ffii.org/log/intro/index.en.html>

auch Patente auf einfachste und grundlegendste Ideen und Verfahren gewährt werden (sog. Trivialpatente). Die Gefahr von volkswirtschaftlich schädlichen Monopolbildungen steigt somit, die gesamtwirtschaftlichen Investitionen in Forschung und Entwicklung könnten unter dem Strich sogar sinken und Arbeitsplätze verloren gehen.

Neben kleinen und mittelständischen Unternehmen sind auch auf dem Open Source Gedanken basierende Entwicklungsmodelle durch Softwarepatente betroffen. Im Unterschied zum Urheberrecht ist es im Falle des Patentrechts durchaus möglich, völlig unwissend eine Rechtsverletzung zu begehen. Das Urheberrecht schützt eine konkrete Implementation eines Verfahrens, das Patentrecht schützt das Verfahren an sich. Wird ein patentgeschütztes Verfahren in einem Open Source Projekt verwendet, so besteht das Risiko vom Patentinhaber verklagt zu werden – auch dann, wenn der Entwickler von dem Patent keine Kenntnis hatte. Da auch Patente auf äußerst grundlegende Ideen und Algorithmen gewährt werden (z.B. Fortschrittsbalken, Karteireiter usw.), ist das Risiko, bei der Softwareentwicklung Patente Dritter zu verletzen, entsprechend hoch bzw. sogar unvermeidbar.

Da in der Zwischenzeit auch schon Patente auf Datenformate gewährt wurden, ist zukünftig auch die Interoperabilität von Open Source Produkten mit ihrer kommerziellen Konkurrenz gefährdet, da die Implementation entsprechender Schnittstellen durch den Patentinhaber untersagt werden kann. Dieses Vorgehen ist in der Praxis auch bereits zu beobachten.

Die Entwickler von Open Source Software werden sich nur in den seltensten Fällen teure Patentanmeldungen leisten können. Auch die ebenso kostspieligen Patentprüfungen werden seltener als nötig durchgeführt werden können. Das Risiko von Rechtsstreitigkeiten steigt damit für Open Source Projekte an. In welchem Umfang das Entwicklungsmodell der Open Source Software als solches durch Softwarepatente bedroht ist, hängt letztendlich von der zukünftigen Gesetzeslage bzw. Rechtsprechung ab. In der derzeit gängigen Rechtsprechung wird bei Patentverletzungen durch einzelne Open Source Entwickler eher davon ausgegangen, dass die Unterlassung einer gründlichen und kostenintensiven Patentprüfung nicht gegen die Sorgfaltspflicht verstößt. Insofern ist immerhin festzustellen, dass diese Open Source Entwickler Schadenersatzansprüche durch Patentinhaber wohl eher weniger zu befürchten haben<sup>22</sup>. Auf Unternehmen, die Open Source Software entwickeln oder im Rahmen von Distributionen anbieten, trifft dies jedoch nicht zu. Ebenso sind Unterlassungsansprüche gegen Open Source Produkte von der Frage des Schadenersatzes

<sup>22</sup> Vgl. Jaeger, Till; Metzger, Axel: Open Source Software – Rechtliche Rahmenbedingungen der freien Software. München: Beck Verlag, 2002. S. 128

unberührt. In Letzteren besteht die eigentliche Gefahr von Softwarepatenten für Open Source Software, da dadurch der Fortbestand eines Open Source Projektes als solches gefährdet sein könnte.

Auch die reinen Anwender von Open Source Software können durchaus von Softwarepatenten betroffen sein – sollte etwa ein Open Source Software Projekt durch die Durchsetzung eines Unterlassungsanspruchs eingestellt werden müssen.

## 2.4 Wirtschaftlichkeit

Die Frage, ob Open Source Software oder proprietäre Software wirtschaftlicher ist, lässt sich nicht allgemeingültig beantworten, sondern ist vom jeweiligen Einzelfall abhängig. Die Durchführung einer Wirtschaftlichkeitsbetrachtung ist daher immer erforderlich.

Für Open Source Software spricht der weitgehende Wegfall von Lizenzgebühren - für proprietäre Software spricht der vergleichsweise größere Umfang an verfügbarem Know-how. Hardwarekosten verhalten sich in einer vergleichenden Betrachtung der beiden Softwarearten weitgehend neutral.

Neuere Studien gehen von einer Einsparmöglichkeit von bis zu 30% durch den Einsatz von Open Source Software aus. Dabei können größere Unternehmen höhere Einsparpotenziale realisieren, als kleinere Unternehmen (da deren Lizenzkostenblock meist höher ist). Finanziell besonders attraktiv sind Open Source Produkte in den Bereichen Büroanwendungen, Server-Betriebssysteme, Datenbanken und Content Management Systeme<sup>23</sup>.

Die Beantwortung der Frage nach dem möglichen Einsatz von Open Source Software kann durch die Durchführung einer Wirtschaftlichkeitsbetrachtung erfolgen, beispielsweise nach dem von Fraunhofer IAO entwickelten 6-Phasen Modell »Wirtschaftlichkeitsbetrachtung und Strategieoptimierung«. Das Modell sieht folgende Phasen vor:

- 1 Ist-Analyse
- 2 Strategiebewertung
- 3 Identifikation zusätzlicher Anwendungen
- 4 Identifikation von Handlungsalternativen
- 5 Bewertung und Abgabe einer Handlungsempfehlung
- 6 Konzeption und Umsetzung

<sup>23</sup> Vgl. Binder, Steffen; Henke, Melanie: Kassensturz. Open Source und proprietäre Software in Vergleich. Frankfurt: Soreon Research, 2003.

Im Rahmen der Wirtschaftlichkeitsbetrachtung werden in Frage kommende Softwareprodukte gegenübergestellt und bewertet, wobei sowohl kommerzielle als auch freie Software nach einem einheitlichen Kriterienraster bewertet wird. Die Lizenzkosten sind hierbei eines dieser Kriterien, aber bei weitem nicht das einzige. Vielmehr müssen auch andere, teilweise nicht monetär bewertbare Kriterien in die Betrachtung einbezogen werden, um eine fundierte Entscheidung für oder gegen den Einsatz von Open Source Software treffen zu können. Letzteres trifft insbesondere auf die umfangreichen mit Open Source Software verbundenen Freiheitsrechte zu. Diese sind quantitativ schwer zu erfassen, können aber für Unternehmen in vielen Fällen einen strategischen Mehrwert darstellen. Dieser muss über eine Nutzwertanalyse in die Betrachtung einfließen. Die einzelnen Kriterien können den drei Bereichen

- Einführungskosten,
- Betriebskosten und
- Strategische Kriterien

zugeordnet werden.

Ein Kriterienraster zur Auswahl von Softwareprodukten findet sich in Tabelle 2.

<b>Einführungskosten</b>	<b>Betriebskosten</b>	<b>Strategische Kriterien</b>
Personalkosten	Personalkosten	Stabilität
Beratungskosten	Wartung	Sicherheit
Lizenzkosten	Hardwarekosten	Herstellerunabhängigkeit
Schulungskosten	Lfd. Schulungskosten	Benutzerfreundlichkeit
Migrationskosten	Updatekosten	Interoperabilität
Installationskosten		Anpassbarkeit an individuelle Zwecke
Einarbeitungsaufwand		Vorhandenes Know-how

Tabelle 2: Kriterienraster zur Bewertung von Softwareprodukten<sup>24</sup>

<sup>24</sup> Angelehnt an: Leipelt, Detlef: Grundgedanken zu Wirtschaftlichkeitsbetrachtungen für den Einsatz von Open Source Software. [<http://www.kbst.bund.de/Anlage305827/Grundgedanken-zu-Wirtschaftlichkeitsbetrachtungen-fuer-den-Einsatz-von-OSS-pdf-13-8-kB.pdf>]

## 2.5 Zusammenfassung und strategische Analyse

Bei der Auswahl von Softwareprodukten sollte Open Source Software auf jeden Fall in die Betrachtung einbezogen werden. In vielen Anwendungsszenarien kann diese für Anwender finanzielle und/oder strategische Vorteile mit sich bringen. Dies muss jedoch immer im Einzelfall über eine Wirtschaftlichkeitsbetrachtung bzw. Nutzwertanalyse ermittelt werden.

Open Source Software wird ihren Marktanteil wohl weiter ausbauen können. Insbesondere bei Serverprodukten wie Datenbanken oder Server-Betriebssystemen sind deutliche Wachstumsraten wahrscheinlich. Für Desktop Produkte wird dieser Prozess vermutlich noch etwas mehr Zeit in Anspruch nehmen, da die Stellung der dort verbreiteten kommerziellen Software noch sehr stark ist. Gerade im Office Bereich erzeugen die etablierten Produkte einen regelrechten »Gruppenzwang«, so dass sich freie Software dort noch nicht in größerem Umfang behaupten konnte.

Gefahren für freie Software entstehen derzeit aus den Bemühungen der Softwareindustrie, das Patentrecht zur Abwehr der freien Konkurrenz einzusetzen, sowie aus nicht offen gelegten Datenformaten, Protokollen und Hardware-spezifikationen. Letztere werden oft auch noch zusätzlich mit dem Patent- und Urheberrecht abgesichert, um sie vor dem Zugriff der Open Source Entwickler zu schützen. Für diese macht sich hierbei die fehlende Lobby in Hinblick auf derzeitige gesetzgeberische Entwicklungen, insbesondere auf dem Gebiet des Patentrechts, negativ bemerkbar.

Positiv für die Stellung der Open Source Software ist die zunehmende Akzeptanz freier Software in öffentlichen Einrichtungen. Diese wird sich aller Voraussicht nach beschleunigend auf die Verbreitung von Open Source Software auch in anderen Bereichen auswirken. Ebenfalls positiv zu vermerken ist das steigende Engagement gerade auch großer Technologieanbieter für Open Source Software. Diese beziehen den Open Source Gedanken zunehmend in ihre Geschäftsmodelle mit ein und nehmen oft aktiv am Entwicklungsprozess teil. Damit tragen sie auch dazu bei, Open Source Software von dem ihr hin und wieder unrichtigerweise immer noch zugeschriebenen »Hackerimage« zu befreien.

## 2.6 Gremien und Plattformen

Es existieren inzwischen zahlreiche Initiativen, Gremien und Plattformen im Open Source Umfeld. Eine Auswahl soll im Folgenden kurz vorgestellt werden.

### Open Source Initiative (OSI)

Die Open Source Initiative (OSI)<sup>25</sup> ist eine non-profit Organisation, deren Ziel hauptsächlich in der Verbreitung des Open Source Gedankens liegt. Die OSI pflegt vor allem die Definition des Begriffes »Open Source« sowie ein Verzeichnis von Open Source Lizenzen. Die OSI vergibt auch ein Gütesiegel für Open Source Lizenzen, mit dem Open Source Software als solche gekennzeichnet werden kann.

### Free Software Foundation (FSF)

Die Free Software Foundation<sup>26</sup> wurde 1985 von Richard Stallman gegründet, primär um das ein Jahr früher gestartete GNU Projekt zu beherbergen. Das GNU Projekt hat zum Ziel, ein UNIX kompatibles, freies Betriebssystem zu schaffen. GNU besteht heute aus zahllosen einzelnen Komponenten, von denen die wohl bekannteste der Betriebssystemkern Linux ist. Weitere wesentliche Teile des GNU Projekts sind der GNU C-Compiler sowie der EMACS Editor.

Die FSF selbst hat zum einen die Verbreitung des Open Source Gedankens (insbesondere des GNU Systems) zum Ziel. Zum anderen engagiert sich die FSF aber auch für allgemeine Fragen, die Open Source Software betreffen können (z.B. das Recht auf freie Meinungsäußerung und das Verbot von Softwarepatenten).

Die FSF unterhält auch Schwesterverbände in Europa und Indien.

### Linuxverband

Der Linuxverband<sup>27</sup> ist ein Branchenverband, in dem sich Unternehmen zusammengeschlossen haben, die freie Software anwenden, produzieren oder Dienstleistungen in deren Umfeld anbieten. Trotz des Namens ist der Linuxverband nicht nur im Umfeld von Linux tätig, sondern für freie Software aller Art. Ziel des Verbandes ist die Vertretung der Interessen dieser Unternehmen gegenüber Politik und Verwaltung, aber auch der Erfahrungsaustausch der Mitglieder untereinander.

<sup>25</sup> <http://www.opensource.org>

<sup>26</sup> <http://www.fsf.org>

<sup>27</sup> <http://www.linuxverband.de/>

### **Institut für Rechtsfragen der freien und Open Source Software (ifrOSS)**

Das Institut für Rechtsfragen der freien und Open Source Software<sup>28</sup> beschäftigt sich mit den in Zusammenhang mit freier Software auftretenden juristischen Aspekten. Dies umfasst sowohl Open Source Software Lizenzen als auch Rechtsgebiete, die für Open Source Software relevant sind, wie etwa Urheber- und Patentrecht. Neben der Veröffentlichung von Publikationen und dem Angebot von Seminaren versucht das Institut auch, sich in öffentliche Diskussionen in diesen Gebieten einzubringen.

### **BerliOS**

BerliOS<sup>29</sup> versteht sich als Mediator für die verschiedenen Interessengruppen im Open Source Umfeld. Zielgruppen sind sowohl Entwickler von Open Source Software als auch deren Anwender, aber auch Unternehmen, die sich im Open Source Umfeld bewegen (z.B. Dienstleister). Unter [www.berlios.de](http://www.berlios.de) wird auch ein Open Source Portal mit verschiedenen Themengebieten bereitgestellt. Es soll nicht unerwähnt bleiben, dass sich Fraunhofer FOKUS sehr stark für BerliOS engagiert.

### **SourceForge.net**

SourceForge.net<sup>30</sup> ist die mit Abstand größte Entwicklungsplattform für Open Source Projekte. Derzeit sind knapp 100 000 Projekte und mehr als 1 000 000 Benutzer bei Sourceforge registriert. Entwickler bekommen kostenfrei Speicherplatz und Entwicklungstools auf der Plattform zur Verfügung gestellt. Einzige Bedingung ist die Verwendung einer Open Source kompatiblen Lizenz für die bei Sourceforge gehosteten Projekte. Kommerzielle Software wird von Sourceforge in der Regel nicht beherbergt.

### **Freshmeat.net**

Freshmeat.net<sup>31</sup> ist ein Indexdienst, der sich hauptsächlich an Open Source Software richtet. Es kann sowohl hierarchisch (über Kategorien) als auch schlagwortbasiert gesucht werden. Der Schwerpunkt von Freshmeat liegt auf plattformunabhängiger Software, insbesondere Linux basierten Anwendungen.

<sup>28</sup> <http://www.ifross.de/>

<sup>29</sup> <http://www.berlios.de/>

<sup>30</sup> <http://www.sourceforge.net>

<sup>31</sup> <http://freshmeat.net/>

### 3 Open Source Software – Eine Ist-Analyse

Um die Frage nach der optimalen Strategie für Open Source Software beantworten zu können, ist es notwendig, zunächst ein Bild der derzeitigen Situation zu zeichnen. Im Folgenden werden die Ergebnisse einer Untersuchung vorgestellt, die in der Fraunhofer-Gesellschaft zum Thema Open Source Software durchgeführt wurde. Die Fraunhofer-Gesellschaft ist eine öffentliche Großforschungseinrichtung, die aus mehr als 50 einzelnen Einrichtungen mit zusammen über 12 000 Mitarbeitern besteht. Sitz der Gesellschaft ist München. Wichtigstes Tätigkeitsgebiet der Fraunhofer-Gesellschaft ist die anwendungsorientierte Forschung, sowohl für öffentliche als auch private Auftraggeber.

Als Grundlage für die Befragung wurde ein standardisierter Fragebogen entwickelt, der im Wesentlichen aus geschlossenen Fragen aufgebaut war und zur Beleuchtung der Ist Situation in der Fraunhofer-Gesellschaft insbesondere folgender Gebiete diente:

- Relevanz von Software-Einsatzgebieten
- Einsatz von Open Source Software in dem jeweiligen Einsatzgebieten
- Einschätzung von Vor- und Nachteilen von Open Source Software allgemein
- Zukünftig geplanter Einsatz von Open Source Software
- Einschätzung des relativen Reifegrades von Open Source Software gemessen an der jeweiligen kommerziellen Konkurrenz.
- Einsatz von Open Source Software in Kundenprojekten
- Eigenentwicklung von Open Source Software

Die Befragung wurde am 10. November 2004 im Rahmen einer IT Leiter Konferenz der Fraunhofer-Gesellschaft in Kassel durchgeführt. Es wurden insgesamt 55 Fragebögen ausgegeben. Ausgewertet werden konnten 42 Fragebögen, was einer Rücklaufquote von ca. 76% entspricht.

#### 3.1 Einsatzbereiche von Software

Im ersten Teil des Fragebogens stand vor allem die Erhebung der Wichtigkeit von Softwarearten bzw. Einsatzbereichen im Vordergrund. Dazu wurde der Begriff »Software« in insgesamt 16 Softwarearten unterteilt. Deren Wichtigkeit für das jeweilige Institut war vom Befragten mittels einer fünf-stufigen Ratingskala zu bewerten (wobei »1« für den niedrigsten und »5« für den höchsten Wert stand). Die nachstehend angeführten 16 Softwarearten dienten als Grundlage für den Aufbau des gesamten Fragebogens:

- Application Service Providing
- Application Server / Portalsoftware
- Betriebssysteme / Netzwerk
- Clusterlösungen / Grid Computing
- Content Management Systeme
- Customer/Supplier Relationship Management
- Datenbanksysteme
- Dokumenten und Workflow Management
- Enterprise Resource Planning Software
- Entwicklungssysteme
- Grafiksoftware
- Groupware
- Office Produkte
- Sprachverarbeitung
- Videosoftware
- Wissenschaftlich-Technische Software

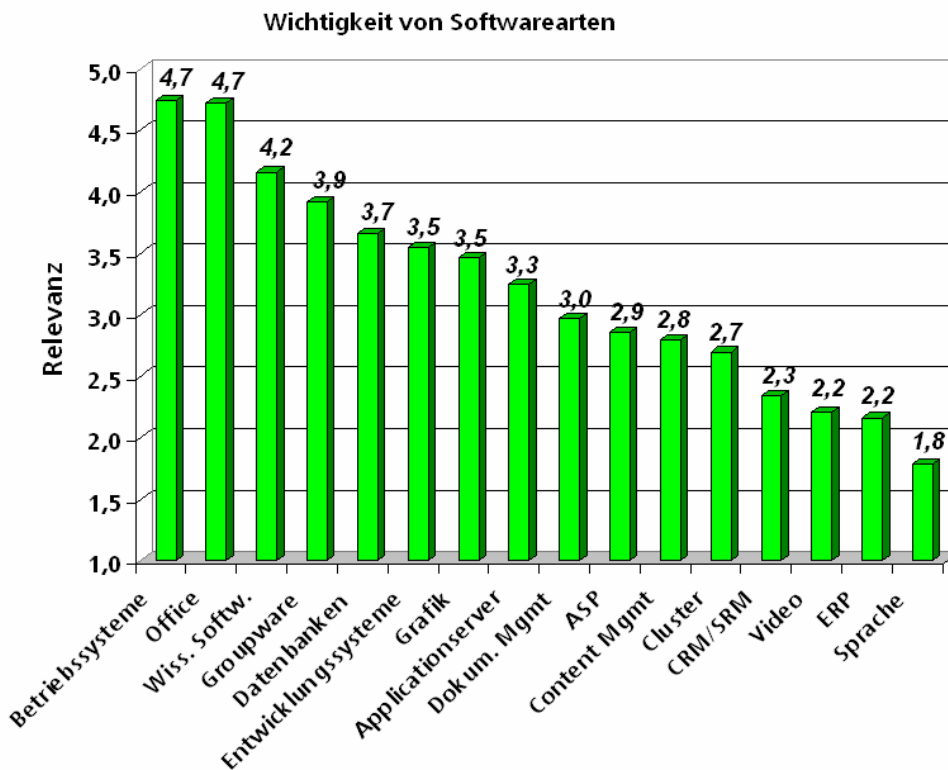


Abbildung 2: Wichtigkeit von Softwarearten für die Fraunhofer-Gesellschaft

Das Ergebnis der Frage nach der Relevanz von Softwarearten ist wenig überraschend. Die mit Abstand größte Relevanz mit einem Wert von jeweils 4,7 wird den beiden Bereichen »Betriebssysteme« sowie »Office« zugemessen. Mit einer Relevanz von 4,2 folgt der Bereich »Wissenschaftlich-Technische Software« auf Platz 3. Auch »Groupware« wurde von den Befragten als eine sehr wichtige Softwareart identifiziert (3,9). Mit Werten zwischen 3,7 und 3,3 liegen »Datenbanken«, »Entwicklungssysteme«, »Grafiksoftware« und »Application Server« im oberen Mittelfeld. Noch überdurchschnittliche Bedeutung wurde den Bereichen »Dokumentenmanagement«, »Application Service Providing«, »Content Management« und »Clustersoftware« zugewiesen (Werte zwischen 3,0 und 2,7). Die Bereiche »Customer/Supplier-Relationship« Management«, »Video-software«, »Enterprise Resource Planning« und »Sprachverarbeitung« markieren das untere Ende der Rangliste. Mit Werten zwischen 2,3 und 1,8 wurde diesen Softwarearten nur eine unterdurchschnittliche Bedeutung für die Fraunhofer-Gesellschaft zuerkannt.

### 3.2 Eigenschaften von Software

Zur Messung der Qualität von Software wird im Rahmen der Studie auf das FURPS Modell<sup>32</sup> zurückgegriffen. Dieses misst Softwarequalität anhand der folgenden fünf Indikatoren:

- Functionality (Funktionalität)
- Usability (Benutzerfreundlichkeit)
- Reliability (Zuverlässigkeit)
- Performance (Leistung)
- Support

Um die relative Wichtigkeit der fünf Indikatoren für die weitere Studie zu messen, wurde eine entsprechende Frage in den Fragebogen aufgenommen. Hierbei hatten die Befragten die fünf Faktoren in eine Reihenfolge zu bringen, wobei die »1« den höchsten und die »5« den niedrigsten Rang repräsentierte. Jeder Rang durfte nur einmal vergeben werden. Im Ergebnis wurde der Faktor Zuverlässigkeit/Sicherheit insgesamt als am wichtigsten betrachtet. 79% der Befragten vergaben die beiden höchstmöglichen Werte an diesen Faktor. Die Funktionalität wurde als der zweitwichtigste Faktor eingeschätzt. 64% der Befragten setzten diesen Faktor auf die beiden höchsten Plätze. Im Mittelfeld der Bewertungen liegt die Benutzerfreundlichkeit, der noch von 36% der Befragten mit einem der beiden höchsten Ränge versehen wurde. Am hinteren Ende der Rangliste befinden sich die Faktoren Leistung (14%) und Support (11%).

<sup>32</sup> Vgl. dazu auch: International Organization for Standardization (ISO): Software engineering – Product quality – Part I: Quality model. ISO/IEC 9126-1. Genf: ISO, 2001

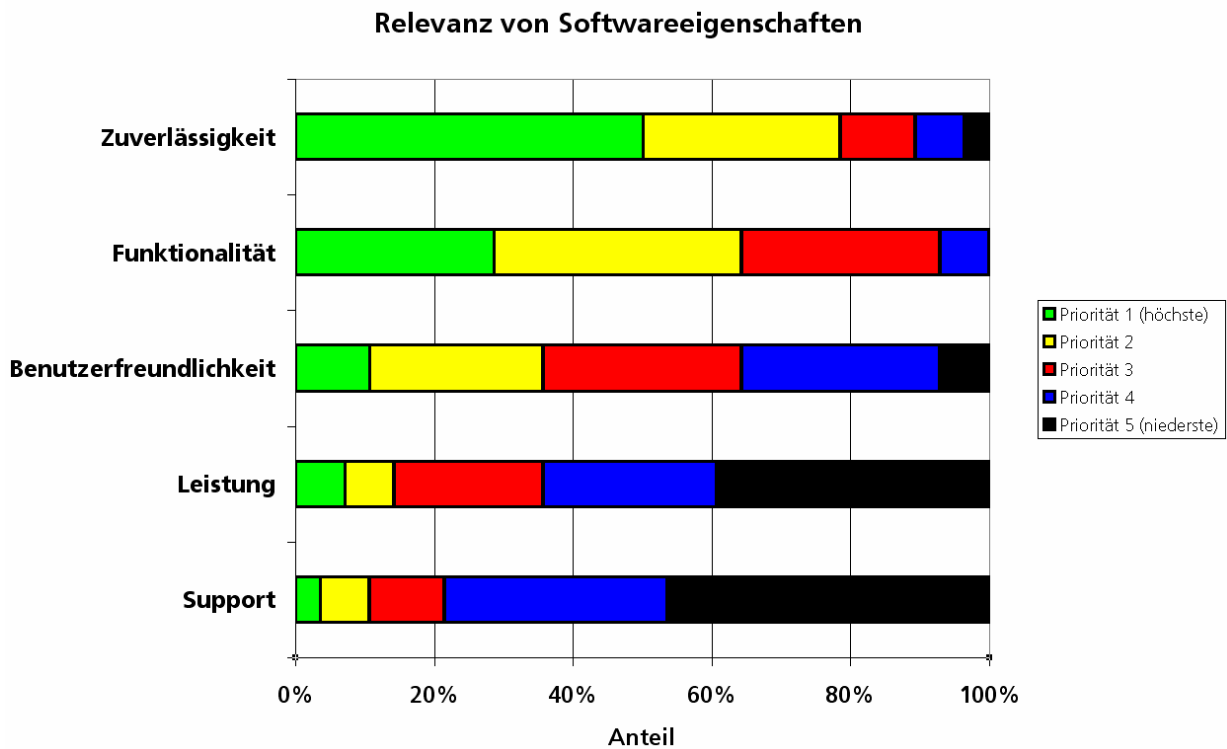


Abbildung 3: Relevanz von Softwareeigenschaften

### 3.3 Eingesetzte Produkte und Einsatzerfahrungen

Eine zentrale Fragestellung der Studie war es, ein Bild der in der Fraunhofer-Gesellschaft derzeit eingesetzten Software zu zeichnen. In die Ist-Aufnahmen sollten sowohl Open Source Produkte als auch kommerzielle Software aufgenommen werden – nicht zuletzt, um deren relative Bedeutung zueinander darzustellen. Es war jedoch nicht das Ziel der Befragung, eine mengenmäßige Abschätzung der installierten Basis vorzunehmen, dies wäre im Rahmen dieser Studie nicht möglich gewesen.

In der Befragung wurde zunächst zu jeder der betrachteten Softwarearten die wichtigsten Softwareprodukte aus beiden Bereichen identifiziert. Die Befragten hatten dann die Einsatzhäufigkeit des jeweiligen Softwareprodukts anhand einer fünf stufigen Ratingskala zu bewerten (mit 1 = sehr wenig im Einsatz und 5 = sehr häufig im Einsatz). Die zu betrachteten Softwareprodukte wurden im Rahmen einer Marktrecherche bzw. durch Expertengespräche identifiziert. Die Befragten hatten die Möglichkeit, weitere Softwareprodukte zu nennen, die nicht in der Liste enthalten waren.

### 3.3.1 Betriebssysteme

Wie kaum anderes zu erwarten, ist Microsoft Windows auch in der Fraunhofer-Gesellschaft das dominierende Betriebssystem. Mit einem Wert von 4,9 lag die gemessene Einsatzhäufigkeit sehr nahe am maximal möglichen. Das einzige weitere Betriebssystem, das es zu einer nennenswerten Verbreitung in der Fraunhofer-Gesellschaft gebracht hat, ist das Open Source Betriebssystem Linux mit einem Wert von 3,4. Die anderen Betriebssysteme sind eher als Nischenprodukte zu bezeichnen.

Als weitere, nicht in der Befragung enthaltene Betriebssysteme wurden noch Sun Solaris (17 Nennungen) und IBM AIX (3 Nennungen) angeführt.

Die folgende Tabelle gibt einen Überblick über die den Betriebssystemen jeweils zugrunde liegende Lizenzart

Software	Lizenztyp
Microsoft Windows	Kommerziell
Linux	Open Source
MAC OS	Kommerziell
Free BSD	Open Source
Open BSD	Open Source
OS/2	Kommerziell
Sun Solaris	Kommerziell
IBM AIX	Kommerziell

Tabelle 3: Betriebssysteme nach Lizenzen

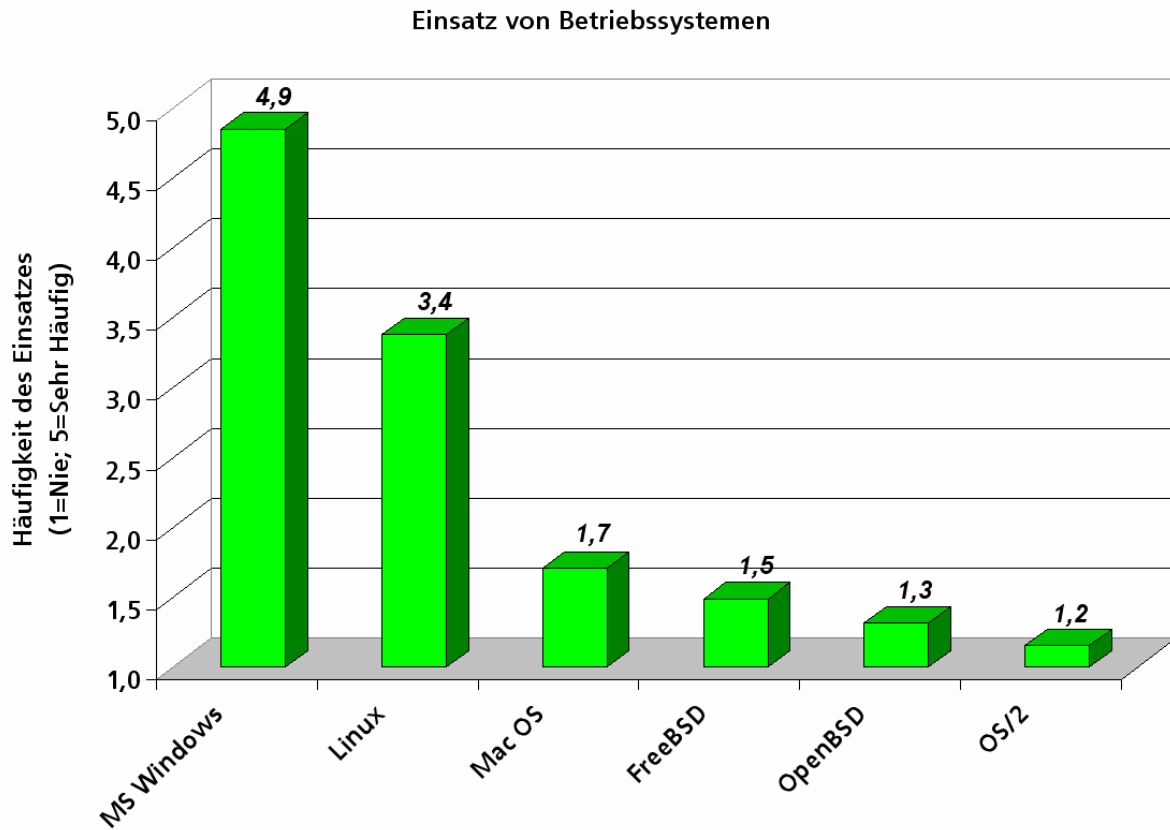


Abbildung 4: Einsatz von Betriebssystemen

### 3.3.2 Office Produkte

Das Bild bei Office Produkten ist noch eindeutiger als bei den Betriebssystemen. Der Marktführer Microsoft Office erreichte auch hier fast den maximal möglichen Wert von 4,9. Von den Alternativen bringt es noch das Open Source Produkt OpenOffice.org zu einer gewissen Relevanz (der Messwert in der Befragung war 2,2), die anderen Produkte sind praktisch unbedeutend.

Software	Lizenztyp
Microsoft Office	Kommerziell
OpenOffice.org	Open Source
KOffice	Open Source
Gnome Office	Open Source
Word Perfect Suite	Kommerziell
IBM Lotus Suite	Kommerziell

Tabelle 4: Office Produkte nach Lizenzen

Als weitere Nennung wurde noch Framemaker von vier Befragten als eingesetztes Office Produkt angeführt.

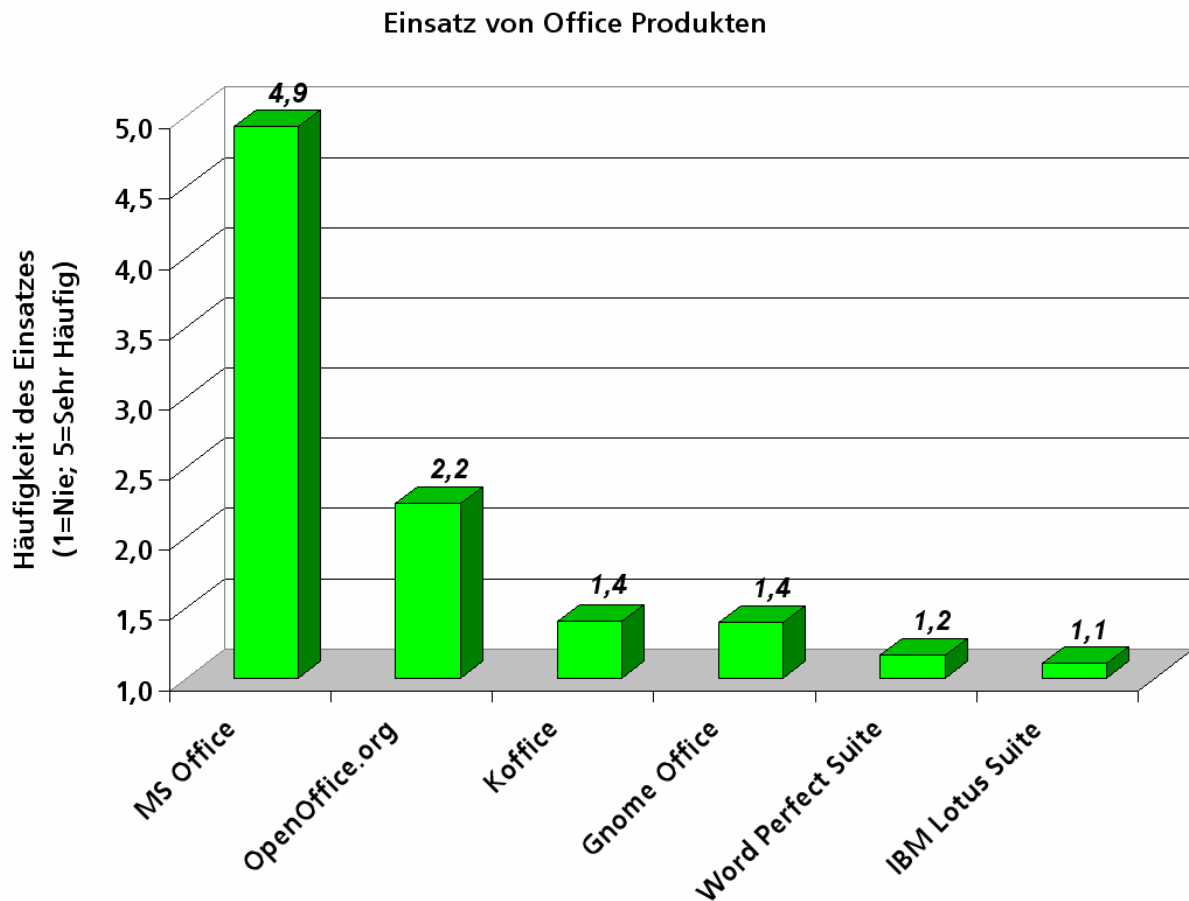


Abbildung 5: Einsatz von Office Produkten

### 3.3.3 Groupware

Auch im Bereich Groupware hat das Produkt aus dem Hause Microsoft derzeit die führende Position inne. In der Befragung erreichte der MS Exchange Server den Wert 3,1. Interessant ist, dass der Open Source Groupware Client Ximian Evolution noch vor dem renommierten Produkt Lotus Notes den zweiten Platz belegen konnte (1,4 bzw. 1,2), allerdings auf einem recht niedrigen Niveau.

Jeweils zwei Befragte nannten noch den SuSE OpenExchange Server und Novell Groupwise als eingesetztes Produkt.

Software	Lizenztyp
Microsoft Exchange Server	Kommerziell
Ximian Evolution	Open Source
Lotus Notes	Kommerziell
Open Groupware	Open Source
Oracle Suite	Kommerziell
PHP Groupware	Open Source
SuSE Open Exchange Server	Kommerziell
Novell Groupwise	Kommerziell

Tabelle 4: Groupware Produkte nach Lizenzen

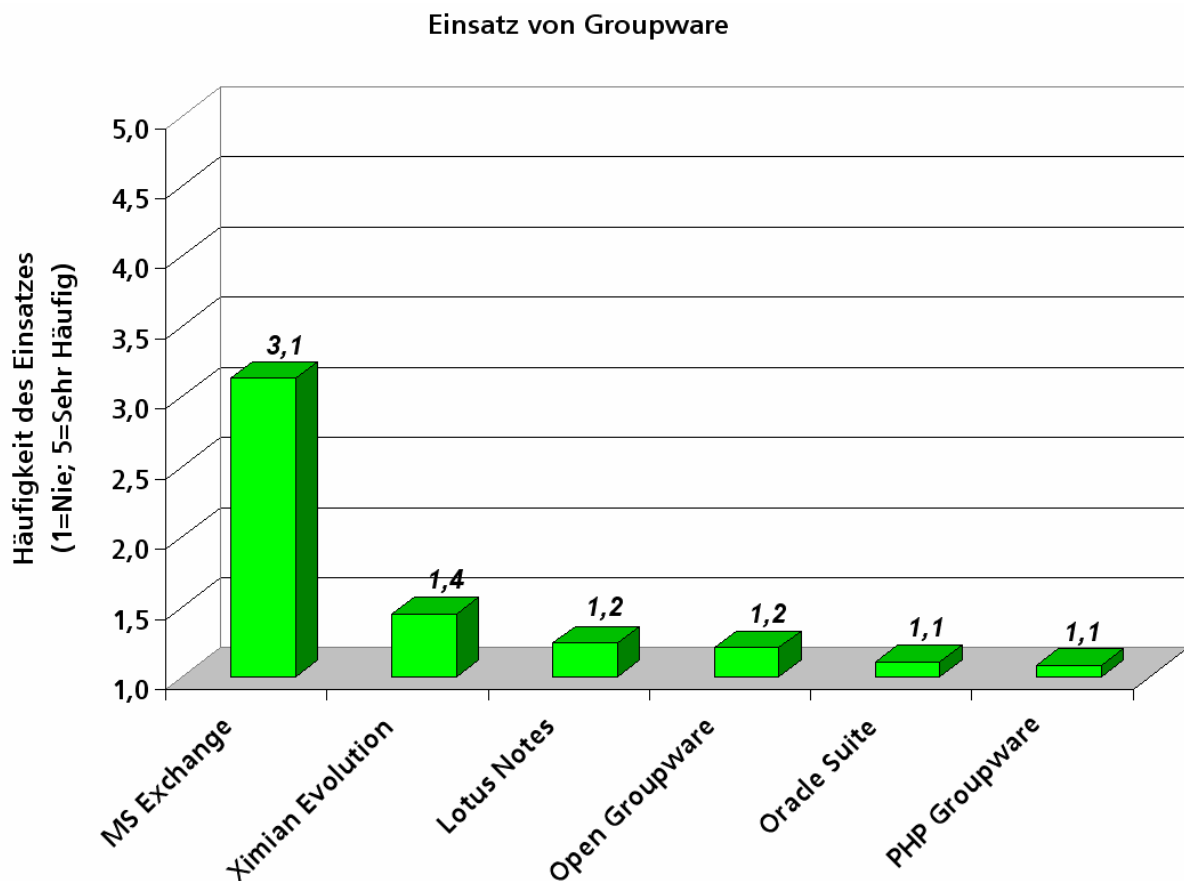


Abbildung 6: Einsatz von Groupware

### 3.3.4 Content Management Systeme

Content Management Systeme spielen derzeit eher noch keine allzu große Rolle in der Fraunhofer-Gesellschaft<sup>33</sup>. Sofern Content Management Systeme überhaupt eingesetzt werden, finden vor allem die beiden Produkte Zope und LiveLink Anwendung. Beide erreichten einen Messwert von 1,6. Alle anderen Systeme werden nur vereinzelt eingesetzt.

Weitere Systeme wurden von den Befragten nicht mehr als einmal genannt.

Software	Lizenztyp
Zope	Open Source
LiveLink	Kommerziell
Typo3	Open Source
Red Dot	Kommerziell
OpenCMS	Open Source
Open Text	Kommerziell
Infopark	Kommerziell

Tabelle 5: Content Management Systeme nach Lizenzen

<sup>33</sup> In der Befragung wurde der Fokus auf Web Content Management Systeme gelegt. Enterprise Content Management Software wurde im Rahmen der Studie nicht näher betrachtet.

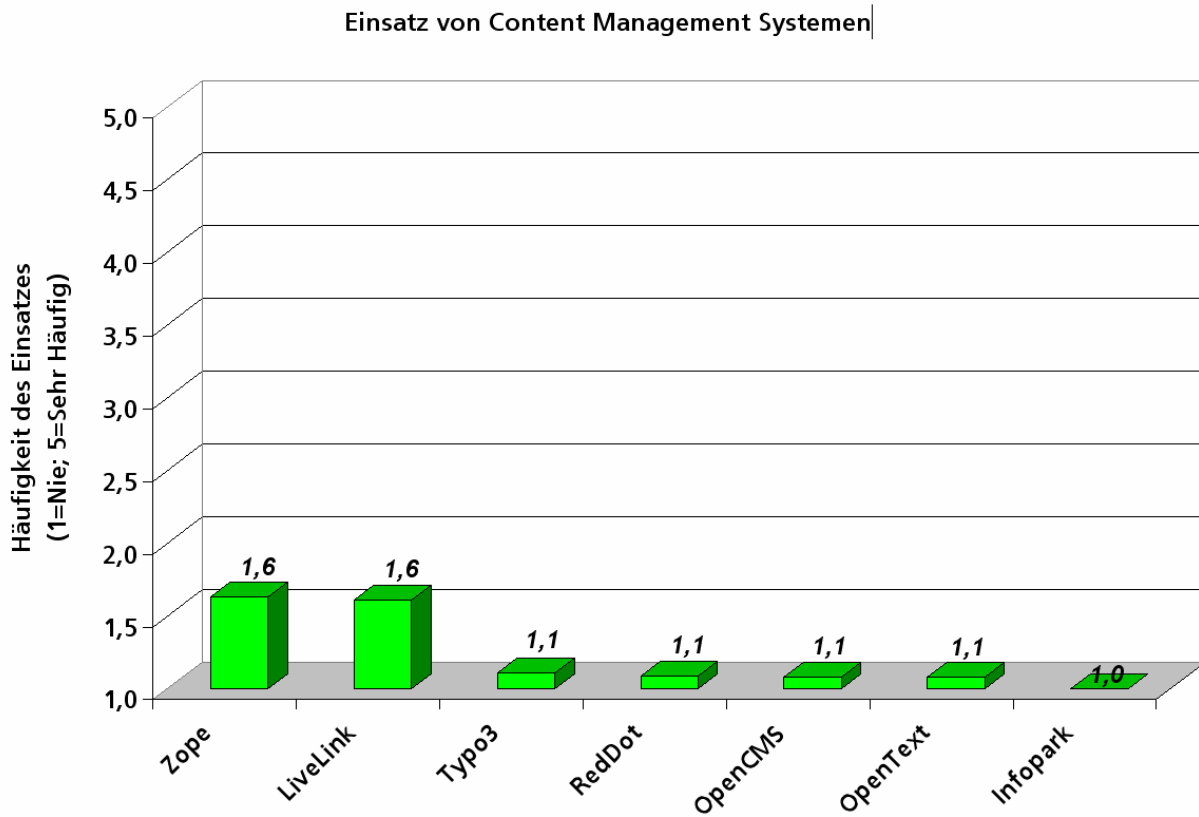


Abbildung 7: Einsatz von Content Management Systemen

### 3.3.5 Dokumenten Management Systeme

Auch für Dokumenten Management Systeme gilt, dass deren Verbreitung in der Fraunhofer-Gesellschaft noch nicht sehr weit gediehen ist. Das noch am häufigsten eingesetzte Produkt ist LiveLink (Messwert 1,6). Vereinzelt wird noch das Produkt aus dem Hause IXOS eingesetzt (1,2). Die anderen Systeme sind unbedeutend.

Software	Lizenztyp
LiveLink	Kommerziell
IXOS	Kommerziell
FileNet	Kommerziell
OpenEIS	Open Source
Dokumentum	Kommerziell
HyperWave	Kommerziell
SER	Kommerziell

Tabelle 6: Dokumenten Management Systeme nach Lizenzen

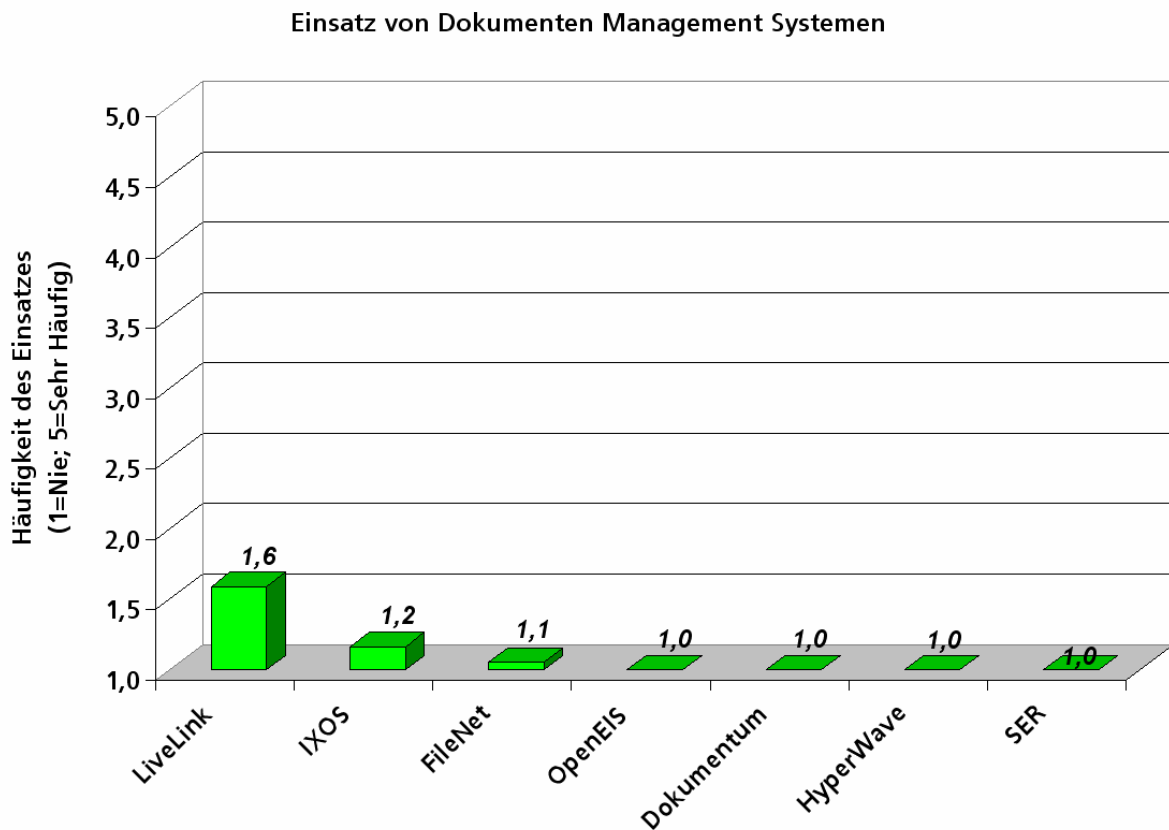


Abbildung 8: Einsatz von Dokumenten Management Systemen

### 3.3.6 Entwicklungssysteme

Der Markt für Entwicklungssysteme ist extrem groß, so dass im Rahmen der Befragung nicht alle bedeutenden Produkte berücksichtigt werden konnten. Es wurde daher versucht, einige wesentliche Softwareprodukte aus verschiedenen Bereichen in die Befragung aufzunehmen, um so die grundlegenden Tendenzen erkennen zu können.

Unter den Oberbegriff der Entwicklungssysteme fallen insbesondere

- Compiler und Interpreter
- Interaktive Entwicklungsumgebungen (IDEs)
- Debugger
- Werkzeuge für Software- und XML Modellierung
- Versionierungs- und Bugtrackingwerkzeuge

In die Befragung wurden nur die ersten beiden Bereiche aufgenommen.

Von den in die Befragung aufgenommenen Entwicklungssystemen wird die Skriptsprache Perl am häufigsten in der Fraunhofer-Gesellschaft eingesetzt. Das Produkt erzielte einen Messwert von 3,4. Auf dem zweiten Platz liegt PHP (3,3). Das nach diesen beiden Open Source Produkten am häufigsten eingesetzte kommerzielle Produkt ist Microsoft Visual Studio (3,1). Der wohl bekannteste Open Source C-Compiler GNU-C erreicht denselben Wert. Noch häufig wird bei der Entwicklung auch auf (das zwar kostenlose aber nicht freie) Produkt JAVA zurückgegriffen (2,7). Die Borland Produkte sowie die freie Entwicklungsumgebung Eclipse stellen das hinter Ende der Liste dar.

Als weitere Produkte sind noch Python und Tcl/Tk mit jeweils zwei Nennungen erwähnenswert.

<b>Software</b>	<b>Lizenztyp</b>
Perl	Open Source
PHP	Open Source
MS Visual Studio	Kommerziell
GNU C	Open Source
JAVA	Freeware
Borland C++/Delphi	Kommerziell
Eclipse	Open Source
Borland JBuilder	Kommerziell
Python	Open Source
Tcl/Tk	Open Source

Tabelle 7: Entwicklungssysteme nach Lizenzen

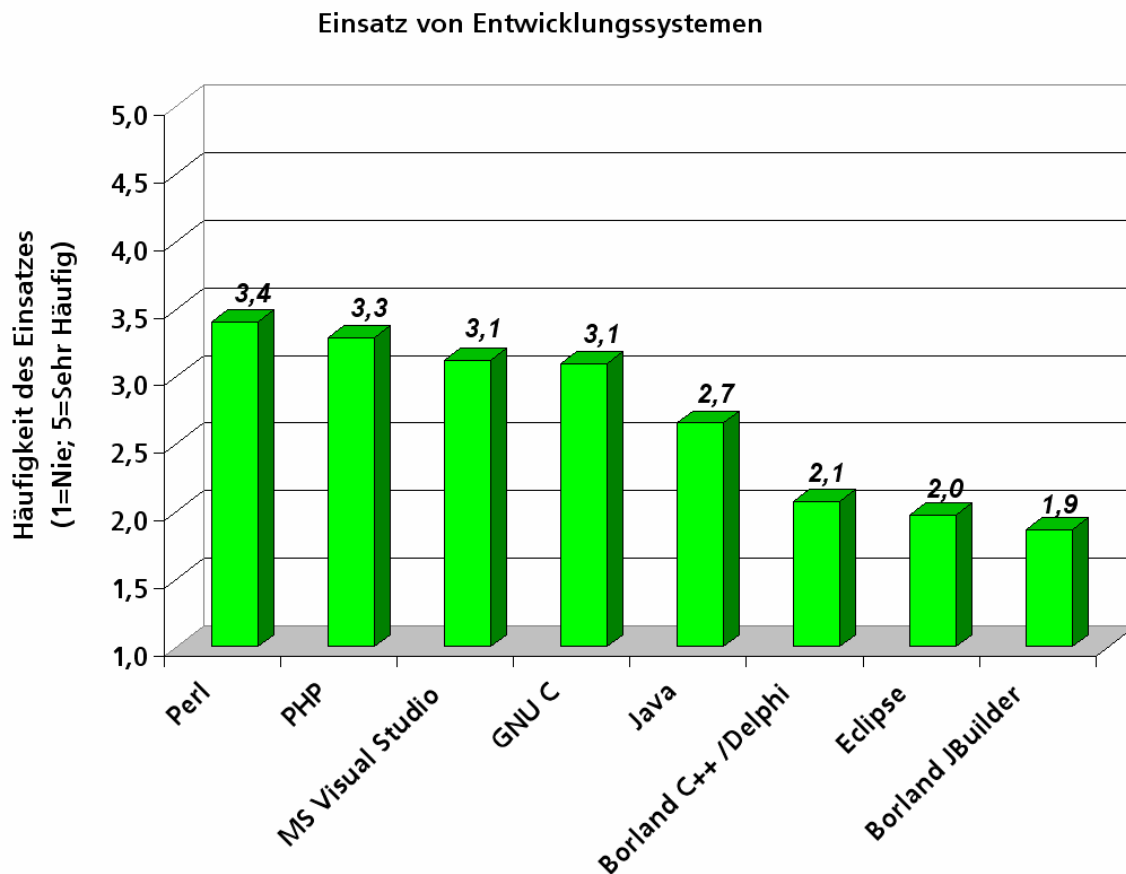


Abbildung 9: Einsatz von Entwicklungssystemen

### 3.3.7 Datenbanken

Auch im Bereich der Datenbanken konnten sich Open Source Produkte in der Fraunhofer-Gesellschaft bereits erfolgreich etablieren. Den höchsten Wert bei der Frage nach der Einsatzhäufigkeit erzielte das Open Source Datenbankmanagement System MySQL (3,3). Auf den zweiten Platz wurde die Desktop Datenbank aus dem Hause Microsoft, MS Access, gesetzt (3,0). Es folgten die beiden Datenbankmanagement Systeme Oracle sowie Microsoft SQL Server (2,8 bzw. 2,6). Ebenfalls noch in der Fraunhofer-Gesellschaft vertreten ist PostGreSQL (1,8). Die erst seit relativ kurzer Zeit unter einer Open Source Lizenz erhältliche SAP DB (Max DB) konnte sich dagegen in der Softwarelandschaft der Fraunhofer-Gesellschaft noch nicht durchsetzen (1,1).

Andere Datenbanken wurden nicht häufiger als einmal genannt.

Software	Lizenztyp
MySQL	Open Source
MS Access	Kommerziell
Oracle	Kommerziell
MS SQL Server	Kommerziell
PostgreSQL	Open Source
SAP DB	Open Source

Tabelle 8: Datenbanken nach Lizenzen

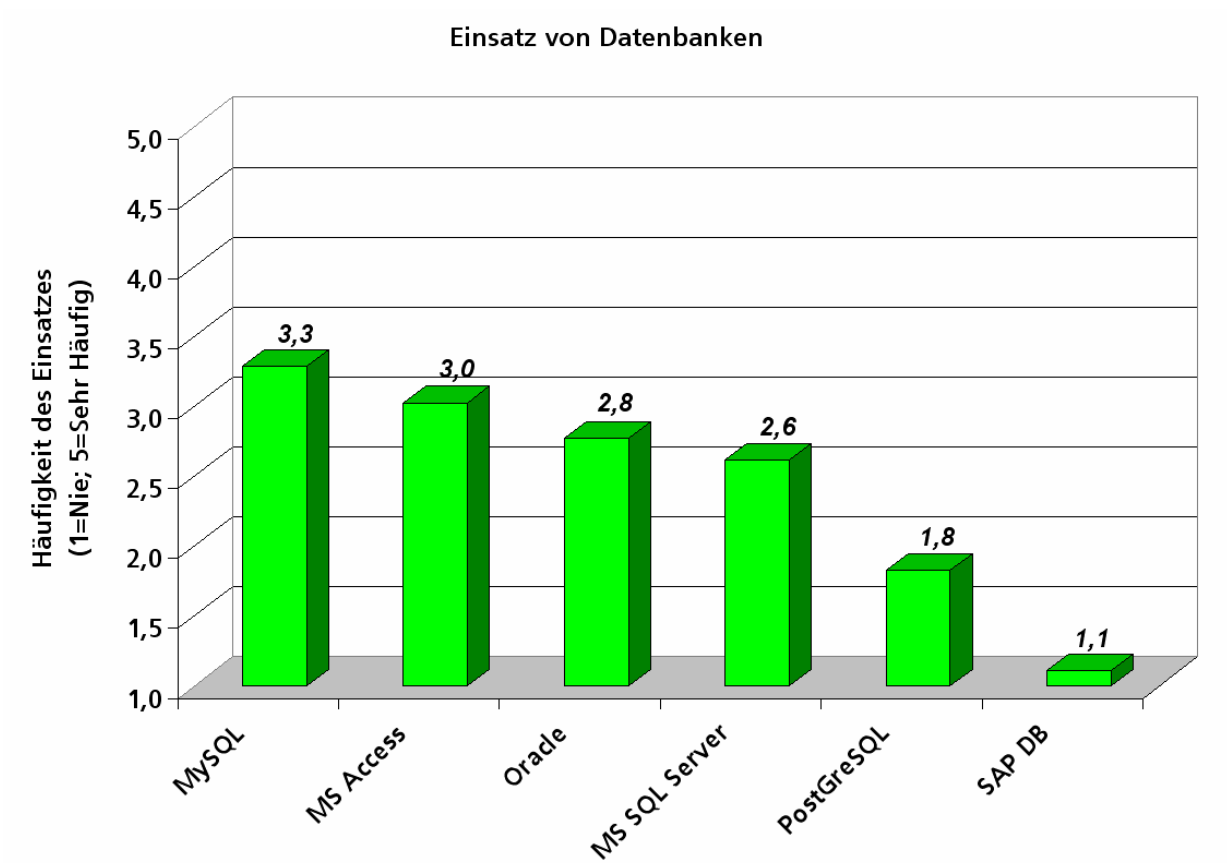


Abbildung 10: Einsatz von Datenbanken

### 3.3.8 Grafiksoftware

Der Markt für Grafiksoftware kennzeichnet sich (ähnlich den Entwicklungssystemen) durch eine sehr hohe Zahl von erhältlichen Softwareprodukten. Es konnten daher auch in diese Frage nur eine begrenzte Anzahl von Produkten aufgenommen werden.

Der Bereich Grafiksoftware wurde zum Zeitpunkt der Befragung in der Fraunhofer-Gesellschaft noch klar von kommerzieller Software dominiert. Der Marktführer Adobe stellte das am meisten eingesetzte Produkt (Messwert 3,4), gefolgt von Corel Draw (3,2). Schon deutlich geringere Einsatzhäufigkeiten konnte die Macromedia Suite verbuchen (2,2). Erst auf dem vierten Platz findet sich das wohl bekannteste Open Source Grafikprogramm GIMP (2,0).

Des Weiteren wurde von den Befragten noch Paintshop Pro (4 Nennungen) sowie Micrografx Designer (2 Nennungen) angeführt.

Software	Lizenztyp
Photoshop	Kommerziell
Corel Draw	Kommerziell
Macromedia Suite	Kommerziell
GIMP	Open Source
Premiere	Kommerziell
XFig	Open Source
3D Studio	Kommerziell
POV Ray	Open Source
Paintshop Pro	Kommerziell
Designer	Kommerziell

Tabelle 9: Grafiksoftware nach Lizenzen

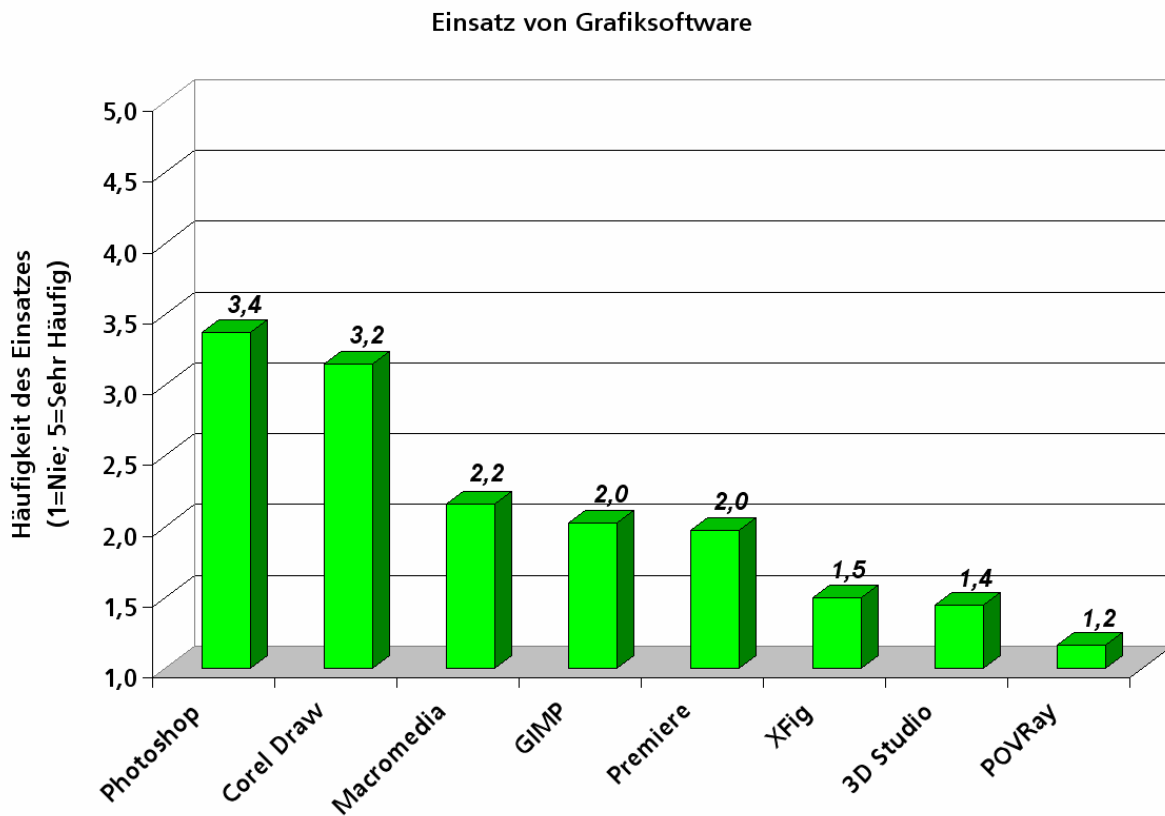


Abbildung 11: Einsatz von Grafiksoftware

### 3.3.9 Sprachsoftware

Die Verarbeitung von Sprache spielt derzeit nicht die Rolle in der Fraunhofer-Gesellschaft, die manche andere Softwareart einnimmt. Wirklich ausgereifte Open Source Produkte existieren in diesem Bereich ohnehin noch nicht und selbst die beiden marktbeherrschenden kommerziellen Produkte wurden von den Befragten sehr zurückhaltend bewertet, was die Einsatzhäufigkeit betrifft. Das Produkt aus dem Hause IBM brachte es gerade einmal auf einen Messwert von 1,3. Die Konkurrenz von Langenscheidt schnitt noch schlechter ab.

Weitere Sprachsoftware wurde nicht angeführt.

Software	Lizenztyp
IBM ViaVoice	Kommerziell
Langenscheidt	Kommerziell

Tabelle 10: Sprachsoftware nach Lizenzen

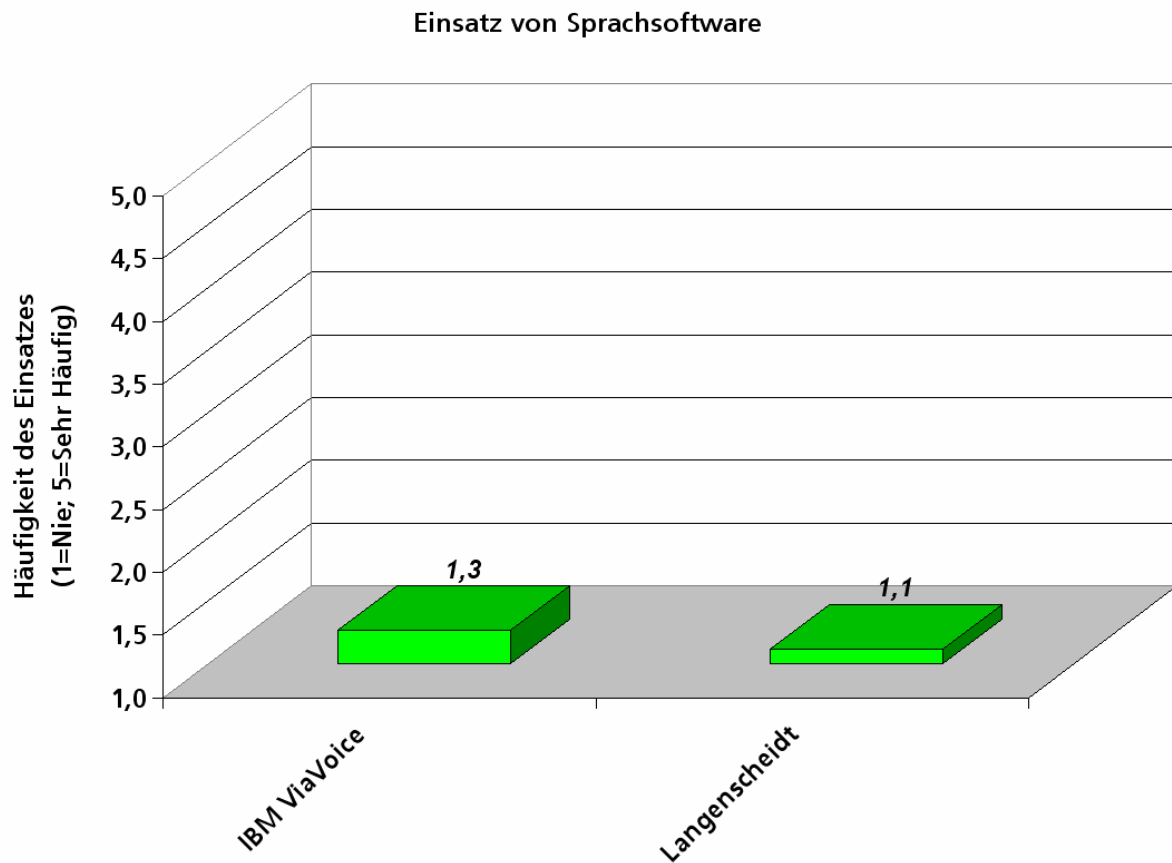


Abbildung 12: Einsatz von Sprachsoftware

### 3.3.10 ERP Systeme

Der Bereich der Enterprise Resource Management Systeme wurde im Rahmen der Befragung nur unter dem Gesichtspunkt »Einsatz in Projekten« betrachtet. Das für die internen Zwecke der Fraunhofer-Gesellschaft eingesetzte SIGMA wurde nicht in die Befragung einbezogen, da es ohnehin von der gesamten Organisation benutzt wird.

ERP Systeme gehören daher ebenfalls zum eher selten genutzten Softwaresegment in der Fraunhofer-Gesellschaft. Am ehesten kommt noch der Marktführer SAP zum Einsatz (Messwert 1,2). Das Open Source Produkt Compiere erzielte den Wert 1,1. Navision sowie Peoplesoft erreichten keine messbare Verbreitung.

Weitere ERP Software wurde nicht genannt.

Software	Lizenztyp
SAP	Kommerziell
Compiere	Open Source
Navision	Kommerziell
Peoplesoft	Kommerziell

Tabelle 11: ERP Systeme nach Lizenzen

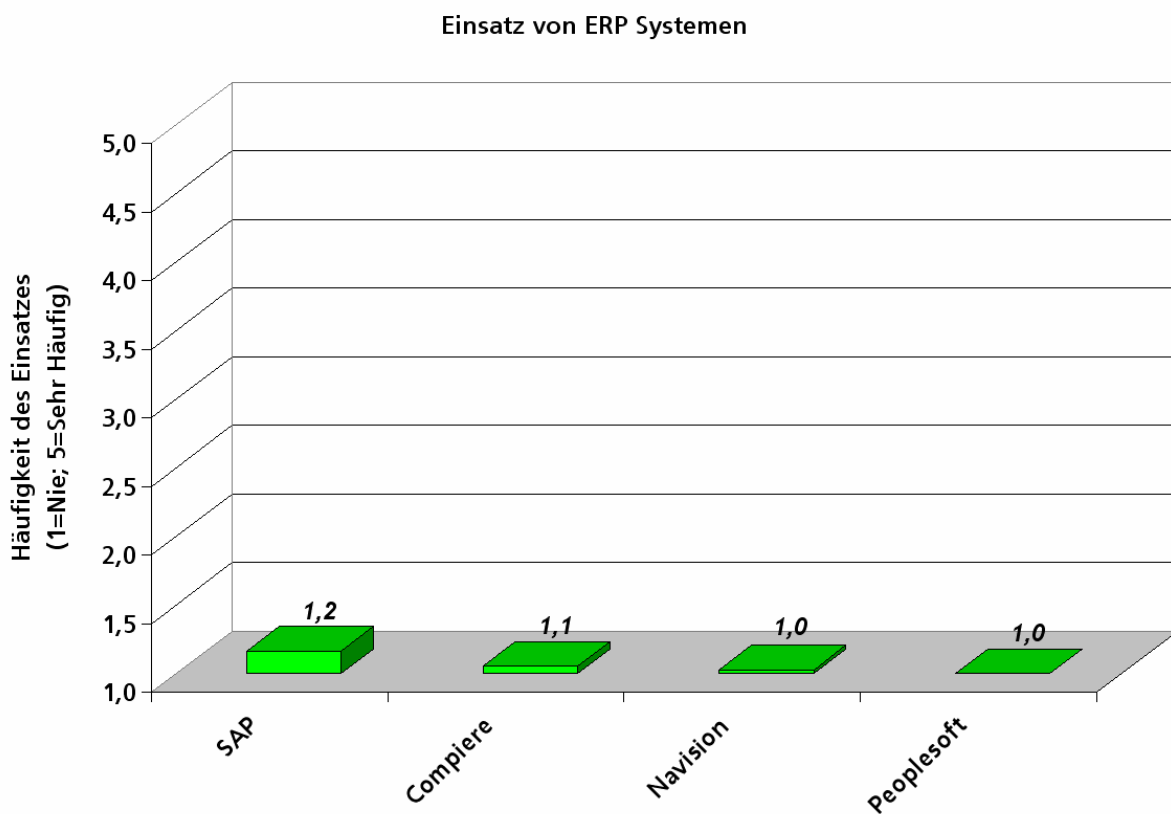


Abbildung 13: Einsatz von ERP Systemen in Projekten

### 3.3.11 Webbrowser

Im Bereich der Webbrowser hat auch in der Fraunhofer-Gesellschaft derzeit noch der Microsoft Internet Explorer die Nase vorn. Er führt die Liste der meisteingesetzten Browser mit einem Messwert von 4,2 an. Allerdings folgt die Open Source Alternative Mozilla Firefox bereits knapp dahinter (4,0). Das proprietäre Schwesterprodukt des Mozilla Browsers, Netscape, liegt mit einer

gemessenen Einsatzhäufigkeit von 3,0 im Mittelfeld. Opera sowie Konqueror spielen eher eine Nischenrolle (2,1 bzw. 1.9).

Weitere Web Browser wurden nur vereinzelt genannt.

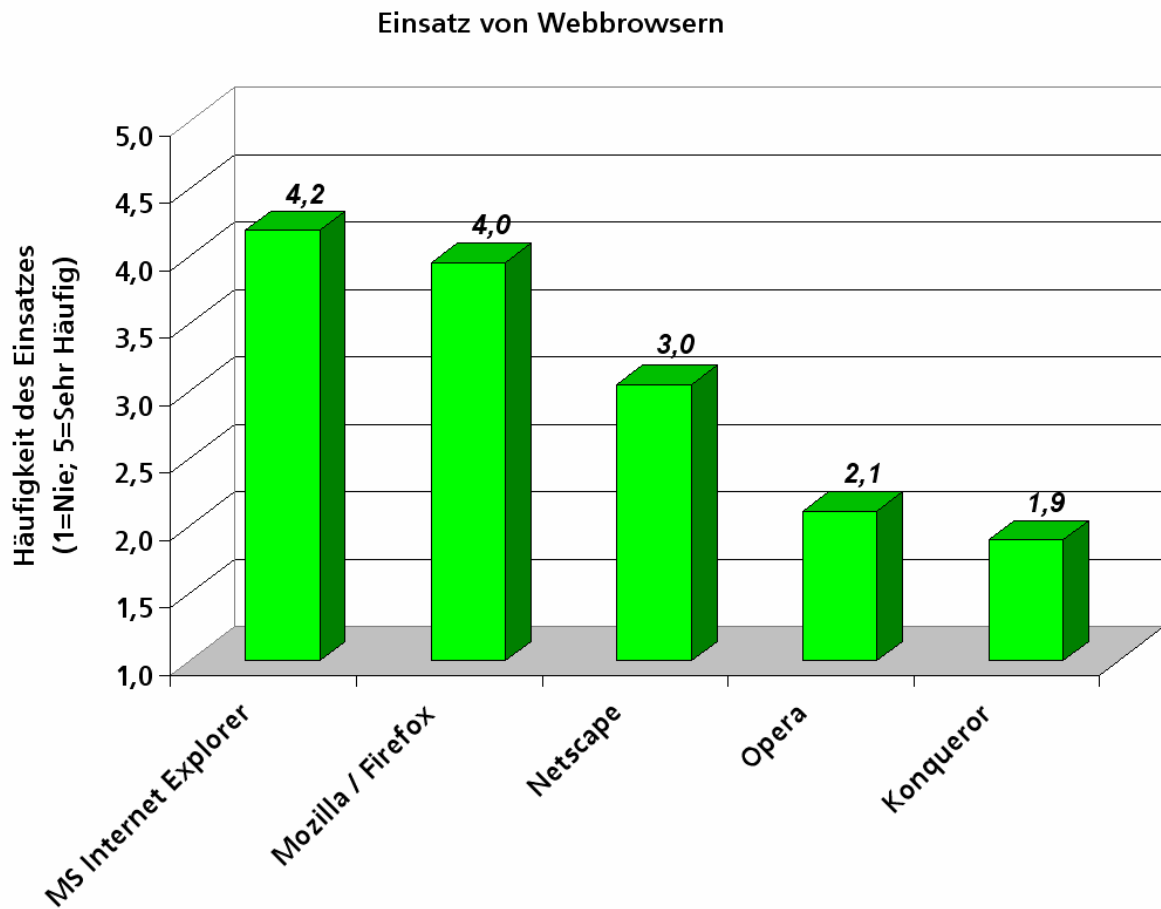


Abbildung 14: Einsatz von Webbrowsern

Software	Lizenztyp
MS Internet Explorer	Kommerziell (Bundle mit Windows)
Mozilla Firefox	Open Source
Netscape	Freeware
Opera	Kommerziell bzw. Freeware (mit Werbeeinblendungen)
Konqueror	Open Source

Tabelle 12: Web Browser nach Lizenzen

### 3.3.12 CRM Systeme

Customer Relationship Management Systeme sind eine weitere Softwareart, die es bisher noch nicht zu einer größeren Verbreitung in der Fraunhofer-Gesellschaft bringen konnte. Die Messwerte lagen allesamt nahe am möglichen Minimum. Vereinzelt werden noch die Produkte von SAP (mySAP CRM) sowie die Open Source Produkte Compiere und OpenCRX eingesetzt (Messwerte alle 1,1).

Einsatz von CRM Systemen

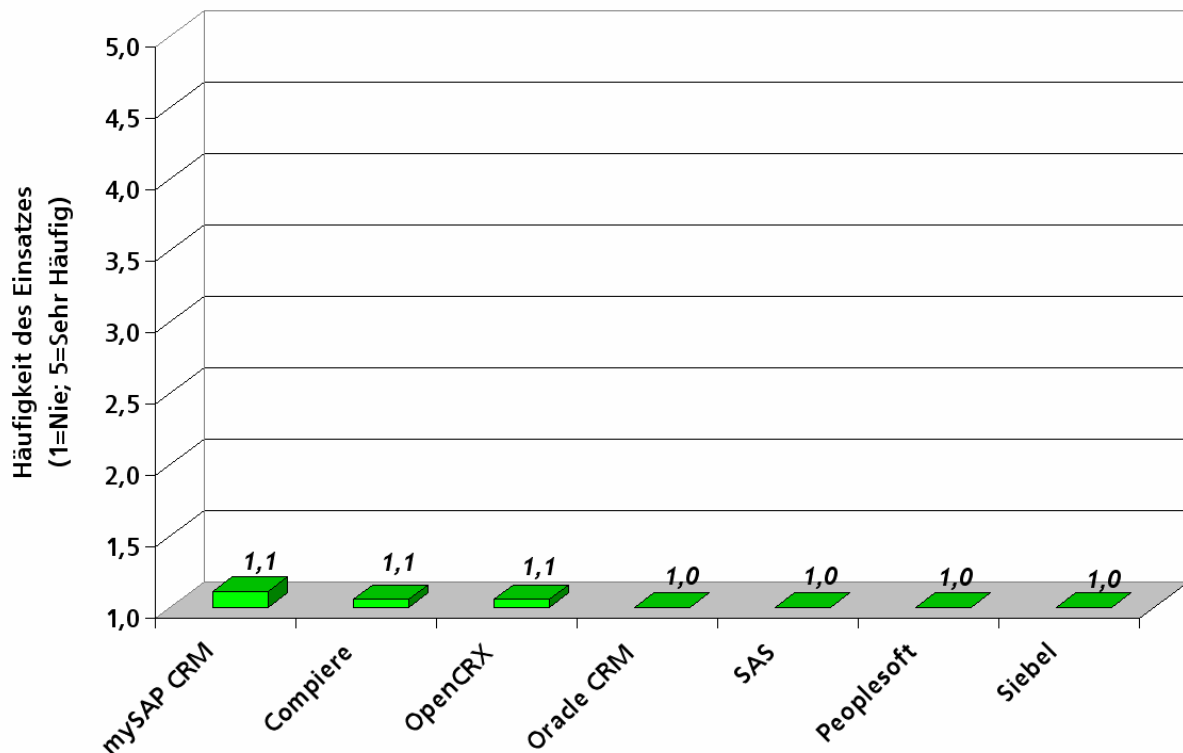


Abbildung 15: Einsatz von CRM Systemen

Software	Lizenztyp
mySAP CRM	Kommerziell
Compiere	Open Source
OpenCRX	Open Source
Oracle CRM	Kommerziell
SAS	Kommerziell
Peoplesoft	Kommerziell
Siebel	Kommerziell

Tabelle 13: CRM Systeme nach Lizenzen

### 3.3.13 SRM Systeme

Das Bild bei Supplier Relationship Management Systemen ist noch eindeutiger als bei den CRM Systemen. Diese Softwareart spielt praktisch keine Rolle in der Fraunhofer-Gesellschaft. Die Messwerte waren allesamt 1,0.

Software	Lizenztyp
mySAP SRM	Kommerziell
Ariba	Kommerziell
Peoplesoft	Kommerziell
SAS	Kommerziell
I2	Kommerziell

Tabelle 14: SRM Systeme nach Lizenzen

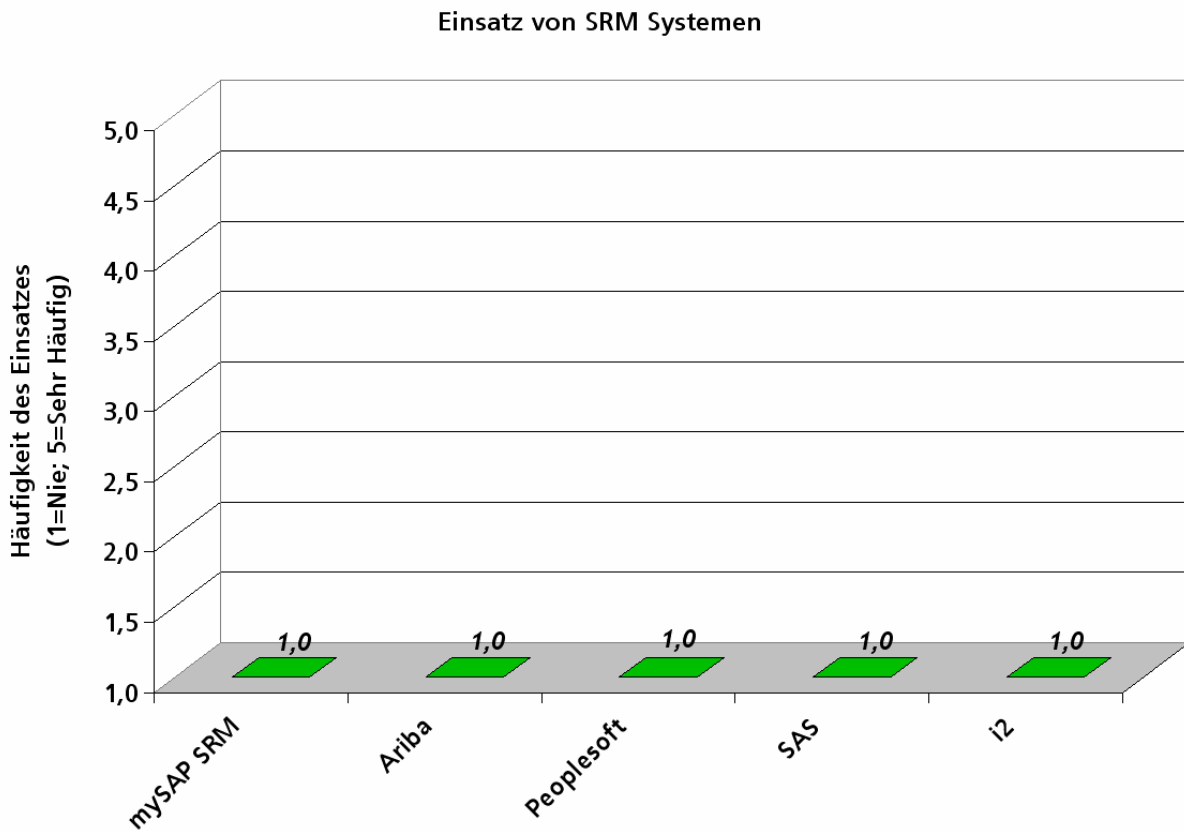


Abbildung 16: Einsatz von SRM Systemen

### 3.3.14 Wissenschaftlich-Technische Software

Ähnlich wie bei den Entwicklungssystemen ist der Begriff Wissenschaftlich-Technische Software eher als Überbegriff für eine ganze Reihe an Untersoftwarearten zu verstehen. Wissenschaftlich-Technische Software existiert etwa für die Bereiche Mathematik, Statistik, Konstruktion oder Simulation, um nur einige wenige zu nennen. Die Frage konnte also wiederum nur einen sehr kleinen Teil der diesem Segment zuzuordnenden Software abdecken. Insgesamt kann festgestellt werden, dass Wissenschaftlich-Technische Software eine große Rolle in der Fraunhofer-Gesellschaft spielt. Von den abgefragten Softwareprodukten erreichten die Mathematikpakete insgesamt die höchsten Einsatzhäufigkeiten. Derzeit ist dieses Gebiet aber noch von kommerzieller Software beherrscht.

Von den Befragten wurde auch Origin als noch häufig eingesetzte Software eingeschätzt (4 Nennungen).

Software	Lizenztyp
Mathematica	Kommerziell
AutoCAD	Kommerziell
MathLab	Kommerziell
GNU Plot	Open Source
SPSS	Kommerziell
SciLab	Open Source
SNNS	Open Source
IT++	Open Source
Origin	Kommerziell

Tabelle 15: Wissenschaftlich-Technische Software nach Lizenzen

### Einsatz von Wissenschaftlich-Technischer Software

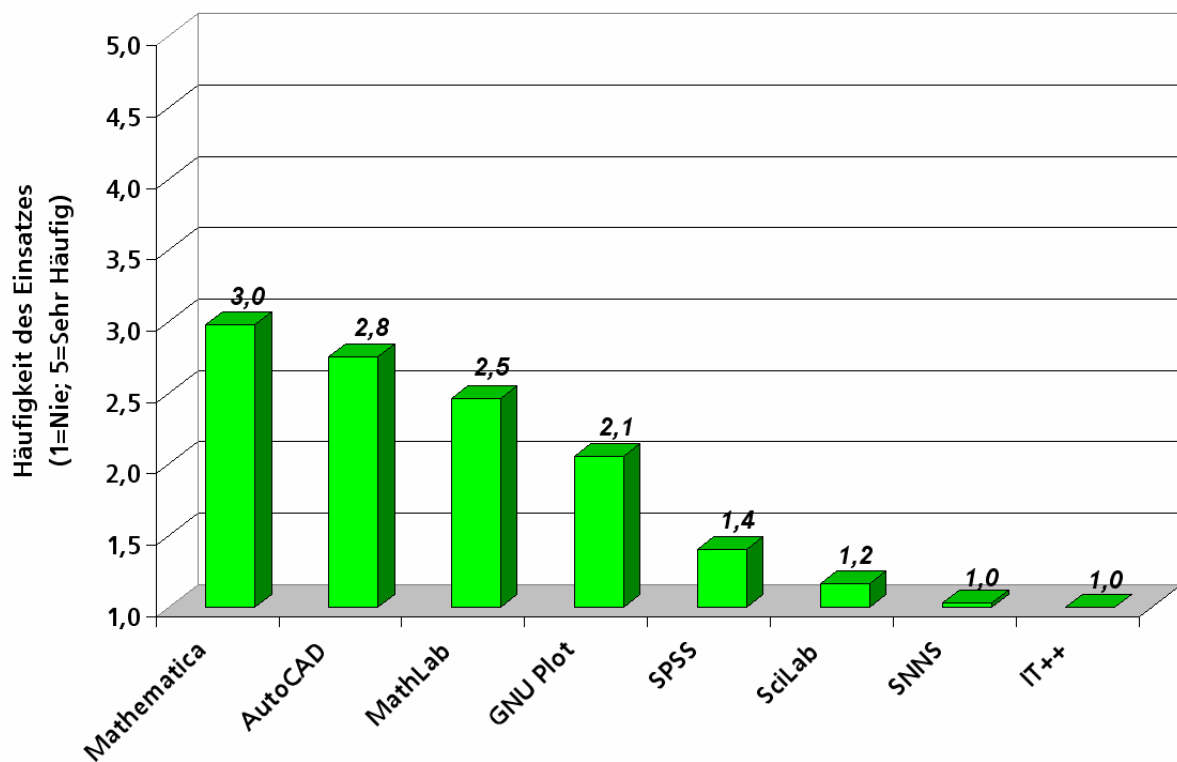


Abbildung 17: Einsatz von Wissenschaftlich-Technischer Software

### 3.3.15 Cluster / Grid Software

In Bezug auf die Anzahl der Installationen ist auch die Cluster und Grid Software nicht sehr bedeutsam in der Fraunhofer-Gesellschaft. Durch laufende Forschungsprojekte dürfte deren Anzahl in nächster Zeit aber noch steigen.

Den höchsten Messwert erreichte Oracle 10g mit 1,3. Es kann aber keine Aussage darüber getroffen werden, ob die neuen Grid Funktionalitäten der Oracle Software durch die Anwender auch genutzt werden. Den zweithöchsten Wert erreichte das GLOBUS Toolkit mit 1,1. Die anderen Produkte erreichten keine nennenswerte Verbreitung.

Von zwei Befragten wurde noch Mosix als eingesetztes Produkt auf dem Fragebogen angegeben.

Software	Lizentyp
Oracle 10g	Kommerziell
GLOBUS	Open Source
Unicore	Open Source
eXeGrid	Open Source
Avaki	Kommerziell
Entropia	Kommerziell
Platform Computing	Kommerziell
Mosix	Freeware

Tabelle 16: Cluster/Grid Software nach Lizenzen

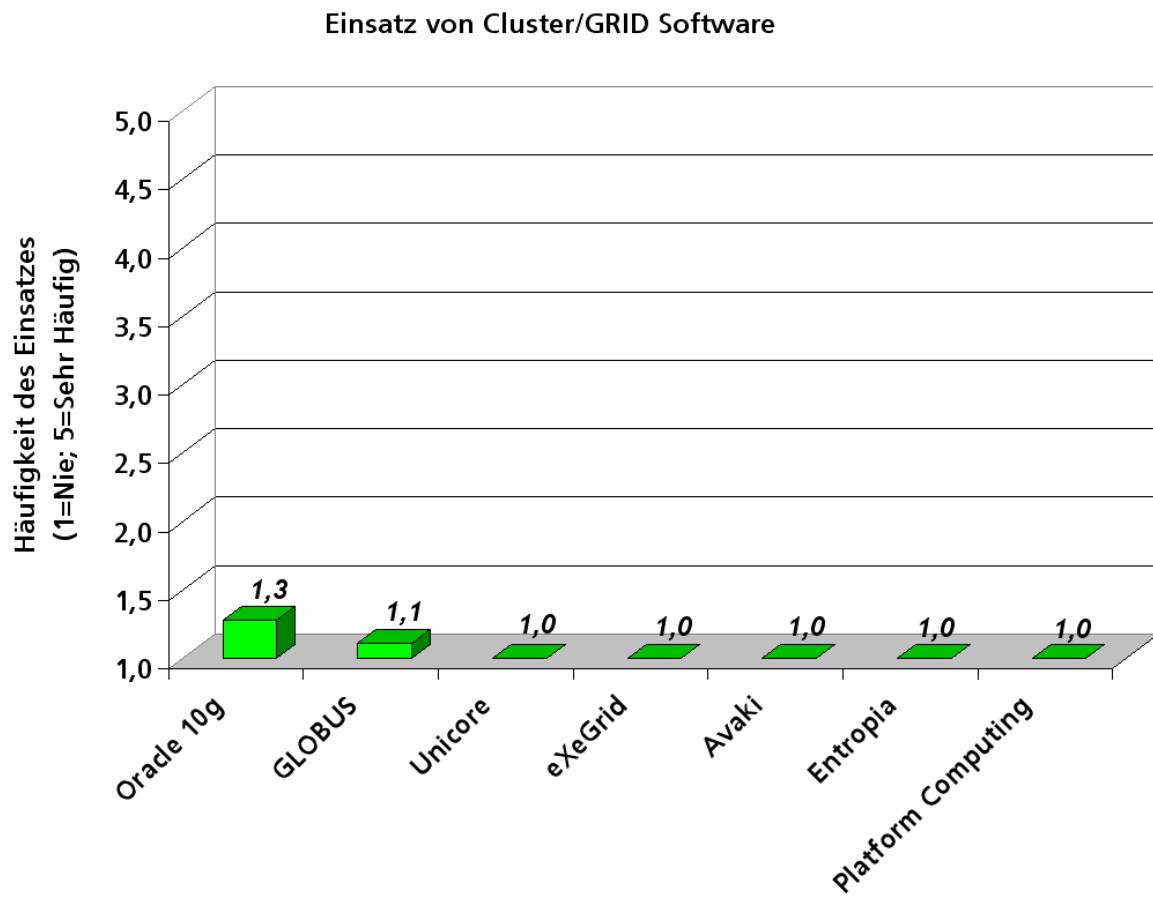


Abbildung 18: Einsatz von Cluster und Grid Software

### 3.4 Einschätzung von Open Source Software

Neben der reinen Einsatzhäufigkeit von Software wurde im Rahmen der Befragung auch die generelle Einschätzung der Befragten in Bezug auf Open Source Software abgefragt. Die in Zusammenhang mit Open Source Software häufig genannten Pro und Contra Argumente wurden den Befragten zur Zustimmung bzw. Ablehnung vorgelegt. Zur Bewertung wurde eine vier-stufige Ratingskala verwendet, die vom Wert 1 (trifft nicht zu) bis 4 (trifft voll zu) reichte.

#### 3.4.1 Vorteile von Open Source Software

Was die Vorteile von Open Source Software anbetrifft, so kann auch in der Fraunhofer-Gesellschaft das Argument »Keine Lizenzkosten« den höchsten Zustimmungsgang verbuchen. Auf der Skala erreichte dieses Kriterium einen Wert von 3,6. Auch das Kriterium »Hohe Interoperabilität« konnte mit 3,2 eine sehr hohe Zustimmung verbuchen. Die »Unabhängigkeit von einem bestimmten Hersteller« wurde ebenfalls als ein Vorteil von Open Source Software gesehen – der Messwert lag bei 3,1. Ebenfalls noch hohe Zustimmung erntete das Kriterium »Support durch Entwickler« (3,0). Das recht oft genannte Argument, Open Source Software sei aufgrund des offenen Quellcodes sicherer als vergleichbare kommerzielle Produkte, fand hingegen nur teilweise Unterstützung (2,6). Auch ist die Haltung zu der These, dass Open Source Software qualitativ höherwertiger ist, als kommerzielle Software, durchaus uneinheitlich (2,5). Interessanterweise liegt das von vielen Open Source Entwicklern und –Befürwortern oft vertretene Kernargument der leichten Anpassbarkeit und Weiterentwickelbarkeit von Open Source Software mit einem Messwert von 2,4 auf dem letzten Platz.

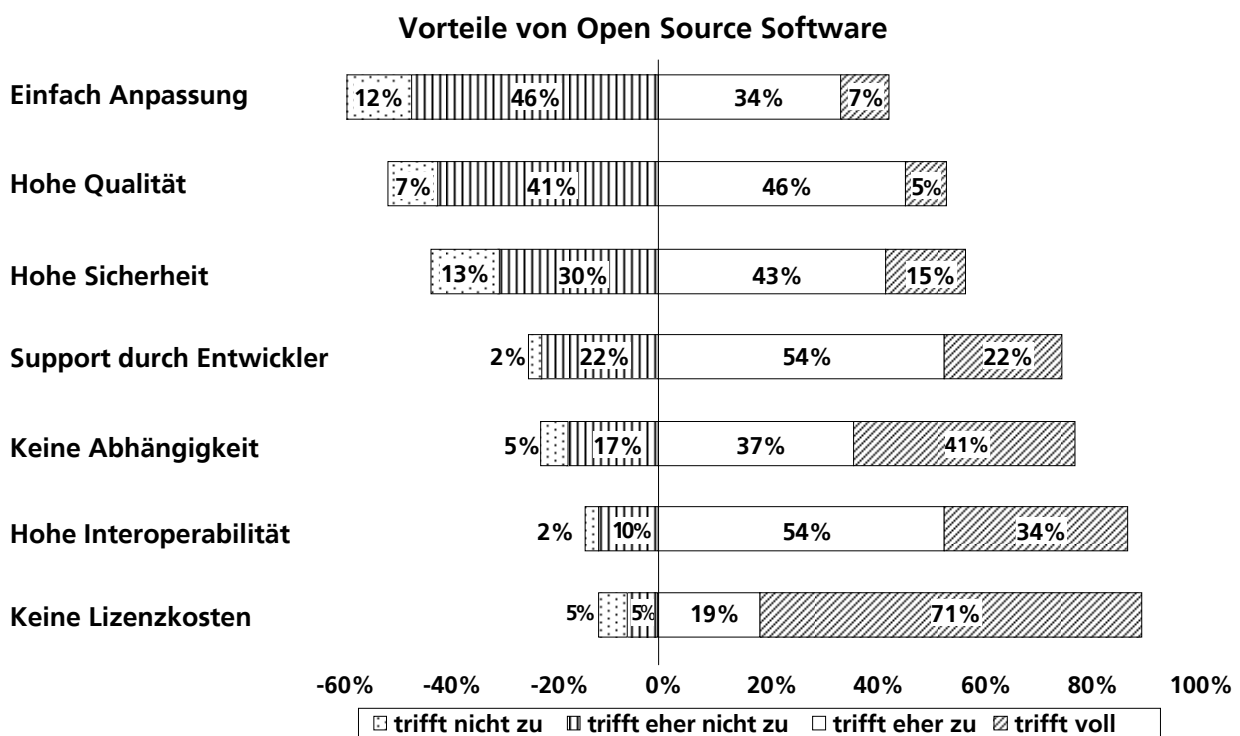


Abbildung 19: Vorteile von Open Source Software

### 3.4.2 Nachteile von Open Source Software

Oftmals wird als ein zentraler Nachteil von Open Source Software die ungewisse Weiterentwicklung genannt – schließlich kann ein Open Source Projekt von seinen Entwicklern jederzeit aufgegeben werden. Dieses Kriterium wurde auch von den Befragten der Fraunhofer-Gesellschaft als am meisten zutreffend gesehen. Es erreichte einen Messwert von 2,8, genau wie das Kriterium der fehlenden Haftungs- und Gewährleistungsrechte. Knapp dahinter wurde der Nachteil des oft fehlenden Supports von Open Source Software eingeordnet (2,7).

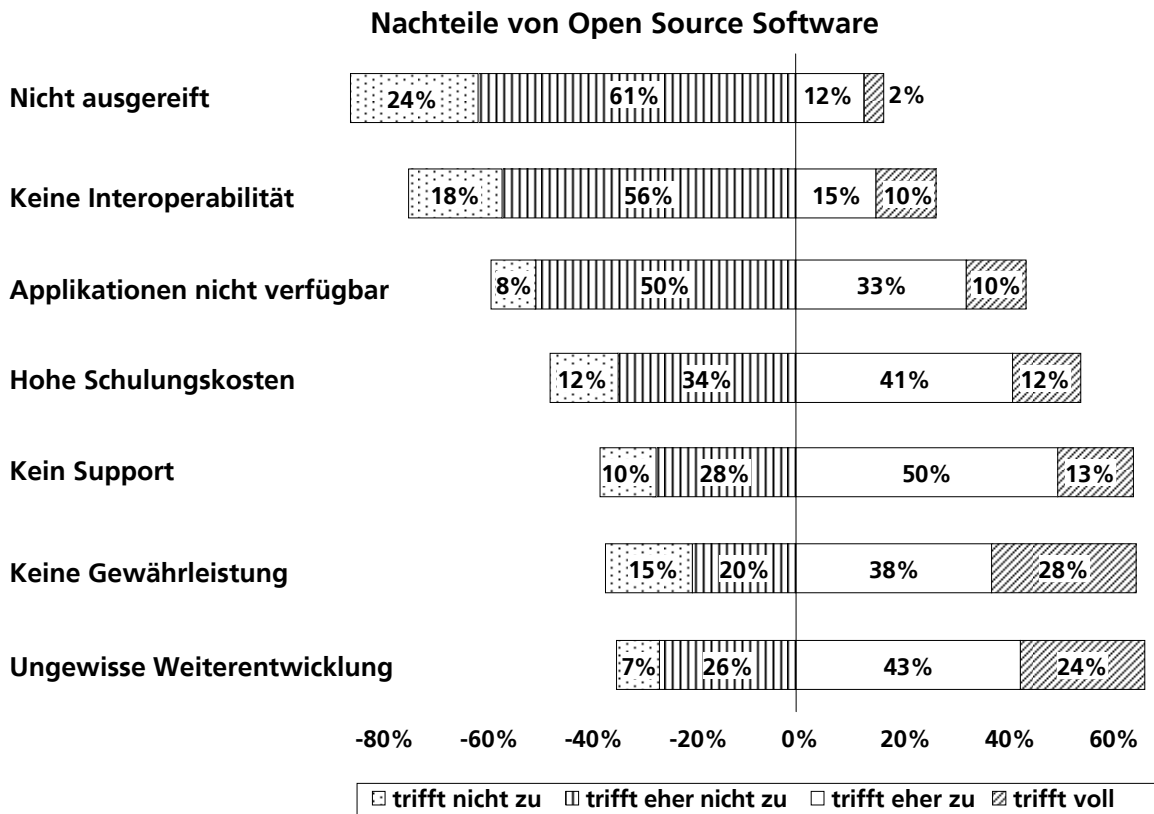


Abbildung 20: Nachteile von Open Source Software

Die hohen Schulungskosten erreichten einen Messwert von 2,5, gleichrangig mit dem Kriterium der mangelnden Verfügbarkeit von Applikationen. Mit etwas Abstand wurde auch die mangelnde Interoperabilität von Open Source Software mit (kommerzieller) Software als Nachteil betrachtet (2,2). Am wenigsten folgen mochten die Befragten der These, dass Open Source Software generell noch nicht ausgereift sei (1,9).

### 3.4.3 Reife von Open Source Software

Neben den Einsatzhäufigkeiten von Open Source Software und den wahrgenommenen Vor- und Nachteilen wurde auch die Einstellung der Befragten zur Reife von Open Source Software abgefragt. Die Reife von Software wurde wiederum anhand der aus dem FURPS Modell abgeleiteten fünf Indikatoren für Softwarequalität

- Functionality (Funktionalität)
- Usability (Benutzerfreundlichkeit)
- Reliability (Zuverlässigkeit)
- Performance (Leistung), sowie
- Support

gemessen. Den Befragten wurde dazu wieder eine fünf-stufige Ratingskala zur Bewertung des jeweiligen Reifegrades vorgelegt (1 = Reifegrad sehr niedrig; 5 = Reifegrad sehr hoch). Der Reifegrad sollte dabei relativ zu relevanten kommerziellen Softwareprodukten betrachtet werden. Ziel der Frage war es, eine Aussage über die Ausgereiftheit von Open Source Software untergliedert nach Softwarearten machen zu können. Dies kann ein erster Anhaltspunkt sein, in welchen Bereichen über den Einsatz von Open Source Software nachgedacht werden kann und sollte.

## Reifegrad der Funktionalität

In Bezug auf die Funktionalität wurden die Open Source Betriebssysteme als am ausgereiftesten betrachtet (Messwert 4,4). Auch den freien Entwicklungssystemen und Datenbanken wird eine sehr hohe funktionale Reife zugesprochen (4,1 bzw. 4,0). Das recht breite Mittelfeld wird von den Office Paketen angeführt (3,8) und reicht bis zur Groupware (3,2). Noch akzeptable Werte erreichte Open Source Software in den Bereichen Application Service Providing Software und Dokumenten Management Systeme (jeweils 2,9). Als funktional nicht sehr ausgereift wurden freie CRM/SRM Systeme, Videosoftware, ERP Systeme und Sprachverarbeitung betrachtet (Messwerte zwischen 2,5 und 2,3).

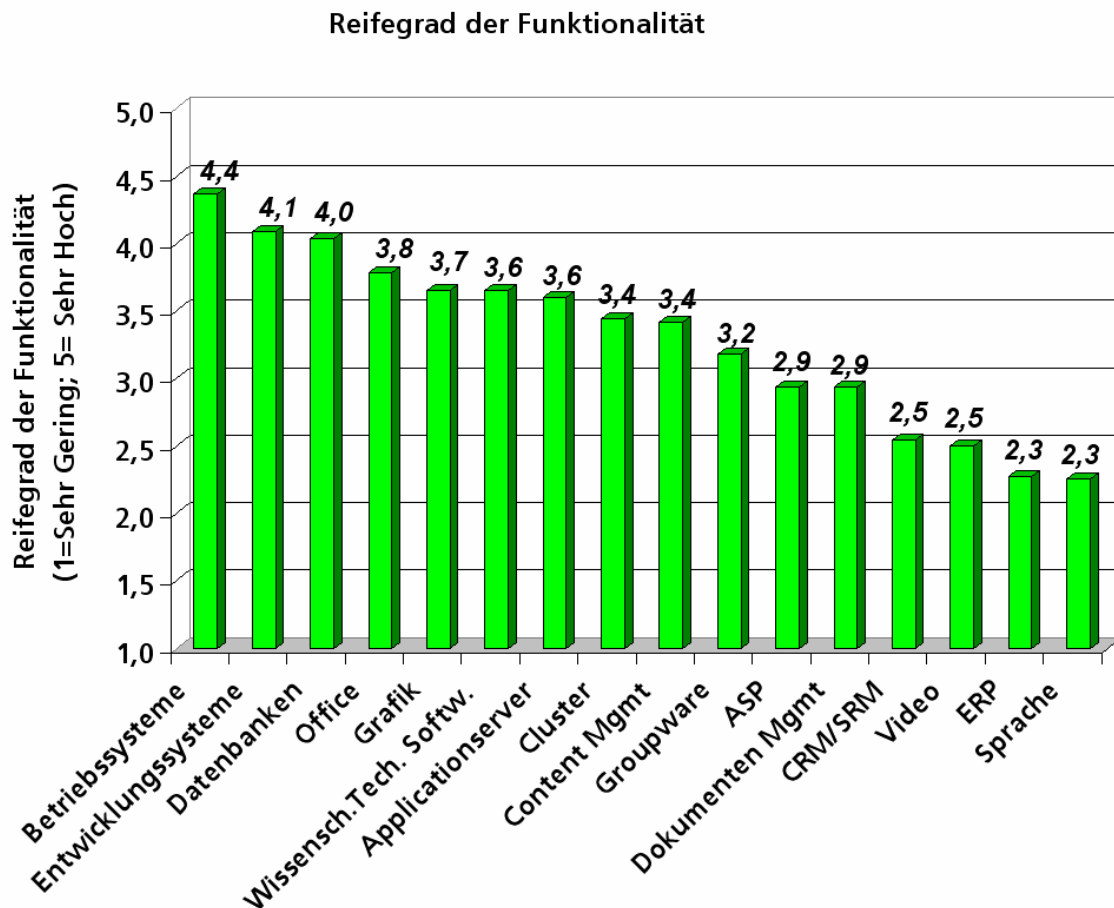


Abbildung 21: Reifegrad der Funktionalität

### Reifegrad der Benutzerfreundlichkeit

Am benutzerfreundlichsten ist Open Source nach der Ansicht der Befragten in den Bereichen Office Produkte sowie Betriebssysteme. Beide Softwarearten erreichten einen Messwert von 3,9. Auch freie Entwicklungssysteme und Datenbanken können mit kommerzieller Software gut mithalten. Diese erreichten einen Messwert von jeweils 3,7. Der Spitzengruppe sind ferner noch die Grafiksoftware sowie die Wissenschaftlich-Technische Software zuzurechnen (3,5 bzw. 3,4). Mittlere Benutzerfreundlichkeit haben nach Ansicht der Befragten die freien Application Server, Groupware Applikationen, Content Management Systeme sowie die Cluster und ASP Lösungen zu bieten (Messwerte von 3,1 bis 2,9). Am wenigsten anwenderfreundlich wurden Open Source CRM/SRM Lösungen, Dokumenten Management Systeme sowie Video- und Sprachapplikationen gesehen (2,7 bis 2,4). Schlusslicht sind ERP Systeme mit einem Wert von 2,3.

### Reifegrad der Benutzerfreundlichkeit

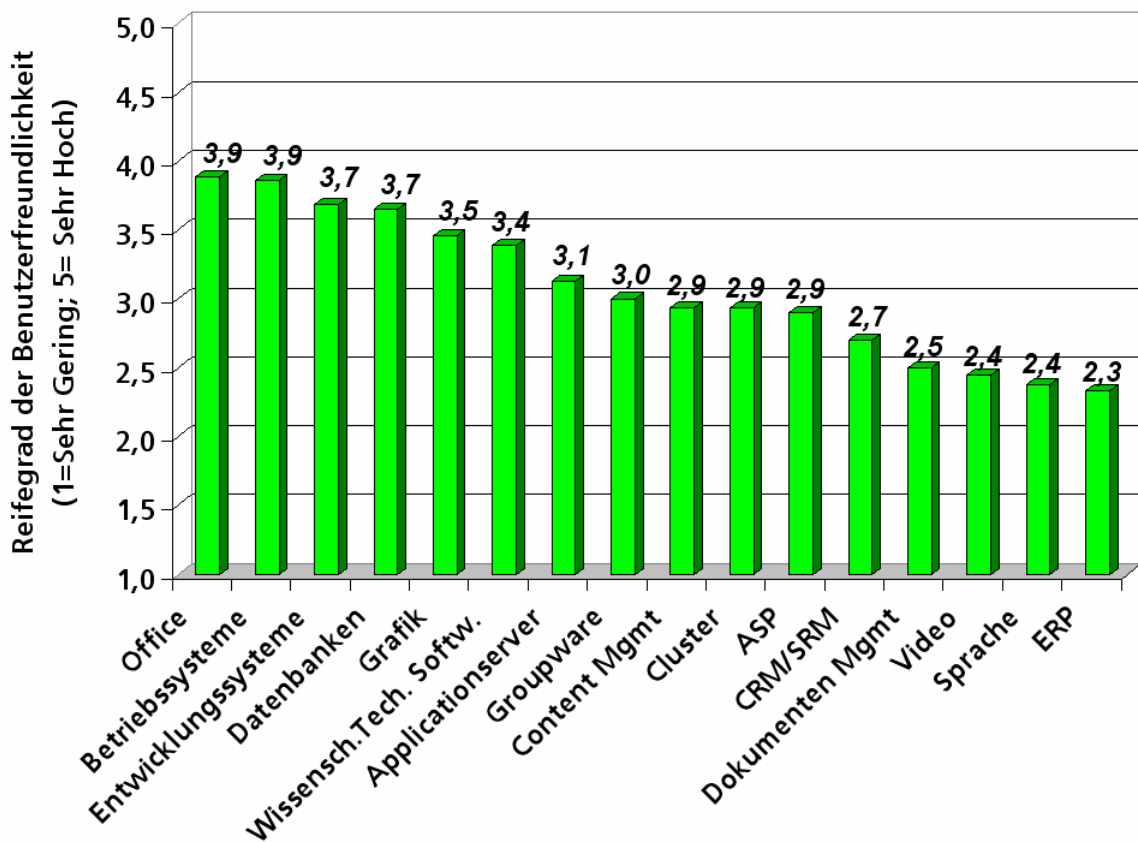


Abbildung 22: Reifegrad der Benutzerfreundlichkeit

### Reifegrad der Zuverlässigkeit

Als relativ zu kommerziellen Produkten am zuverlässigsten und sichersten wurden die freien Betriebssysteme, Datenbanken und Entwicklungssysteme betrachtet (Messwerte 4,5 bis 4,3). Ebenfalls noch gute Werte erreichten in dieser Kategorie die Open Source Office Produkte und Grafiksoftware (3,9 und 3,8). Mittlere Werte konnten Open Source Produkte von dem Bereich Cluster-Software bis hin zu ASP Lösungen erreichen (Messwerte zwischen 3,6 und 3,2). Als relativ wenig zuverlässig und sicher wurde freie Software der Bereiche Sprachapplikationen, Videosoftware, CRM/SRM und ERP Systeme gesehen (alle 2,6).

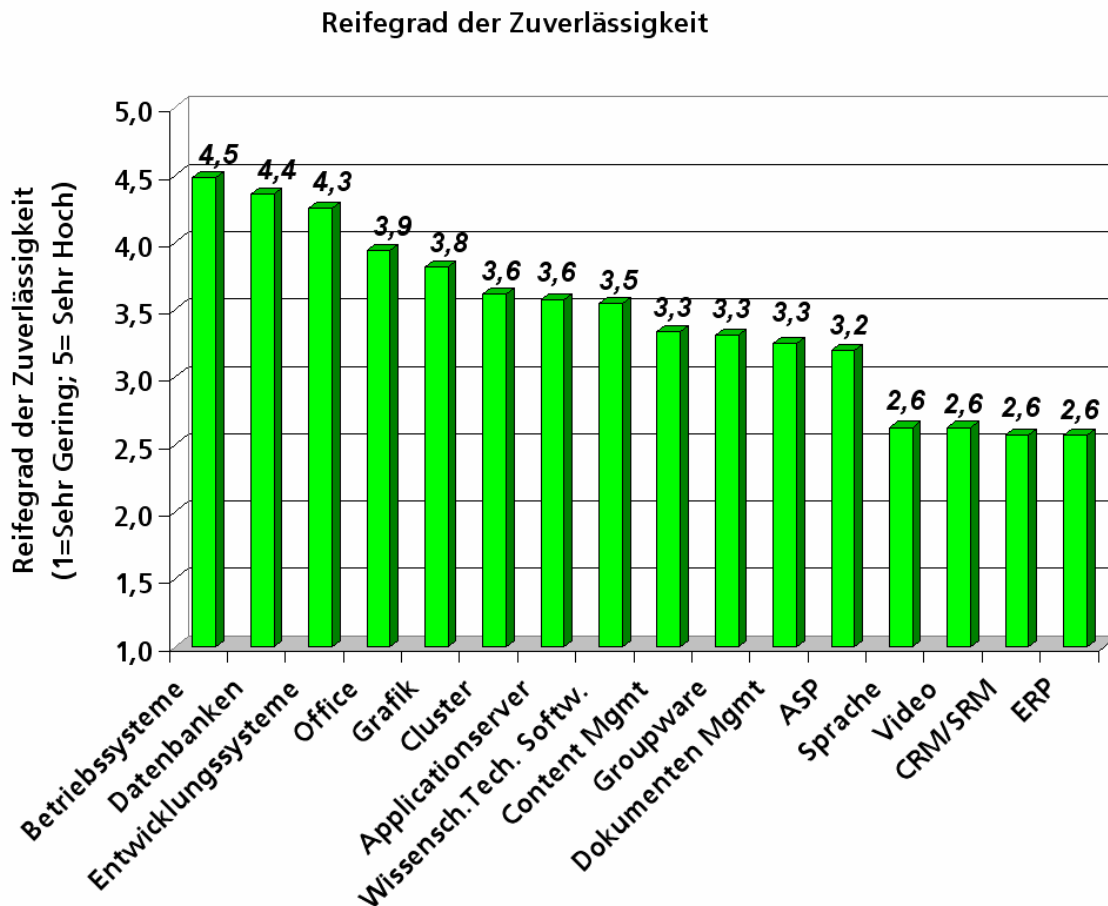


Abbildung 23: Reifegrad der Zuverlässigkeit

### Reifegrad der Leistungsfähigkeit

In diesem Bereich gehen die freien Betriebssysteme als klarer Sieger im Open Source Markt hervor. Gemessen an kommerzieller Software erreichte die Leistungsfähigkeit der freien Alternativen einen Messwert von 4,5. Des Weiteren wurde auch den freien Entwicklungssystemen, Clusterlösungen und Datenbanken eine überdurchschnittlich hohe Leistungsfähigkeit bescheinigt (4,1 bis 4,0). Gute Werte erreichten ferner die Grafik- und Wissenschaftlich-Technische Software (3,8 bzw. 3,7). Das Mittelfeld beginnt hier bei Office Produkten und endet bei Videosoftware – deren Leistungsfähigkeit erreichte Messwerte zwischen 3,5 und 3,1. Unterdurchschnittliche Leistungsfähigkeit wurden freien ERP Systemen, Sprachapplikationen und CRM/SRM Systemen attestiert.

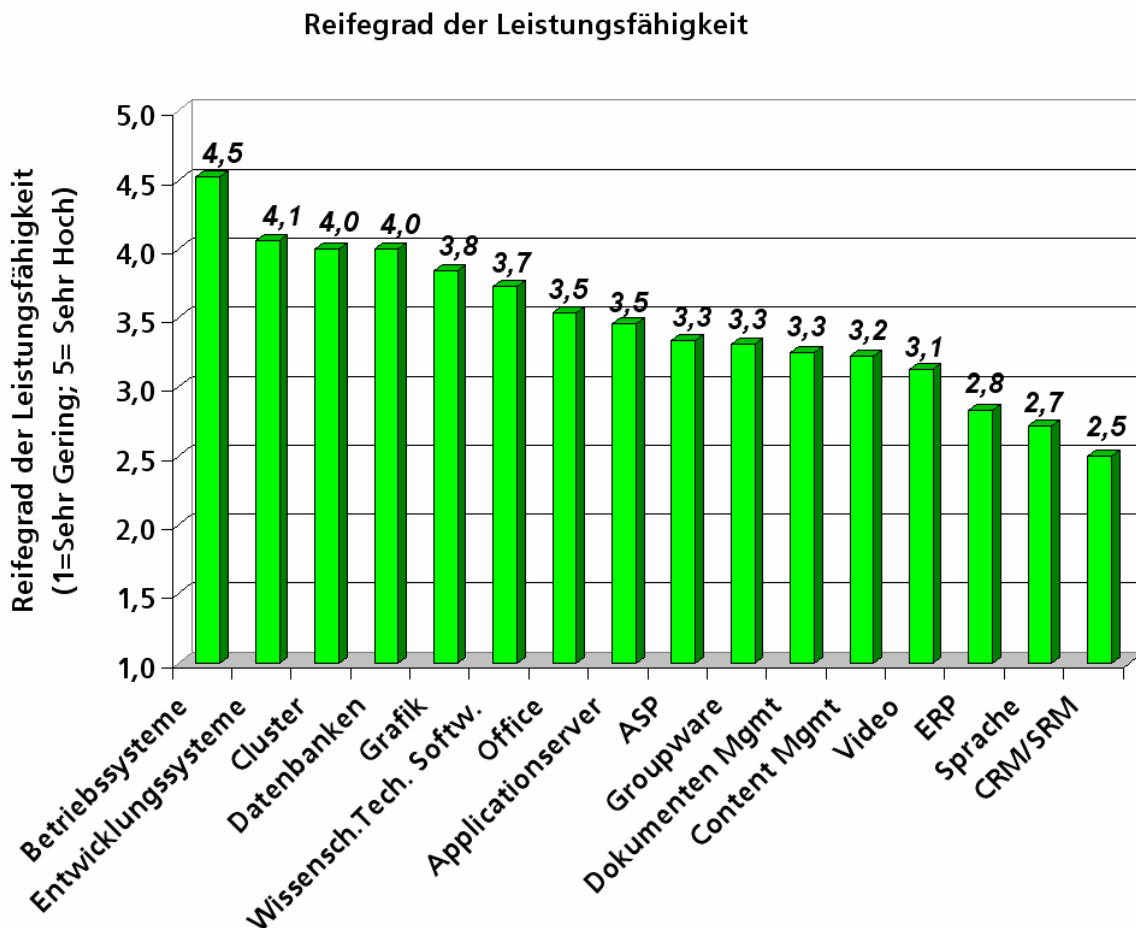


Abbildung 24: Reifegrad der Leistungsfähigkeit

### Reifegrad des Supports

Auch in der Frage des Supports haben die freien Betriebssysteme die Nase vorn. Die relative Güte des Supports wurde hier mit 3,9 bewertet. Noch guter Support ist für freie Datenbanken und Entwicklungssysteme erhältlich (Messwert jeweils 3,4). Das Mittelfeld reicht von Wissenschaftlich-Technischer Software bis zu Dokumenten Management Systemen (Werte zwischen 3,2 und 2,7). Mit der kommerziellen Konkurrenz nicht mithalten kann der Support freier Software in den Bereichen ERP, ASP, Content Management sowie CRM/SRM Systeme und Videosoftware (2,3 bis 2,1).

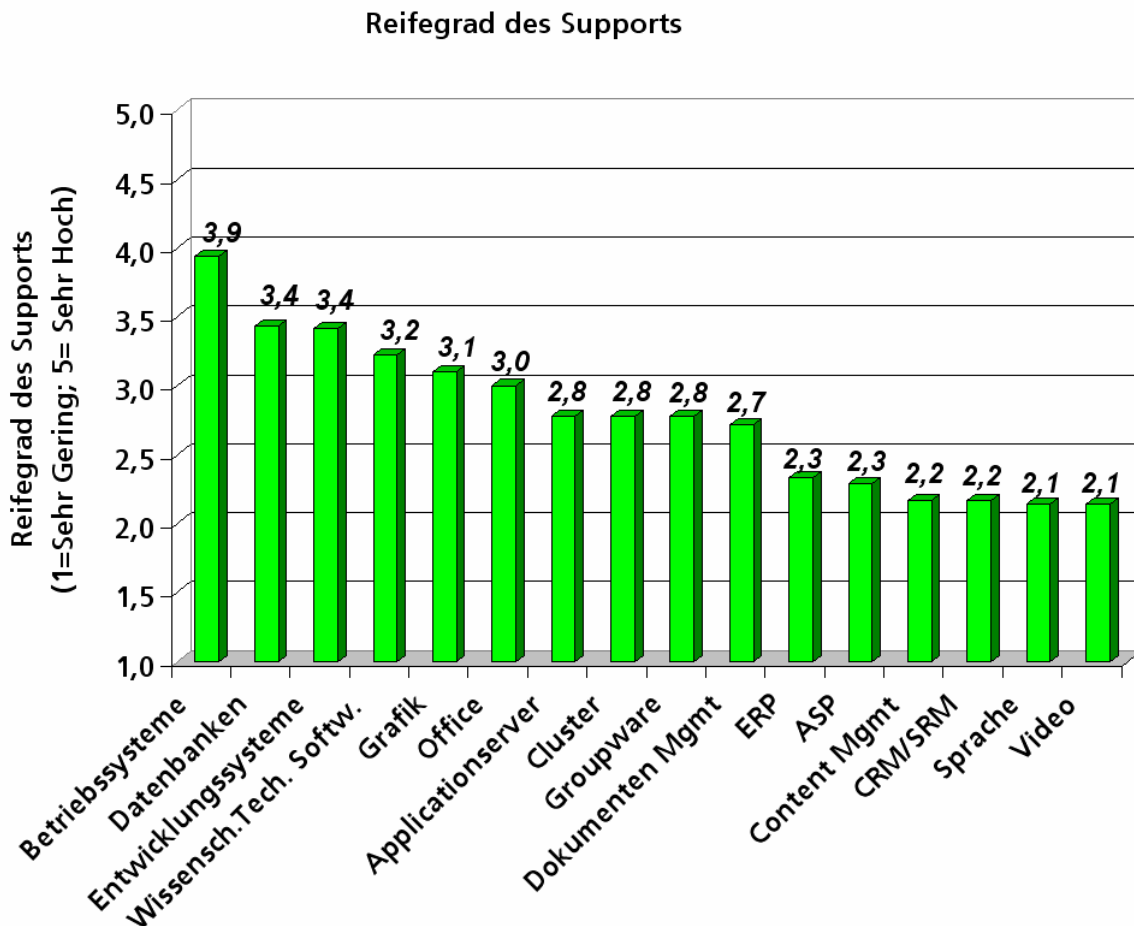


Abbildung 25: Reifegrad des Supports

### 3.5 Einsatz von Open Source Software in Kundenprojekten

In der Fraunhofer-Gesellschaft wurde Open Source Software bisher eher zurückhaltend in Kundenprojekten eingesetzt. Die Messwerte deuten hier auf eine durchweg unterdurchschnittliche Repräsentanz freier Software hin (alle unter 3,0). Am ehesten kamen freie Betriebssysteme, Datenbanken und Entwicklungssysteme zum Einsatz (2,9 bis 2,6). In anderen Bereichen konnte sich freie Software noch nicht wirklich in Kundenprojekten behaupten.

Einsatz von Open Source Software in Kundenprojekten

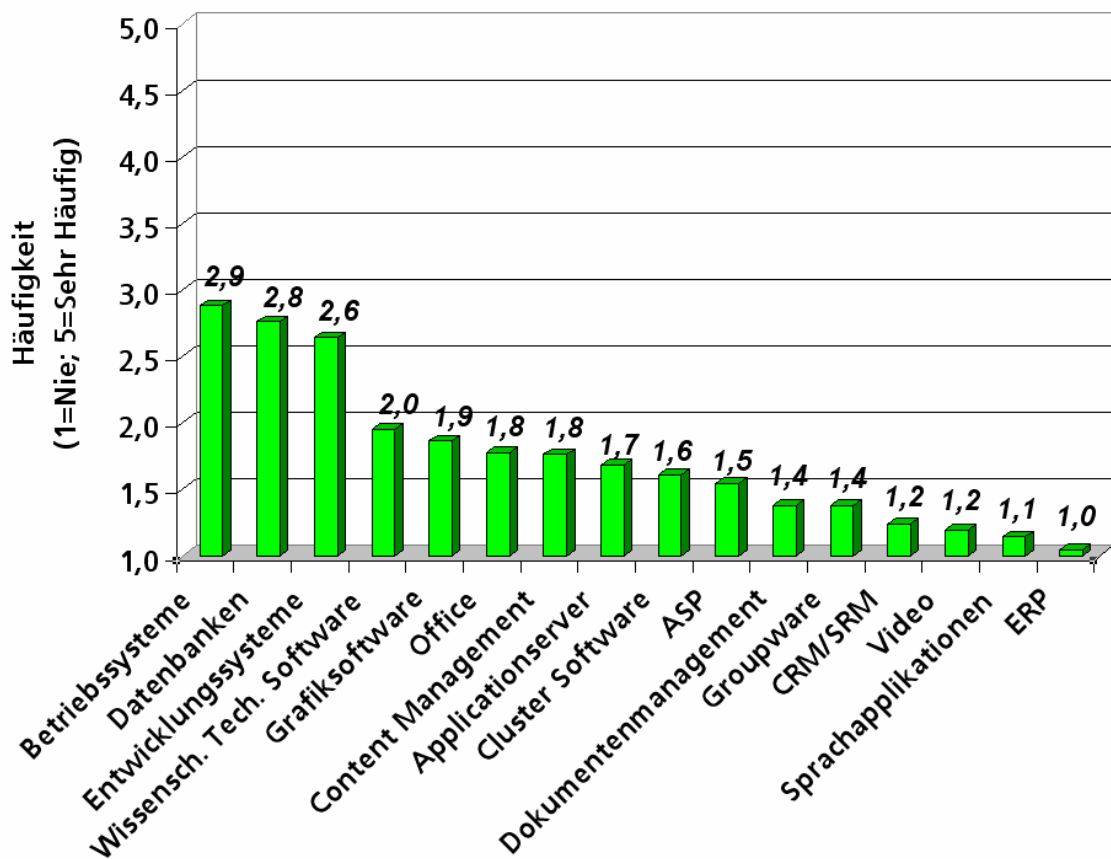


Abbildung 26: Einsatz von Open Source Software in Kundenprojekten

### 3.6 Eigenentwicklung von Open Source Software

Eigenentwicklung bzw. Weiterentwicklung (also die Veröffentlichung von Software unter einer Open Source Lizenz) spielte bisher in der Fraunhofer-Gesellschaft noch keine große Rolle. Wenn überhaupt, dann wurden am ehesten noch Betriebssystemkomponenten von der Fraunhofer-Gesellschaft als Open Source freigegeben. In allen anderen Bereichen kam dies so gut wie nie vor.

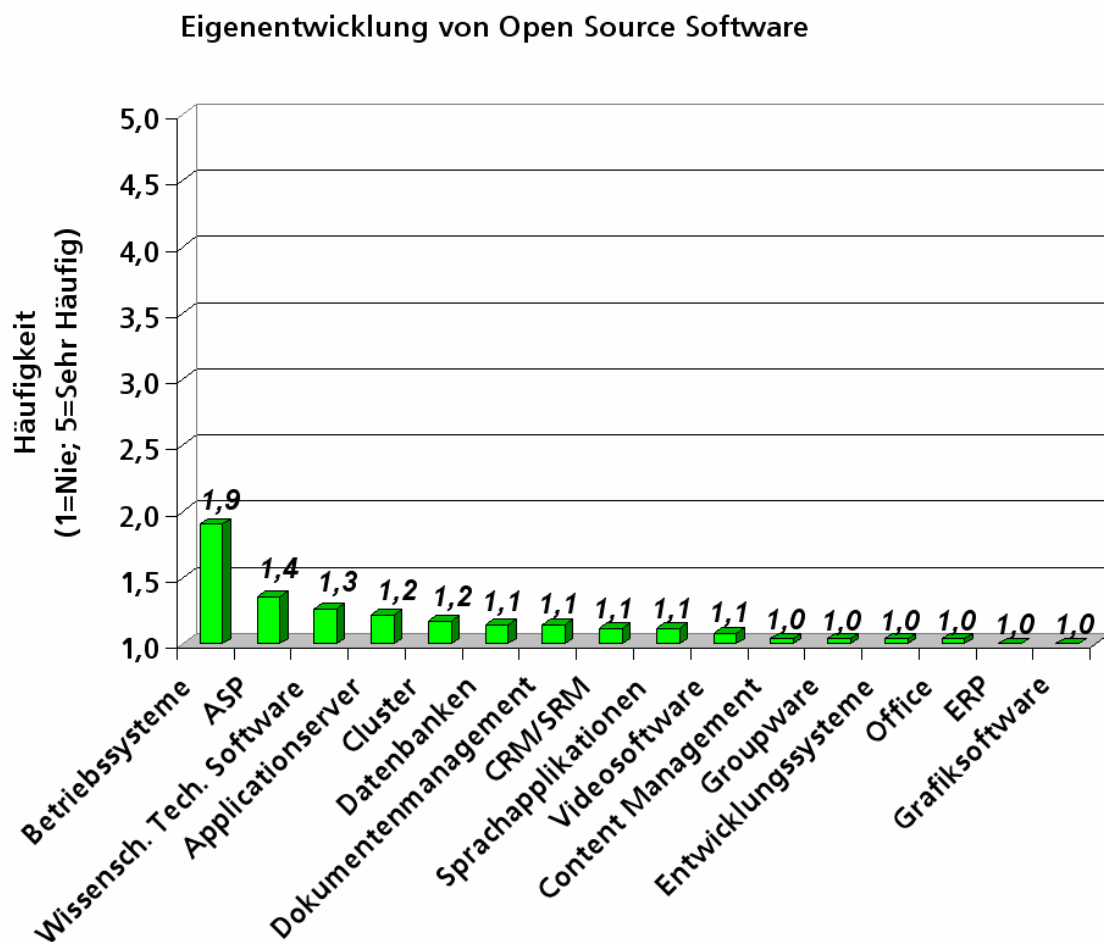


Abbildung 27: Eigenentwicklung von Open Source Software

Die in der Frage nach den Gründen für eine Eigenentwicklung von Open Source Software vorgegebenen Antwortmöglichkeiten fanden alle nur eine unterdurchschnittliche Zustimmung. Die Motivation zur Veröffentlichung von Open Source Software kann somit durchaus als begrenzt bezeichnet werden. Am ehesten versprochen sich die Befragten noch Vorteile durch die Einbeziehung der Community in den Entwicklungsprozess. Von Kunden oder Fördergebern wird der Wunsch nach einer Freigabe von Software unter einer Open Source Lizenz nur selten geäußert.

### Gründe für die Eigenentwicklung

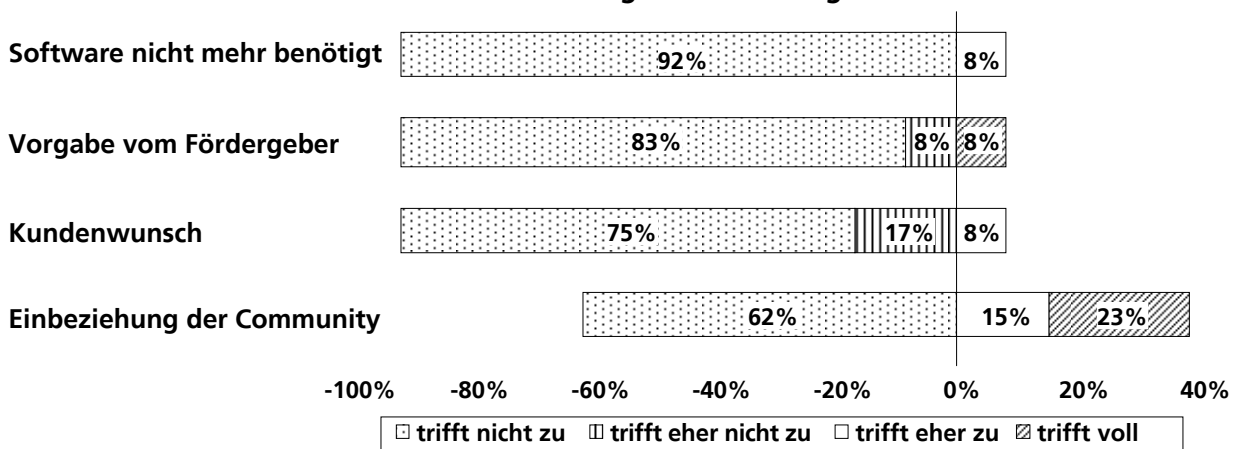


Abbildung 28: Gründe für die Eigenentwicklung

Für die Nicht-Freigabe von eigen entwickelter Software unter einer freien Lizenz ist vor allem der Wunsch nach einer kommerziellen Verwertung der Software verantwortlich – zusammen mit dem hohen Aufwand, der nach Einschätzung der Befragten für eine Veröffentlichung anfallen würde. Teilweise wird die Veröffentlichung auch vom Auftraggeber untersagt bzw. ist nicht gewünscht.

### Gründe gegen die Eigenentwicklung von Open Source Software

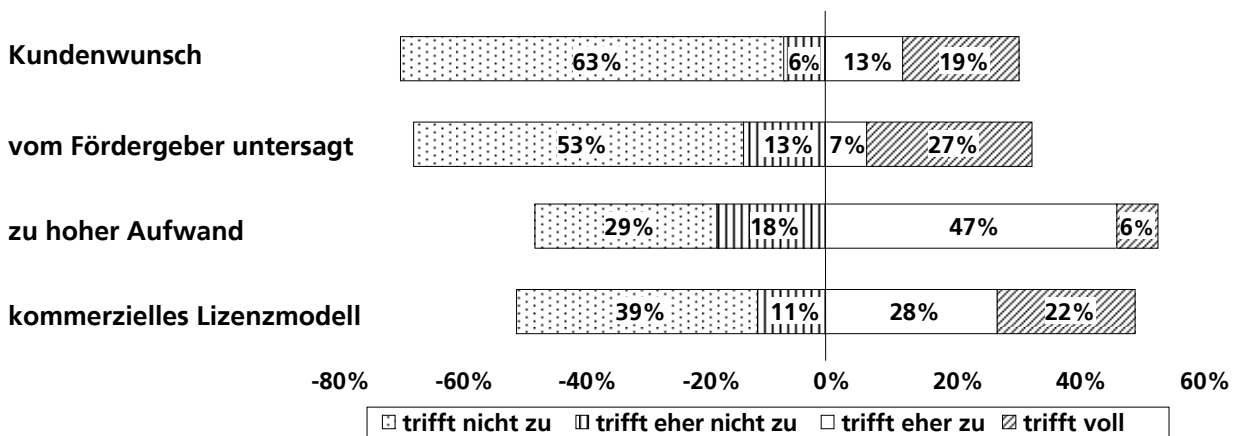


Abbildung 29: Gründe gegen die Entwicklung von Open Source Software

## Zusammenfassung

Open Source Software hat durchaus bereits einen festen Platz in der Fraunhofer-Gesellschaft gefunden. Allerdings kann von einer Ablösung von kommerzieller Software durch freie Alternativen bisher nicht gesprochen werden, da sehr viele Einsatzgebiete noch praktisch ausschließlich von kommerzieller Software dominiert werden.

Open Source Software ist insbesondere in den Gebieten

- Datenbanken
- Entwicklungssysteme
- Web Browser und –Server
- Content Management Systeme
- Cluster und Grid Software

und zu einem relativ geringeren Ausmaß in den Bereichen

- Betriebssysteme
- Office Produkte
- Grafik Software
- Wissenschaftlich-Technische Software

bereits präsent.

Die anderen Einsatzgebiete weisen bisher keine nennenswerte Durchdringung von Open Source Software auf, was teilweise sicherlich mit der mangelnden Ausgereiftheit bzw. Verfügbarkeit von freier Software in diesen Gebieten zu tun hat.

Die im Vergleich zu kommerzieller Software insgesamt größte Reife von Open Source Software sahen die Befragten in den Bereichen

- Betriebssysteme
- Entwicklungssysteme
- Datenbanken
- Office Produkte

Als noch relativ unreif betrachteten die Befragten Open Source Software insbesondere in den Bereichen

- Sprachverarbeitung
- Videosoftware
- ERP Systeme
- CRM/SRM Systeme

Ingesamt kann aus den Ergebnissen der Befragung geschlossen werden, dass die Grundhaltung gegenüber Open Source Software in der Fraunhofer-Gesellschaft eher positiv ist. Die Auswertung der Frage nach der Zustimmung zu Vorteilen bzw. Nachteilen von Open Source Software zeigte eine insgesamt höhere Zustimmung zu den Vorteilen von freier Software als dies bei den Nachteilen der Fall war. Die Vorteile scheinen somit insgesamt stärker ins Gewicht zu fallen.

Auch für die Fraunhofer-Gesellschaft ist der Wegfall der Lizenzkosten der bedeutendste Vorteil freier Software. Allerdings spielen auch strategische Kriterien wie die Verwendung von offenen Standards und die fehlende Abhängigkeit von bestimmten Softwareanbietern eine nicht zu verachtende Rolle. Überraschenderweise wurde die von vielen Open Source Befürwortern als Kernvorteil von freier Software genannte Anpassbarkeit und Wiederverwendbarkeit von Software als nicht sehr bedeutsam wahrgenommen.

Bei den Nachteilen freier Software fällt vor allem die hohe Zustimmung zum Kriterium der fehlenden Gewährleistung bzw. Haftung auf – vor allem da die Gewährleistung und Haftung auch von kommerziellen Anbietern sehr häufig beschränkt wird. Des Weiteren ist die ungewisse Weiterentwicklung von freier Software ein großes Problem aus der Sicht der Befragungsteilnehmer. Es waren nur relativ wenige Befragte der Ansicht, dass Open Source Software gemessen an kommerziellen Produkten nicht ausgereift ist.

Wenig wird Open Source Software derzeit in Kundenprojekten eingesetzt. Wenn, dann kommen vorwiegend freie Betriebssysteme, Datenbanken und Entwicklungswerkzeuge zum Einsatz. Die Freigabe von eigen entwickelter Software unter einer Open Source Lizenz ist dagegen höchst selten. In vielen Fällen wird mit selbst entwickelter Software doch eine kommerzielle Verwertungsabsicht verfolgt bzw. der Aufwand wird durch die Urheber als zu hoch wahrgenommen.

### **3.7 Ergebnisse einer Befragung bei deutschen Unternehmen**

Im Rahmen des von Wegweiser und Fraunhofer IAO erstellten eBusiness Konjunkturbarometers wurden deutsche Unternehmen unter anderem auch zu ihrer Haltung zu Open Source Software befragt.

Auch die Unternehmen sahen als wesentlichen Vorteil von Open Source Software den Wegfall von Lizenzkosten. Aber auch die nicht vorhandene Herstellerabhängigkeit wird als gewichtiger Vorteil betrachtet. Geringere, aber noch mehrheitliche Zustimmung finden die Kriterien der Stabilität von Open Source Software und die Anpassbarkeit von freier Software an eigene Erfordernisse.

Als Nachteile werden vor allem die ungewisse Weiterentwicklung und das Fehlen von professionellem Support. Auch fehlen nach Ansicht der Befragten

noch benötigte Applikationen. Schließlich betrachtete eine Mehrheit der Befragten auch die hohen Schulungskosten und die nicht gegebene Interoperabilität freier Software mit vorhandenen Systemen als Nachteil.

### Vorteile durch Open Source Einsatz

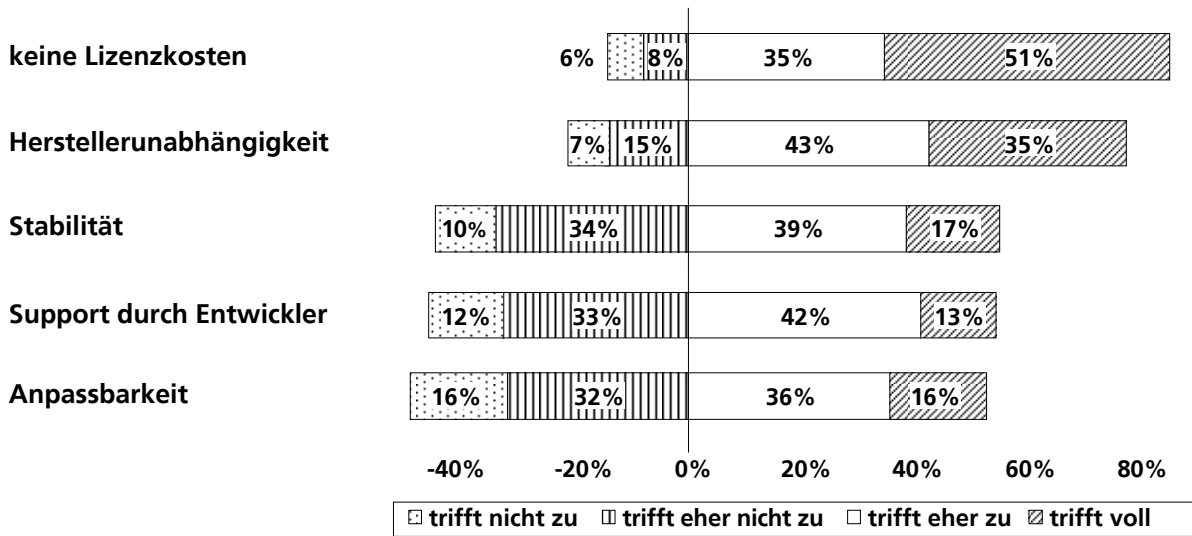


Abbildung 30: Vorteile durch Open Source Einsatz

### Nachteile durch Open Source Einsatz

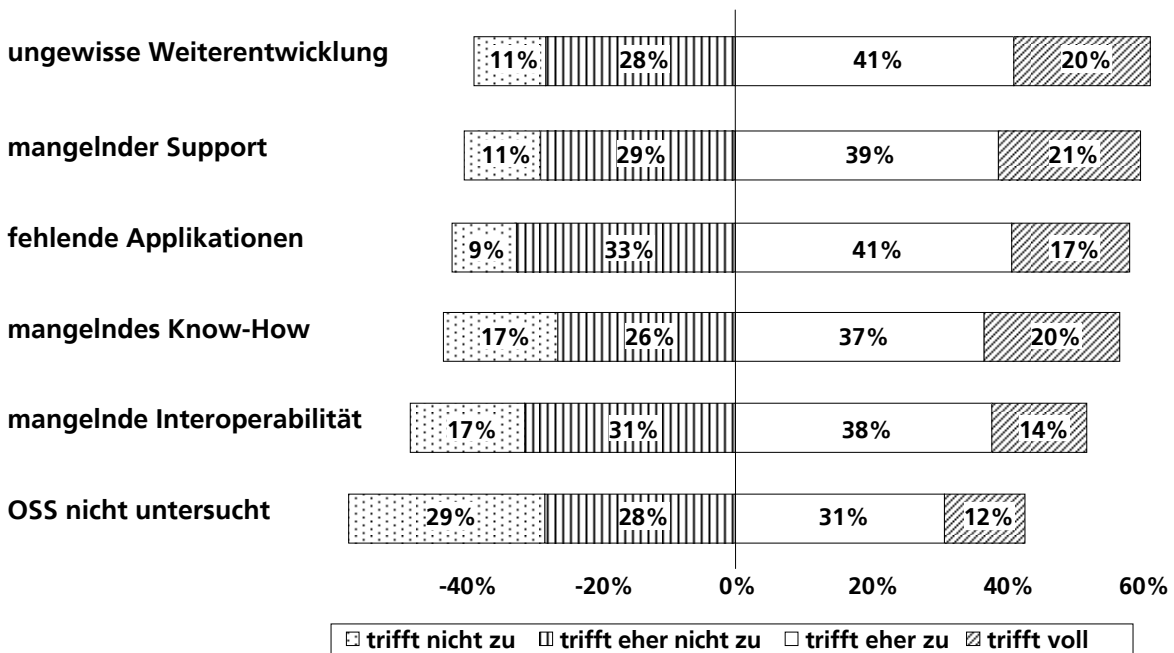


Abbildung 31: Nachteile von Open Source

## 4 Anwendungsfelder

In diesem Abschnitt werden Softwareprodukte aus verschiedenen Anwendungsbereichen jeweils kurz vorgestellt und gegenübergestellt. Ziel ist es, herauszuarbeiten, in welchen Einsatzgebieten und für welche Einsatzzwecke Open Source Software in Betracht gezogen sollte und in welchen Gebieten dies eher noch nicht der Fall ist. Dazu werden in den jeweiligen Einsatzgebieten die wichtigsten (bzw. eine Auswahl der wichtigsten) Softwareprodukte miteinander verglichen. Dabei wird sowohl kommerzielle als auch freie Software unter Verwendung eines für das jeweilige Einsatzgebiet einheitlichen Kriterienrasters gegenübergestellt. Kerngerüst für das Kriterienraster stellen wiederum die fünf Indikatoren für Softwarequalität:

- Functionality (Funktionalität),
- Usability (Benutzerfreundlichkeit),
- Reliability (Zuverlässigkeit),
- Performance (Leistung) und
- Support

Auf Softwareprodukte, die in der Darstellung der Befragung als eher relevant eingestuft wurden, wird dabei in größerer Detailtiefe eingegangen. Für diese werden die jeweils relevanten Unterkriterien der fünf Indikatoren abgeleitet und die Software anhand dieser verglichen. Für die als weniger relevant eingestuften Softwarearten wird auf die detaillierte Ableitung der Unterkriterien verzichtet.

### 4.1 Einführung und Überblick

Wie bereits in der Darstellung der Ist Analyse erläutert, wurde der Softwaremarkt in verschiedene Softwarearten unterteilt. Die Ist Analyse ergab dabei folgende Reihenfolge in der die Relevanz dieser Softwarearten bewertet wurde:

- 1 Betriebssysteme
- 2 Office Produkte
- 3 Wissenschaftlich-Technische Software
- 4 Groupware
- 5 Datenbanksysteme
- 6 Entwicklungssysteme
- 7 Grafiksoftware
- 8 Application Server / Portalsoftware
- 9 Dokumenten Management Systeme

- 10 Application Service Providing
- 11 Content Management Systeme
- 12 Clusterlösungen / Grid Computing
- 13 Customer/Supplier Relationship Management
- 14 Videosoftware
- 15 Enterprise Resource Planning Software
- 16 Sprachverarbeitung

Den Bereichen Betriebssysteme, Office Software, Wissenschaftlich-Technische Software, Groupware, Datenbanken sowie Entwicklungssysteme wird im Folgenden eine besondere Aufmerksamkeit gewidmet. Im Fall der restlichen Einsatzgebiete wird eine generelle Darstellung der Position freier Software in den jeweiligen Bereichen gegeben und Trendempfehlungen ausgesprochen.

## 4.2 Betriebssysteme

Betriebssysteme sind die grundlegende Software zum Betrieb eines Computers. Wichtige Teilaufgaben eines Betriebssystems sind die Verwaltung von Ressourcen (z.B. Speicher, Rechenzeit), von angeschlossenen Peripheriegeräten (Bildschirm, Drucker), des Dateisystems sowie die Ausführung von Applikationen.

### 4.2.1 Merkmale von Betriebssystemen

#### Funktionalität

Betriebssysteme unterscheiden sich funktional insbesondere aufgrund folgender Kriterien:

- Mehrbenutzerunterstützung
- Netzwerkunterstützung
- Unterstützung von Peripheriegeräten
- Einzelprogramm/Mehrprogrammbetrieb
- Anzahl/Qualität von Dienstprogrammen

Für moderne Betriebssysteme ist die Mehrbenutzerunterstützung absolut notwendig. Dazu gehört insbesondere eine Benutzerverwaltung sowie die damit zusammenhängenden Sicherheitsmechanismen (Authentifikation, Zugriffskontrolle usw.).

Da die wenigsten Rechner in einer Organisation isoliert betrieben werden, muss das Betriebssystem den Betrieb in einer Client/Server Umgebung bzw. Arbeitsgruppenorganisation unterstützen. Hierzu gehören insbesondere Netzwerkprotokolle und die gemeinsame Nutzung von Ressourcen (wie z.B. Drucker). Die Eignung des Betriebssystems als File- oder Druckserver fällt ebenfalls unter dieses Kriterium.

Das Betriebssystem sollte ferner alle gängigen Peripheriegeräte unterstützen. Dies wird insbesondere durch die Art und Anzahl der zu Verfügung stehenden Treiber bestimmt.

Für moderne Arbeitsabläufe ist es notwendig, dass das Betriebssystem den gleichzeitigen Ablauf mehrerer Applikationen gestattet. Diese Forderung richtet sich nicht nur an Server, sondern auch an Clients (der gleichzeitige Betrieb mehrerer Clientapplikationen ist heutzutage absolut üblich).

### **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit eines Betriebssystems kann in den administrativen Bereich und den Anwendungsbereich unterteilt werden. Unter den administrativen Bereich fallen Aufgaben, die mit der Einrichtung, Wartung und Systemverwaltung zu tun haben. Dazu gehören beispielsweise

- Einrichtung von Benutzern und Benutzerrechten
- Einrichtung von Hardware und Peripherie
- Verwaltung des Dateisystems
- Konfiguration von Netzwerkdiensten

Anwender kommen mit Betriebssystemfunktionalitäten vor allem über die grafische Benutzerschnittstelle oder eine Kommandozeileneingabe in Berührung. Diese sollte sich möglichst intuitiv bedienen lassen und eine geringe Einarbeitungszeit erfordern.

### **Zuverlässigkeit / Sicherheit**

Unter Zuverlässigkeit fallen bei Betriebssystemen insbesondere die Stabilität bzw. Ausfallsicherheit. Das Kriterium der Sicherheit bestimmt sich wiederum über mehrere einzelne Anforderungen.

Eine grundlegende Forderung für Betriebssystemsicherheit ist, dass nur berechtigte Benutzer Zugang zum System und dessen Anwendungen sowie Daten haben dürfen. Dies wird im Betriebssystem über Zugriffskontrolle und Authentifikationsmechanismen bewerkstelligt. Diese stellen auch die Vertraulichkeit von den im Dateisystem abgelegten Daten sicher.

Eine weitere Forderung ist der Schutz des Systems vor Sabotage. Dies beinhaltet eine Resistenz gegenüber böswilligen oder irrtümlichen Angriffsversuchen und Schadprogrammen (wie Viren und Würmer).

Die Forderung nach Datenintegrität (im Sinne von einem Schutz der im Dateisystem gehaltenen Daten vor Veränderung oder Verlust) wird in vielen Fällen durch Datensicherungsprogramme (Backup) sichergestellt. Aber auch die

Fähigkeit des Dateisystems, Fehler zu vermeiden oder fehlerhafte Dateien als solche zu erkennen und ggf. zu reparieren, fallen unter diese Anforderung.

### **Leistungsfähigkeit**

Die Leistungsfähigkeit eines Betriebssystems wird in erster Linie durch die Ablaufgeschwindigkeit seiner Prozesse und die Skalierbarkeit definiert.

### **Support**

Das Kriterium des Supports misst vor allem, wie gut ein Betriebssystem durch Dienstleister unterstützt wird. Im Rahmen dieser Studie soll auch Art und Umfang des am Markt verfügbaren Know-hows an dieser Stelle mit einbezogen werden. Ein wesentliches Kriterium für die Einsatzfähigkeit eines Betriebssystems ist auch die Unterstützung durch Applikationen. Betrachtet werden soll daher auch die Art und Menge der zur Verfügung stehenden Software für das jeweilige Betriebssystem.

## **4.2.2 Gegenüberstellung von Betriebssystemen**

### **Microsoft Windows**

Windows ist das meist eingesetzte PC Betriebssystem weltweit. Der Marktanteil von Microsoft Windows auf Clientsystemen beträgt rund 95%<sup>34</sup>. Auf der Serverseite ist die Microsoft-Vorherrschaft allerdings nicht ganz so eindeutig. Rund 55% aller Server laufen unter Windows<sup>35</sup>.

Windows ging ursprünglich als grafische Benutzeroberfläche für MS-DOS hervor, ist aber seit der Version Windows 95 ein eigenständiges Betriebssystem. In den aktuellen Versionen Windows XP (Client) sowie Windows Server 2003 (Server) unterstützt Windows sowohl den Mehrbenutzer- als auch den Mehrprogrammbetrieb.

In den Client- bzw. Serverversionen von Windows sind sämtliche Funktionalitäten für den Netzwerkbetrieb in einer großen Organisation enthalten. Mit Active Directory steht ein mächtiger Verzeichnisdienst für die Benutzerverwaltung zur Verfügung. Das NTFS Dateisystem genügt den Anforderungen für den Einsatz in einer großen Organisation ebenfalls in vollem Umfang. Des Weiteren stellt die Servervariante von Windows die notwendigen Dienstprogramme für Druckdienste, DNS und DHCP zur Verfügung. Auch ein

<sup>34</sup> Vgl. z.B. <http://www.webhits.de/deutsch/index.shtml?webstats.html>

<sup>35</sup> Vgl. <http://www.golem.de/0310/27815.html>

Betrieb als Webserver ist mit Windows 2003 möglich – entsprechende Dienstprogramme werden mitgeliefert. Die Clientversion von Windows (XP) ist nicht für den Einsatz als Server vorgesehen und enthält daher auch die dafür notwendigen Dienstprogramme nicht.

Bedingt durch seine Marktführerschaft ist die Unterstützung von Peripheriegeräten in Windows hervorragend – es gibt kaum Hardware, die von Windows nicht unterstützt wird.

Die Benutzerfreundlichkeit von Windows ist generell gut. Insbesondere im Anwendungsbereich kann es von den meisten Nutzern intuitiv und komfortabel bedient werden. Im administrativen Bereich fällt die zum Teil sehr hohe Komplexität der Administrationsmodule auf. Konfigurationen können zwar zumeist über grafische Oberflächen vorgenommen werden, deren Bedienung ist allerdings aufgrund der Vielzahl der Einstellmöglichkeiten eher schwierig und teilweise kontraintuitiv. Im schlimmsten Fall können Einstellungen nicht über das Administrationsinterface vorgenommen werden, sondern müssen über das manuelle Setzen von Werten in der Registrierdatenbank erfolgen.

Die Zuverlässigkeit galt lange Zeit als eine der Schwachstellen von Windows. Was die Systemstabilität anbetrifft, so muss aber zugunsten von Windows festgestellt werden, dass die von permanenten Systemabstürzen gezeichneten Zeiten von Windows 95 vorbei sind. Sowohl Windows 2003 Server als auch Windows XP laufen im täglichen Betrieb sehr stabil. Schwierigkeiten entstehen manchmal durch das als nicht wirklich durchdacht erscheinende Konzept der Registrierdatenbank, in der zentral System- und Applikationseinstellungen abgelegt sind. Diese kann im Alltagsbetrieb relativ leicht beschädigt werden, da viele Prozesse Schreibrechte auf diese Datenbank benötigen. Dies führt dann möglicherweise zu nicht mehr nutzbaren Installationen.

Gegen Schadprogramme wie Viren und Würmer ist Windows aber auch heute noch sehr anfällig. Dies ist zwar teilweise mit der auf die hohe Verbreitung dieses Betriebssystems zurückzuführende Attraktivität von Windows für Angreifer zu erklären – aber nicht vollständig. Die Systemarchitektur von Windows macht den Programmierern von Schadsoftware das Leben auch unnötig einfach. Zu viele Aufgaben werden unter Windows mit Systemverwalterrechten ausgeführt, die zugriffsrechtliche Trennung zwischen Administratoren und normalen Benutzern ist weit weniger konsequent umgesetzt, als dies bei anderen Systemen der Fall ist.

Die Leistungsfähigkeit von Windows ist als hoch zu bezeichnen. Bei verschiedenen Benchmarks schnitt Windows im Vergleich mit anderen Betriebssystemen stets gut ab. Die Ergebnisse der Benchmarks scheinen

allerdings teilweise vom jeweiligen Auftraggeber abzuhängen<sup>36</sup>, so dass diese Vergleichszahlen mit großer Vorsicht zu genießen sind.

Das Kriterium Support wird von Windows in außerordentlich hohem Umfang erfüllt. Eine große Anzahl Unternehmen bietet Dienstleistungen rund um Windows an. Windows Know-how ist in den meisten Organisationen vorhanden und ggf. auf dem Arbeitsmarkt leicht zu beschaffen. Es stehen nahezu alle benötigten Applikationen für Windows zur Verfügung - die Unterstützung durch Softwareanbieter ist für dieses Betriebssystem deutlich höher als für jedes andere.

## Linux

Linux ist ein freies UNIX Betriebssystem<sup>37</sup>. Die Entwicklungsarbeiten an Linux begannen im Jahr 1991, damals als Hobbyprojekt von Linus Torvalds. Die Version 1.0 des Linux Kernels wurde 1994 freigegeben. Linux ist seit 1992 Teil des GNU Projektes und unter der GPL frei verfügbar. Die aktuelle Version des Kernels ist 2.6.

Aufgrund der freien Verfügbarkeit von Linux ist die Abschätzung seines Marktanteils entsprechend schwierig. Schätzungen gehen von einem Marktanteil von ca. 3-4% auf Clients und rund 25% auf Servern aus<sup>38</sup>. Linux kann derzeit in Bezug auf den Marktanteil auch stark zulegen. Man geht von einer Verdopplung des Marktanteils auf dem Client in den nächsten Jahren und von einem weiteren starken Wachstum im Serverbereich aus.

Linux kann sowohl im Mehrbenutzer- als auch im Mehrprogrammbetrieb eingesetzt werden. Das Betriebssystem verfügt über eine weitreichende Unterstützung des Netzwerkbetriebs. Alle gängigen Linuxdistributionen enthalten Dienstprogramme, die es erlauben Linux als Fileserver, Webserver oder gar als Datenbankserver zu betreiben. Als Verzeichnisdienst wird unter Linux üblicherweise LDAP eingesetzt, DNS und DHCP Module stehen ebenfalls zur Verfügung. Die Rechteverwaltung unter Linux ist einfacher gehalten als unter Windows, ist aber deswegen letztendlich nicht weniger mächtig. Sämtliche benötigten Zugriffsrechte können mit Linux gesetzt und verwaltet werden. Die Rechteverwaltung von Linux zwingt den Administrator lediglich zu einer stringenten Konzeption – was sicherlich nicht als Nachteil zu betrachten ist.

<sup>36</sup> Siehe dazu auch [http://searchwin2000.techtarget.com/originalContent/0,289142,sid1\\_gci1027603,00.html](http://searchwin2000.techtarget.com/originalContent/0,289142,sid1_gci1027603,00.html)

<sup>37</sup> Im allgemeinen Sprachgebrauch wird mit dem Begriff Linux meistens eine vollständige Linuxdistribution referenziert. Eigentlich ist Linux aber lediglich der unter <http://www.kernel.org/> bereitgestellte Betriebssystemkern. Bekannte Linuxdistributionen sind SuSE (<http://www.suse.de>), Debian (<http://www.debian.org>), oder Fedora Core (<http://www.fedora.redhat.com>). Diese unterscheiden sich vorwiegend durch die Art und Anzahl der mitgelieferten Software sowie den bereitgestellten Installations- und Verwaltungstools.

<sup>38</sup> Vgl. <http://www.golem.de/0310/27815.html>

Für Linux stehen verschiedene Dateisysteme zur Auswahl. Die beiden wohl am häufigsten eingesetzten dürften ext3 und ReiserFS sein. Beide Dateisysteme sind für den Einsatz auf einem Fileserver geeignet.

Die Unterstützung von Hardware ist in Linux hinreichend, aber in manchen Bereichen verbesserungswürdig. Während es mit den meisten üblichen Peripheriegeräten (wie Drucker, Scanner, USB Sticks etc.) kaum noch Probleme gibt, kann es dennoch im Einzelfall zu Schwierigkeiten kommen. Dies ist insbesondere der Fall, wenn Hardwareanbieter keine Treiber für Linux anbieten und gleichzeitig ihre Spezifikationen nicht offen legen, so dass auch der Open Source Community die Entwicklung eines Treibers nicht möglich ist. Dies ist aufgrund der noch recht geringen Verbreitung von Linux durchaus oft, aber mit abnehmender Tendenz zu beobachten.

In Bezug auf die Benutzerfreundlichkeit für den Anwender steht eine moderne Linux Distribution dem von Windows gewohnten Komfort kaum noch nach. Mit KDE und GNOME stehen gleich zwei ausgereifte grafische Benutzerschnittstellen für die Anwender zur Auswahl<sup>39</sup>. Fortgeschrittene Anwender werden auch die im Vergleich zu Windows deutlich komfortablere Kommandozeileingabe (Shell) zu schätzen wissen.

Die Administration eines Linux Systems unterscheidet sich jedoch gravierend von einem Windows-basierten System. Die meisten Einstellungen werden unter Linux nach wie vor in textbasierten Konfigurationsdateien vorgenommen. Dies muss kein Nachteil sein, da ein umständliches Navigieren in einer grafischen Administrationskonsole entfällt. Auch gehen in letzter Zeit immer mehr Anbieter von Linux Distributionen dazu über, grafische Schnittstellen für die Administration bereitzustellen, so dass die Administratoren zunehmend wählen können, ob sie lieber textbasiert oder grafisch arbeiten wollen.

Die Zuverlässigkeit ist eine der klaren Stärken von Linux. Die Stabilität des Linux Kernels ist herausragend, Systemabstürze sind unter diesem Betriebssystem eine absolute Ausnahme. Die Bedrohung durch Viren und Würmer ist minimal. Auch wenn die Aufmerksamkeit der Schadsoftwareprogrammierer durch seine steigende Verbreitung zukünftig auch vermehrt auf Linux gerichtet werden könnte, so bietet Linux doch einige entscheidende Vorteile durch seine Architektur. Da ein normaler Benutzer unter Linux ohne Systemrechte arbeiten kann, ist der mögliche Schaden etwa durch Viren auch lokal begrenzt.

<sup>39</sup> Es gibt noch zahlreiche weitere Fenstermanager für Linux, die aber überwiegend von fortgeschrittenen Anwendern genutzt werden, die mehr Wert auf Effizienz als auf Komfort legen.

Linux ist ein sehr leistungsfähiges Betriebssystem, das sich auch für den Betrieb in einer Hochleistungsumgebung eignet. Wie für Windows gilt aber auch hier, dass die existierenden Benchmarks teilweise mit Vorsicht zu genießen sind.

Was den Support anbetrifft, so gibt es mittlerweile eine ganze Reihe von Dienstleistern, die Support für Linux anbieten. Linux Know-how ist allerdings nicht ganz so verbreitet, wie dies für Windows der Fall ist. Insbesondere bei Anwendern sind Linux Kenntnisse eher rar. Die Verfügbarkeit von Applikationen ist in manchen Bereichen sehr gut und in manchen Bereichen mangelhaft. Hervorragende Unterstützung genießt Linux insbesondere in den Bereichen Serversoftware (z.B. Datenbanken, Webserver, Application Server, Web Content Management etc.), Entwicklungssysteme und Office Software. Manche Softwarearten sind für Linux kaum bzw. nicht in hinreichender Qualität zu beschaffen, z.B. Videoschnittsoftware. Die Verfügbarkeit von Spezial- oder Branchensoftware ist ebenfalls nicht in allen Fällen gewährleistet. Das Softwareangebot kann insgesamt nicht mit Windows mithalten, wird aber zunehmend besser.

## **BSD**

Die Berkeley Software Distribution (BSD) ist ebenfalls ein UNIX Betriebssystem, das von der Universität Berkeley seit den siebziger Jahren entwickelt und später als Open Source Software unter der gleichnamigen Lizenz freigegeben wurde. Heutzutage werden vor allem die von BSD abgeleiteten Systeme FreeBSD<sup>40</sup> und OpenBSD<sup>41</sup> eingesetzt. Die Eigenschaften von BSD sind denen von Linux sehr ähnlich und brauchen daher an dieser Stelle nicht noch einmal detailliert aufgeführt zu werden. FreeBSD gilt als eines der stabilsten und leistungsfähigsten Betriebssysteme überhaupt. OpenBSD wurde vor allem mit dem Fokus Sicherheit entwickelt. BSD wird aufgrund dieser Eigenschaften besonders gerne für Webserver und Firewalls verwendet.

In Bezug auf den Support kann BSD allerdings weder mit Linux noch mit Windows mithalten. Dies gilt auch für das Angebot an verfügbaren Applikationen.

## **Mac OS X**

Mac OS X ist die neueste Version des Mac OS Betriebssystems für Apple Macintosh Computer. Es ist sowohl in einer Client als auch in einer Servervariante erhältlich. Mac OS X wurde teilweise von dem UNIX Betriebssystem BSD abgeleitet, so dass die dort angeführten Feststellungen

<sup>40</sup> <http://www.freebsd.org/>

<sup>41</sup> <http://www.openbsd.org/>

bezüglich Funktionalität und Leistungsfähigkeit weitgehend auch für Mac OS X gelten. Auch weist Mac OS X die von einem UNIX System zu erwartende hohe Zuverlässigkeit auf. Die an sich in einem UNIX System vorhandene hohe System-sicherheit wurde an manchen Stellen aber nicht ganz konsequent in die Benutzerschnittstelle fortgeführt, so dass Mac OS X hier nicht so hoch zu bewerten ist wie andere UNIX Derivate. Mac OS X scheint insgesamt eher für den Einbenutzerbetrieb ausgelegt zu sein.

Das grafische Benutzerinterface Aqua wurde von Apple speziell für Mac OS X entwickelt. Bei vielen Anwendern gilt Aqua als die benutzerfreundlichste Oberfläche überhaupt.

Das Softwareangebot ist in manchen Bereichen sehr gut. Macintosh Computer sind insbesondere in der Design Branche sehr stark verbreitet, was zu einem überragenden Angebot an Grafik- und Videosoftware geführt hat. Auch gängige Office Produkte sind für Mac OS X erhältlich. Etwas schwächer sieht es in manchen anderen Bereichen aus (z.B. Entwicklungstools).

Als einziges hier vorgestelltes Betriebssystem läuft Mac OS X nicht auf X86 basierter Hardware, so dass die Entscheidung für oder gegen einen Einsatz auch immer im Kontext der Hardware erfolgen muss.

### 4.2.3 Einsatzerfahrungen

Microsoft Windows ist das meisteingesetzte Betriebssystem. Es dürfte auf der überwiegenden Mehrheit aller Clients und auch auf vielen Servern zum Einsatz kommen. Allerdings kann in vielen Organisationen trotzdem nicht von einer reinen Microsoft Landschaft gesprochen werden

Auf sehr vielen Servern kommen alternative Betriebssysteme zum Einsatz, insbesondere Linux und Sun Solaris. Die Präsenz dieser Betriebssysteme als Web- und Netzwerkinfrastrukturserver ist nicht zu verachten.

Im Clientbereich dominiert hingegen Windows. Dies hängt wohl nicht zuletzt mit der Tatsache zusammen, dass manche benötigte Applikationen für alternative Betriebssysteme nicht bzw. nicht in der erforderlichen Qualität erhältlich sind. Auch mag es wohl nicht zuletzt an der Entscheidung vieler Organisationen liegen, Microsoft Office einzusetzen. Diese Office Suite ist für freie Betriebssysteme nicht erhältlich und kann auf diesen Systemen nur über Emulationssoftware zum Laufen gebracht werden.

Der Reifegrad von Open Source Betriebssystemen wurde im Rahmen der Erhebung relativ zu relevanten kommerziellen Softwareprodukten und ebenfalls auf einer fünf-stufigen Skala bewertet (auch hier wurde mit 5 der höchstmögliche und mit 1 der niederstmögliche Wert kodiert). Der Reifegrad

wurde separat für die bereits mehrfach erwähnten fünf Indikatoren für Softwarequalität erhoben:

- Funktionalität: 4,4
- Benutzerfreundlichkeit: 3,9
- Zuverlässigkeit, Sicherheit: 4,5
- Leistungsfähigkeit: 4,5
- Support: 3,9

Damit liegen die freien Betriebssysteme unter den bezüglich des Reifegrades als am höchsten bewerteten Open Source Softwarekategorien. Insbesondere bei der Funktionalität und bei der Leistungsfähigkeit konnten die freien Betriebssysteme den mit Abstand höchsten Wert aller betrachteten Softwarearten verbuchen. Auch bei den Kriterien Support und Zuverlässigkeit erzielten die Open Source Betriebssysteme den besten Wert.

#### 4.2.4 Bewertung und Empfehlung

Die folgende Tabelle fasst die dargestellten Ausführungen zusammen.

Kriterium	Windows	Linux	BSD	Mac OS X
<b>Lizenz</b>	Kommerziell	OSS (GPL)	OSS (BSD)	Kommerziell
<b>Funktionalität</b>	●	●	◐	◐
Mehrbenutzerbetrieb	●	●	●	◐
Mehrprogrammbetrieb	●	●	●	●
Netzwerkunterstützung	●	●	●	●
Unterstützung von Peripheriegeräten	●	◐	○	◐
Dienstprogramme	◐	●	●	◐
<b>Benutzerfreundlichkeit</b>	◐	◐	◐	●
Anwender	◐	◐	◐	●
Administration	◐	◐	○	●
<b>Zuverlässigkeit und Sicherheit</b>	◐	●	●	◐
Systemstabilität	◐	●	●	●
Zugriffskontrolle	◐	●	●	◐
Schutz vor Angriffen	○	●	●	◐
Datenintegrität	●	●	●	●
<b>Leistungsfähigkeit</b>	●	●	●	●
Ablaufgeschwindigkeit	●	●	●	●
<b>Support</b>	●	◐	○	○
Unterstützung durch Dienstleister	●	◐	○	●
Vorhandenes Know-how	●	◐	○	○
Unterstützung durch Applikationen	●	◐	○	○

Tabelle 17: Gegenüberstellung von Betriebssystemen  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung)

Funktional liegen Microsoft Windows und Linux praktisch gleich auf. Linux hat seine Stärken vor allem im Bereich Zuverlässigkeit/Sicherheit und in der Art und Menge der mitgelieferten Dienstprogramme – hier wieder speziell in der Unterstützung von Netzwerk- und Serverfunktionalitäten. Windows kann insbesondere mit der hohen Anzahl an verfügbaren Applikationen, der Unterstützung von Peripheriegeräten sowie der Menge an vorhandenem Know-how punkten – genau diese Kriterien sprechen am meisten gegen BSD. Ein insgesamt sehr rundes Betriebssystem ist Mac OS X – wäre da nicht die Tatsache, dass es nur auf Apple Hardware läuft und somit nur eine recht geringe Verbreitung erreichen konnte. Allerdings weist auch Mac OS X einige Schwachstellen im Bereich Support auf.

Sowohl BSD als auch Mac OS X werden wohl nur in bestimmten Bereichen zum Einsatz kommen. BSD eignet sich vor allem als Webserver oder Firewall während Mac OS X seine Stärken im grafischen Bereich ausspielen kann. Mac OS X kann grundsätzlich auch in heterogenen Umgebungen betrieben werden (etwa als Client in einem Windows Netzwerk). Allerdings muss dafür entsprechendes Know-how sowohl bei Anwendern als auch Administratoren vorhanden sein – gerade dieses dürfte in vielen Bereichen außerhalb der Design-Branche eher rar sein. Vor diesem Hintergrund kann ein Einsatz von Mac OS X und BSD nur in Ausnahmefällen empfohlen werden.

Sowohl Linux als auch Windows erfüllen grundsätzlich alle Anforderungen an ein modernes Betriebssystem. Eine allgemeingültige Empfehlung für den Einsatz von Windows oder Linux kann an dieser Stelle aber trotzdem nicht gegeben werden, da ein großer Teil der Entscheidung vom jeweiligen Einsatzbereich und hier speziell von der Verfügbarkeit von benötigter Software abhängt. Praktisch uneingeschränkt kann Linux als Serverbetriebssystem empfohlen werden. Insbesondere als Projektserver, aber auch für Web-, Netzwerkinfrastruktur- und Datenbankserver gibt es keinen Grund, Linux nicht zum Einsatz zu bringen. Sofern nicht zwingend Active Directory als Verzeichnisdienst verwendet werden soll<sup>42</sup>, spricht auch nichts gegen einen Linux Einsatz als File-/Printserver.

Im Clientbereich hängt die Entscheidung – wie bereits angedeutet – sehr stark mit den zu verwendenden Applikationen zusammen. Für einige Einsatzbereiche sind für Linux kaum Applikationen in ausreichender Qualität zu bekommen. Besonders stark betroffen sind die Bereiche Videosoftware, Sprachverarbeitung, Dokumenten- und Workflow-Management und CRM/SRM Systeme. Auch der Einsatz so mancher Spezial- oder Branchenlösung setzt zwingend ein Windows System voraus. Aber selbst hier ist das Bild keineswegs eindeutig. So existiert

<sup>42</sup> Dies ist für manche Microsoft Serverprodukte zwingend erforderlich – z.B. für den Microsoft Exchange Server

beispielsweise sehr wohl eine Linux Version von Mathematica, während AutoCAD eine solche vermissen lässt. Von SPSS existiert eine Linux Server Version – der Client ist jedoch nur für Windows erhältlich.

Stark ist Linux im Bereich Entwicklungssoftware. Viele qualitativ hochwertige Entwicklungswerkzeuge werden mit den meisten Linux Distributionen gleich mitgeliefert, so manches kommerzielle Produkt ist auch als Linux Version erhältlich (z.B. Borland JBuilder / Kylix). Im Bereich Entwicklungswerkzeuge ist Linux wohl sogar als insgesamt stärker einzuschätzen als Windows, auch wenn nicht alle kommerziellen Entwicklungssuiten auch für Linux erhältlich sind.

Sofern nicht zwingend Microsoft Office eingesetzt werden soll, spricht auch überhaupt nichts gegen Linux im Büro. Bis auf das genannte Microsoft Produkt stehen die meisten gängigen Office Pakete auch in einer Linux Variante zur Verfügung.

Zusammenfassend kann festgestellt werden, dass die Art der einzusetzenden Applikationen letztendlich auch die Möglichkeiten des Betriebssystemeinsatzes determiniert. Neben den oben genannten Gebieten mit bisher wenig Linux Unterstützung gilt grundsätzlich, dass je mehr Software aus dem Hause Microsoft eingesetzt werden soll, desto eher auch zwingend Windows als Betriebssystem eingesetzt werden muss. Im Folgenden kann aber aufgezeigt werden, dass es sehr wohl kommerzielle und freie Alternativen zu den meisten derzeit eingesetzten Microsoft Produkten gibt. Des Weiteren gibt es immer Lösungen, auch nicht direkt für Linux erhältliche Software trotzdem unter diesem Betriebssystem zur Verfügung zu stellen. Hier wäre etwa der Einsatz eines Terminal Servers zu nennen, der von Linux Clients aus genutzt werden kann.

### 4.3 Office Pakete

Office Pakete sind eine Zusammenstellung von Software zur Erledigung gewöhnlicher Büroarbeiten, wie das Verfassen von Texten, die Erstellung von Kalkulationen, Präsentationen oder Terminverwaltung und Aufgabenplanung. Typische Komponenten eines Office Paktes sind

- Textverarbeitung
- Tabellenkalkulation
- Präsentationssoftware
- Mail Client
- Terminplaner
- Adressverwaltung
- Datenbanken

Viele Office Produkte enthalten auch noch weitere Komponenten. Im Folgenden werden die Office Pakete anhand der Komponenten Textverarbeitung, Tabellenkalkulation und Präsentationssoftware verglichen. Die Groupware- und Datenbank Komponenten werden in den jeweiligen Abschnitten betrachtet.

### 4.3.1 Merkmale von Office Paketen

Office Pakete sind eines der grundlegendsten Softwareprodukte, die in den meisten Organisationen zum Einsatz kommen, so dass auf die Office Software ein besonderes Augenmerk gerichtet werden muss. Im Folgenden werden die wichtigsten Anforderungen an Office Software und deren Merkmale dargestellt.

#### **Funktionalität**

Die benötigten Funktionen einer Textverarbeitungssoftware sind vor allem:

- Bearbeitung von Text (unter Nutzung von Formatvorlagen)
- Layout
- Einbindung von Grafiken
- Einbindung von Tabellen
- Erstellung von Indizes und Verzeichnissen
- Erstellung von mathematischen Formeln
- Rechtschreibkorrektur
- Druck bzw. Seriendruck
- Im- und Export von Dokumenten

Eine Tabellenkalkulation sollte mindestens folgende Funktionen umfassen.

- Bearbeitung von Zahlenmaterial und Formeln
- Auswertungs- und Sortierfunktionen
- Erzeugung von Diagrammen
- Druck
- Im- und Export von Daten

Funktionen einer Präsentationssoftware sind:

- Bearbeitung/Layout von Text
- Bearbeitung von Zeichnungen
- Einbindung von Grafiken / Diagrammen / Tabellen
- Erstellung von Animationen
- Im- und Export von Dokumenten
- Druck
- Präsentationskomponente zur Darstellung der Folien über Bildschirm oder Beamer

### **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit einer Office Software ergibt sich hauptsächlich über die Gestaltung der Benutzeroberfläche. Wünschenswert ist eine möglichst einfache Bedienung, sowie eine stringente und nachvollziehbare Anordnung der Funktionen. Auch die leichte und intuitive Erlernbarkeit trägt zur Benutzerfreundlichkeit bei.

### **Zuverlässigkeit / Sicherheit**

Ein zuverlässiges Office Produkt zeichnet sich insbesondere durch eine hohe Stabilität der Software aus. Im Fehlerfall müssen auch Mechanismen zur Datenrettung vorhanden sein. Das verwendete Datenformat muss möglichst fehlertolerant sein.

Es sollten auch Sicherheitsfunktionen wie Dateiverschlüsselung vorhanden sein. Die verwendeten Datenformate sollten gegenüber Angriffen und Schadsoftware (Viren und Würmer) resistent sein.

### **Leistungsfähigkeit**

Die Leistungsfähigkeit eines Office Paketes definiert sich hauptsächlich über den Umgang mit komplexen und großen Dokumenten. Aber auch die Ablaufgeschwindigkeit der Software (z.B. bei Neuberechnung von Seitenumbrüchen) fällt unter dieses Kriterium.

### **Support**

Das Kriterium des Supports misst bei Office Paketen insbesondere, inwiefern die Software durch Dienstleister unterstützt wird (z.B. Angebot von Schulungen), aber auch das vorhandene Know-how bei Anwendern sowie die Unterstützung des Paketes durch Add On Software.

## **4.3.2 Gegenüberstellung von Office Paketen**

### **Microsoft Office**

Die Office Lösung aus dem Hause Microsoft ist mit großem Abstand Weltmarktführer. Der Marktanteil von MS Office beträgt um die 90%, teilweise werden sogar bis zu 95% vermutet<sup>43</sup>. Die Marktpräsenz dieses Office Paketes ist so stark, dass seine Datenformate als de facto Datenaustauschstandards genutzt werden –

<sup>43</sup> Vgl. <http://www.us-market.de/profile/MSFT.htm>

den Umstand ignorierend, dass sie dazu aufgrund ihres proprietären Charakters sowie technischer Kriterien<sup>44</sup> eigentlich völlig ungeeignet sind.

Microsoft Office besteht aus den Komponenten Word (Textverarbeitung), Excel (Tabellenkalkulation) und Powerpoint (Präsentation). Des Weiteren wird noch die PIM Software Outlook im Standardpaket mitgeliefert. Je nach Produktvariante kommen noch weitere Komponenten hinzu, z.B. die Desktop Datenbank Access oder die Web Design Software FrontPage.

Die Textverarbeitungskomponente Microsoft Word verfügt über umfassende Funktionen zur Bearbeitung von Text, auch unter Verwendung von Format- und Dokumentvorlagen zur Sicherstellung eines einheitlichen Erscheinungsbildes. Word kann Grafiken in allen gängigen Formaten sowie Tabellen in den Text einbinden und stellt eine Vielzahl von Funktionen zur Layoutgestaltung zur Verfügung. Rechtschreib- und Grammatikprüfung sind für deutschen und englischen Text (weitere Sprachpakete können ebenfalls installiert werden) vorhanden. Word kann auch Serienbriefe mit Datenmaterial aus unterschiedlichen Quellen (u. a. aus Excel, Access oder ODBC) erzeugen. Die Erstellung von Inhalts-, Abbildungs- und Tabellenverzeichnissen sowie Indizes ist möglich. Dokumente können im proprietären DOC Format, sowie als RTF, HTML, ASCII und XML gespeichert werden. Ein PDF Export ist nicht möglich, hier muss auf Fremdsoftware zurückgegriffen werden. Auch die meisten Datenformate anderer Officeprodukte werden nur sehr zurückhaltend unterstützt. In den meisten Fällen ist nur ein Import aber kein Export von Fremdformaten möglich.

Die Tabellenkalkulation Microsoft Excel bietet eine Vielzahl an mathematischen Funktionen sowie Auswertungs- und Sortierfunktionen. Diese dürften den meisten Alltagsaufgaben gerecht werden. Aus den Tabellen können Diagramme verschiedensten Typs generiert werden. Die unterstützten Diagrammtypen dürften mit wenigen Ausnahmen die Mehrzahl der benötigten Typen abdecken. Die Diagrammbearbeitungsfunktionen sind jedoch etwas dürftig. Elemente müssen oft manuell verschoben werden, um die gewünschten Effekte zu erzielen. Wie bei Word, so lassen auch bei Excel die Exportfunktionalitäten praktisch alle Wünsche offen. Bei Tabellen werden aber wenigstens die gängigsten kommerziellen Konkurrenzprodukte unterstützt (beispielsweise Lotus 1-2-3). Die Datenformate von Open Source Produkten können aber auch mit Excel nicht verarbeitet werden. Ein grafischer Export von Diagrammen fehlt völlig – man ist hier entweder auf OLE Verknüpfungen oder gar auf Screenshots angewiesen. Auch können weder Diagramme noch Tabellen als PDF exportiert werden.

<sup>44</sup> Die Formate garantieren z.B. keine Layouttreue beim Datenaustausch

Die Präsentationssoftware PowerPoint bietet umfassende Funktionen zur Gestaltung von Vortragsfolien. Grafiken, Videoclips und Audiodaten können in den meisten gängigen Formaten importiert werden. Auch PowerPoint unterstützt weder PDF noch die Datenformate von Konkurrenzprodukten – immerhin können Folien in den verbreiteten Grafikformaten exportiert werden (z.B. GIF, JPG, BMP). Die Funktionen zum Layout von Text sind ausreichend, aber nicht ganz so komfortabel wie in Word. Stark ist PowerPoint im Erstellen von Zeichnungen – die diesbezüglichen Funktionen lassen nicht allzu viele Wünsche offen. Auch die Möglichkeiten zur Gestaltung von Folienanimationen sind für die meisten Anforderungen völlig hinreichend. Schließlich verfügt PowerPoint noch über Funktionen, die ein gesteuertes oder automatisches Ablaufen der Präsentation am Bildschirm ermöglichen.

Die Funktionalitäten von Microsoft Office genügen bis auf die absolut mangelhaften Import und Exportfunktionalitäten insgesamt den gestellten Anforderungen. Aus der Sicht von Microsoft ist die zurückhaltende Unterstützung von Fremdformaten zwar verständlich – für den Anwender ist diese Politik jedoch als nicht zu verachtendes Ärgernis zu bezeichnen. Insbesondere für die Zusammenarbeit mit Dritten wäre eine bessere Interoperabilität der verschiedenen Office Produkte wünschenswert.

Die Benutzerfreundlichkeit von Microsoft Office ist gut. Die meisten Funktionen können recht intuitiv erfasst werden, die Oberfläche ist zudem an persönliche Präferenzen anpassbar.

Die Zuverlässigkeit von Microsoft Office ist uneinheitlich. Excel und Powerpoint laufen im Betrieb weitestgehend stabil, Word ist hingegen weniger zuverlässig. Immerhin stehen Datenrettungsfunktionen für diese Fälle zur Verfügung, die in vielen Fällen den Schaden auch begrenzen können. Microsoft Office bietet Funktionen zur Verschlüsselung von Dateien, die für den einfachen Gebrauch ausreichen – allerdings ist die Office Verschlüsselung für die Verwendung für Dokumente mit hohem Sicherheitsanspruch eher wenig ratsam. Hier sollte doch eher zu spezialisierten Werkzeugen gegriffen werden.

Eine der größten Schwachstellen von Microsoft Office ist die Anfälligkeit seiner Datenformate gegenüber Makroviren. Die Sicherheitsmechanismen der Makrosprache wurden zwar mit neueren Office Versionen verbessert, insgesamt sind diese aber auch heute noch völlig unzureichend. Die Mächtigkeit der Makrosprache wirkt sich aus der Perspektive der Sicherheitsanforderungen eher

verhängnisvoll aus. Der Einsatz von Microsoft Office stellt damit immer noch ein Sicherheitsrisiko dar<sup>45</sup>.

Die Leistungsfähigkeit der Office Komponenten ist uneinheitlich. Während Excel und PowerPoint auch mit großen Dokumenten in ansprechender Geschwindigkeit umgehen können, weist Word in diesem Bereich eklatante Schwächen auf. Insbesondere der Umgang mit komplexen Dokumenten, die viele Grafiken und Tabellen enthalten, bereitet Word große Schwierigkeiten, die nur mit viel Know-how und Disziplin seitens des Anwenders zu überwinden sind.

Der Support von Microsoft Office ist dank seines Marktanteils als sehr gut zu bezeichnen. Die meisten Anwender verfügen über MS Office Know-how. Auch sind viele Zusatzprodukte von Drittanbietern für MS Office erhältlich.

### **OpenOffice.org**

OpenOffice.org<sup>46</sup> ging aus der kommerziellen Office Suite Star Office hervor. Mit der Übernahme von Star Office durch Sun wurde OpenOffice.org als Open Source Software unter mehreren Lizenzen freigegeben (LGPL und SISSL). Das Produkt wird trotzdem weiterhin im Wesentlichen von Sun weiterentwickelt und gepflegt.

Der Marktanteil von OpenOffice wird zwischen 3% und 15% gesehen, die Beschaffung genauer Zahlen ist, wie generell bei Open Source Produkten, schwierig. Im Jahr 2004 konnte OpenOffice bereits 30 Millionen Downloads verbuchen – die nicht zu vernachlässigende Verbreitung über Linux Distributionen ist in dieser Zahl nicht enthalten<sup>47</sup>.

OpenOffice enthält eine Textverarbeitung (Writer), eine Tabellenkalkulation (Calc), eine Präsentationskomponente (Impress) und ein Zeichenprogramm (Draw). Ab der Version 2.0 ist auch eine Desktop Datenbank enthalten (Base). Es gibt derzeit keine PIM Komponente für OpenOffice.

OpenOffice wird als direktes Konkurrenzprodukt zu Microsoft Office positioniert. Daher ist es wenig verwunderlich, dass OpenOffice eine sehr starke Ähnlichkeit zu Microsoft Office besitzt. Die Übereinstimmung zwischen sämtlichen Komponenten von Microsoft Office und OpenOffice ist so hoch, dass die obigen Ausführungen bezüglich der Funktionalitäten und Benutzerfreundlichkeit weitgehend auch für OpenOffice gelten. Die Import und Export Funktionalitäten sind bei OpenOffice allerdings deutlich besser. Ein Export direkt nach PDF ist möglich. OpenOffice kommt mit den Datenformaten der meisten Konkurrenz-

<sup>45</sup> Siehe dazu auch: <http://www.pcwelt.de/know-how/sicherheit/31584/index6.html>

<sup>46</sup> <http://www.openoffice.org>

<sup>47</sup> Vgl. [http://de.openoffice.org/presse/interviews/interview\\_macgadget\\_12\\_2004.html](http://de.openoffice.org/presse/interviews/interview_macgadget_12_2004.html)

produkte klar, darunter auch Microsoft Office. Es sollte jedoch nicht unerwähnt bleiben, dass aufgrund des proprietären Charakters der Microsoft Datenformate keine 100% Kompatibilität besteht. Im Einzelfall – besonders bei komplexen Dokumenten – kann es beim Datenaustausch zwischen Microsoft Office und OpenOffice zu Problemen kommen, insbesondere wenn auf beiden Seiten nicht nur lesend, sondern auch bearbeitend auf die Dokumente zugegriffen werden soll.

OpenOffice läuft insgesamt etwas stabiler als Microsoft Office. Des Weiteren verfügt OpenOffice über vergleichbare Sicherheitsfunktionalitäten. Eine vergleichbare Gefahr durch Makroviren, wie dies bei MS Office der Fall ist, existiert derzeit bei OpenOffice nicht.

Die Leistungsfähigkeit von OpenOffice ist als sehr hoch zu bewerten. Alle Komponenten dieses Softwarepaktes kommen mit großen und komplexen Dokumenten gut klar. In dieser Beziehung ist es der Konkurrenz aus dem Hause Microsoft klar überlegen.

Der Support durch Dienstleister kann sich derzeit noch nicht mit MS Office messen. Es werden aber sowohl Unterstützungsdienstleistungen als auch Schulungen angeboten. Natives OpenOffice Know-how ist bei Anwendern kaum vorhanden – aber auch kaum nötig, da sich die meisten MS Office Kenntnisse direkt auf OpenOffice übertragen lassen. Dies wird sich zukünftig sogar noch verbessern, insbesondere da sich die Bedienung der neuen OpenOffice Version 2.0 noch stärker als bisher an MS Office anlehnt.

### **Word Perfect Suite**

Die kommerzielle Office Suite aus dem Hause Corel verfügt über eine Textverarbeitung, eine Tabellenkalkulation und eine Präsentationssoftware. Eine Datenbankkomponente fehlt ebenso wie eine PIM Software.

Die Funktionalität der Word Perfect Suite kann sich mit Microsoft Office und OpenOffice durchaus messen, teilweise bietet sie sogar darüber hinausgehende Funktionen, insbesondere beim Layout von Texten und bei Diagrammen in der Tabellenkalkulation. Die Benutzerführung ist stringent und einfach, dafür weist die Textverarbeitungskomponente ähnliche Schwächen beim Umgang mit sehr großen und komplexen Dokumenten auf wie Microsoft Word.

Die Import und Export Filter sind mit die stärksten im Office Bereich – ein Datenaustausch etwa mit MS Office funktioniert relativ reibungslos. Die OpenOffice Formate kann die aktuelle Version aber noch nicht verarbeiten. Ein PDF Export von Dokumenten ist jedoch möglich.

Der Marktanteil der Word Perfect Suite betrug einst ca. 50%, ist heutzutage aber verschwindend gering. Anwender Know-how ist nur schwer oder gar nicht mehr zu bekommen, ebenso wie Dienstleistungen rund um dieses Office Paket.

### **Lotus Smart Suite**

Lotus Smart Suite ist ebenfalls eine kommerzielle Office Suite, die mit dem Kauf der Firma Lotus auf IBM übergang. Sie besteht aus der Textverarbeitung WordPro (früher AmiPro), der Tabellenkalkulation Lotus 1-2-3, der Präsentationssoftware Freelance sowie der Desktop Datenbank Approach.

Auch diese Office Suite erfüllt die üblicherweise gestellten Anforderungen bezüglich der Funktionalität. Die Import und Export Funktionalitäten sind gut. Die Benutzerfreundlichkeit ist eine der Stärken dieses Office Paketes. Bezüglich der Leistungsfähigkeit und Zuverlässigkeit fällt Smart Suite weder positiv noch negativ auf.

Auch Lotus Smart Suite hatte einst einen beachtlichen Marktanteil, ist heute aber unbedeutend und wird von IBM auch nur noch zurückhaltend vermarktet.

### **KOffice**

KOffice<sup>48</sup> ist eine freie Office Suite unter dem Dach des KDE Projektes. Die GPL lizenzierte Software besteht aus den Komponenten KWord (Textverarbeitung), KSpread (Tabellenkalkulation), KPresenter (Präsentationssoftware) sowie einem Vektorzeichenprogramm und einem Malprogramm. Eine Komponente zur Erstellung von Flussdiagrammen ist ebenso vorhanden wie ein Reportgenerator.

Die KOffice Komponenten sind unterschiedlich ausgereift. In manchen Bereichen reicht der Funktionsumfang durchaus an die anderen vorgestellten Produkte heran, aber insgesamt kann KOffice mit diesen noch nicht mithalten. KOffice bietet aber einige Komponenten, die über das übliche Maß einer Office Suite hinausgehen.

Die Import und Export Funktionen sind vorhanden aber noch verbesserungsbedürftig, insbesondere im Zusammenspiel mit MS Office. Ein PDF Export steht zur Verfügung. Als Datenformat verwendet KOffice dasselbe Datenformat wie OpenOffice – ein Austausch mit dieser Office Suite funktioniert daher reibungslos.

KOffice ist sehr schlank und übersichtlich gehalten und kann daher als dementsprechend benutzerfreundlich und einfach zu erlernen betrachtet werden.

<sup>48</sup> <http://www.koffice.org/>

### Gnome Office

Gnome Office<sup>49</sup> versteht sich weniger als eine voll integrierte Office Suite, sondern eher als ein loser Verbund von Applikationen. Kernapplikationen sind eine Textverarbeitung (AbiWord), eine Tabellenkalkulation (Gnumeric) und eine Datenbankkomponente (Gnome DB). Teilweise zählt man noch weitere Komponenten zu Gnome Office, insbesondere das Zeichenprogramm GIMP und die PIM Software Evolution. Eine Präsentationskomponente ist nicht (mehr) vorhanden. Die Gnome Office Komponenten sind GPL lizenziert.

Generell muss aber festgehalten werden, dass bis auf die recht ausgereifte Tabellenkalkulation Gnumeric die Gnome Office Komponenten fortgeschrittenen Anforderungen eher noch nicht genügen werden.

#### 4.3.3 Einsatzerfahrungen

Microsoft Office ist die dominierende Office Suite. Dies wird wohl des Öfteren nicht zuletzt mit dem Wunsch nach Verwendung der verbreiteten Microsoft Datenformate und dem bei vielen Nutzern vorhandenen Anwender Know-how zusammenhängen. Mit rein funktionalen Gründen kann die Dominanz von MS Office jedenfalls nicht erklärt werden – andere hier vorgestellte Office Suites erfüllen die alltäglichen Anforderungen mindestens ebenso gut – in manchen Fällen sogar deutlich besser. Insbesondere die Textverarbeitung Word wird von fast allen hier betrachteten Textverarbeitungen übertroffen.

Der Reifegrad von Open Source Office Paketen wurde relativ zu relevanten kommerziellen Softwareprodukten und ebenfalls auf einer fünf-stufigen Skala bewertet. Der Reifegrad wurde separat für die fünf Indikatoren für Softwarequalität erhoben:

- Funktionalität: 3,8
- Benutzerfreundlichkeit: 3,9
- Zuverlässigkeit, Sicherheit: 3,9
- Leistungsfähigkeit: 3,5
- Support: 3,0

Damit liegen die freien Office Pakete unter den am höchsten bewerteten Open Source Softwarekategorien. Insbesondere die Benutzerfreundlichkeit der freien Office Software wurde als bemerkenswert hoch betrachtet. Zusammen mit den freien Betriebssystemen konnten Open Source Office Pakete die beste Bewertung in diesem Kriterium erreichen.

<sup>49</sup> <http://www.gnome.org/gnome-office/>

#### 4.3.4 Bewertung und Empfehlung

Die folgende Tabelle fasst die diskutierten Ergebnisse der Gegenüberstellung der freien und kommerziellen Office Pakete in einer Übersichtsdarstellung zusammen:

Kriterium	Microsoft Office	Open Office.org	WordPerfect Suite	Smart Suite	KOffice	Gnome Office
<b>Lizenz</b>	Kommerziell	OSS (LGPL, SISSL)	Kommerziell	Kommerziell	OSS (GPL)	OSS (GPL)
<b>Funktionalität Textverarbeitung</b>	●◐	●	●	●	●◐	○
Textbearbeitung / Layout	●◐	●	●	●	●◐	●◐
Tabellen / Grafiken	●	●	●	●	●◐	○
Indizes / Verzeichnisse	●◐	●◐	●◐	●◐	○	○
Formeleditor	●◐	●◐	●	●	●	○
Rechtschreibkorrektur	●	●	●	●	●◐	●◐
Druck / Seriendruck	●◐	●	●◐	●◐	●◐	●◐
Import und Exportfilter	○	●	●	●◐	●◐	●◐
<b>Funktionalität Tabellenkalkulation</b>	●	●	●	●	●◐	●◐
Bearbeitung von Zahlenmaterial und Formeln	●	●	●	●	●◐	●◐
Auswertungs- und Sortierfunktionen	●	●	●	●◐	○	○
Diagramme	●◐	●◐	●◐	●◐	●◐	○
Druck	●◐	●	●	●	●◐	●◐
Import und Exportfilter	○	●	●	●	●◐	●◐
<b>Funktionalität Präsentationssoftware</b>	●	●	●	●◐	○	—
Bearbeitung / Layout von Text	●◐	●◐	●	●◐	○	—
Bearbeitung von Zeichnungen	●	●	●	●◐	○	—
Einbindung Grafiken/ Diagrammen/Tabellen	●	●	●	●◐	○	—

Kriterium	Microsoft Office	Open Office.org	WordPerfect Suite	Smart Suite	KOffice	Gnome Office
Erstellung von Animationen	●	●	●	◐	○	—
Druck	●	●	●	●	◐	—
Bildschirmpräsentation	●	●	●	●	◐	—
Import und Exportfilter	○	●	◐	◐	○	—
<b>Benutzerfreundlichkeit</b>	◐	◐	◐	●	●	●
Bedienung	●	●	●	●	●	●
Erlernbarkeit	◐	◐	◐	●	●	●
<b>Zuverlässigkeit / Sicherheit</b>	◐	●	●	●	◐	◐
Stabilität	◐	●	●	●	●	●
Sicherheitsfunktionen	◐	◐	◐	◐	○	○
Schutz vor Angriffen	○	●	●	●	◐	◐
<b>Leistungsfähigkeit</b>	◐	●	●	●	◐	◐
Ablaufgeschwindigkeit	◐	●	●	●	●	●
Umgang mit großen komplexen Dokumenten (Textverarbeitung)	○	●	◐	◐	◐	○
Umgang mit großen komplexen Dokumenten (Tabellenkalkulation und Präsentation)	●	●	●	●	◐	◐
<b>Support</b>	●	◐	○	○	○	○
Unterstützung durch Dienstleister	●	◐	○	○	○	○
Vorhandenes Know-how	●	●	○	○	○	○
Vorhandene Zusatzsoftware	●	◐	○	○	○	○

Tabelle 18: Gegenüberstellung von Office Paketen  
 (Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung)

Für den produktiven Einsatz derzeit eher nicht zu empfehlen ist GNOME Office. Dieser Office Suite mangelt es an praktisch allen fortgeschrittenen Funktionen und ist lediglich für einfache Aufgaben geeignet. Auch arbeiten die nur lose integrierten Komponenten nicht hinreichend genug zusammen. Als Ausschlusskriterium erweist sich weiterhin das Fehlen einer Präsentationssoftware.

Auch KOffice bietet derzeit noch nicht alle benötigten oder gewünschten Funktionen, ist aber ausgereifter als GNOME Office. Noch kann sich diese Office Suite im Funktionsumfang nicht mit den meisten anderen Office Produkten messen, allerdings sind teilweise rasante Entwicklungsfortschritte erkennbar, so dass KOffice zukünftig durchaus zur Alternative werden könnte, nicht zuletzt durch die Verwendung des durch OASIS standardisierten Datenformates.

Für die kommerziellen Produkte WordPerfect Suite und Smart Suite kann festgestellt werden, dass beide die üblichen Anforderungen der Fraunhofer-Gesellschaft erfüllen würden. Die beiden Produkte können im Markt allerdings nur noch eine Nischenstellung aufweisen und werden durch ihre jeweiligen Hersteller auch nicht mehr konsequent vermarktet. Die Vermutung liegt nahe, dass die Rechteinhaber der beiden Office Suites den Kampf gegen den Marktführer Microsoft Office weitgehend aufgegeben haben, was Fragen bezüglich der Zukunftssicherheit der beiden Produkte aufwirft. Da beide Office Suites auch keinen nennenswerten Vorteil gegenüber Microsoft Office und OpenOffice bieten, kann auch der Einsatz dieser Produkte nicht empfohlen werden.

Für den produktiven Einsatz bieten sich daher die Produkte Microsoft Office und OpenOffice an, die weitgehend funktionsäquivalent sind und die gestellten Anforderungen erfüllen. Einige, in MS Office vorhandene Funktionalitäten sind zwar in OpenOffice nicht vorhanden – dies trifft aber umgekehrt ebenso zu. In der Regel handelt es sich dabei ohnehin eher um weniger oft benötigte Funktionen.

Für Microsoft Office spricht vor allem dessen Verbreitung. Insbesondere im Datenaustauschverkehr versenden viele Anwender die MS Office Dateiformate. Es sind auch besonders viele Vorlagen für diese Formate erhältlich. Nicht zuletzt verfügen sehr viele Anwender über MS Office Know-how.

Aus funktionaler Sicht spricht gegen Microsoft Office insbesondere die schwache Textverarbeitung Word. Diese genügt den Anforderungen an eine moderne Textverarbeitung nur mit Einschränkungen, insbesondere aufgrund des mangelhaften Umgangs mit großen und komplexen Dokumenten. In diesem Punkt sind fast alle hier gegenübergestellten Textverarbeitungen stärker.

Gegen Microsoft Office spricht auch die in Zusammenhang mit diesem Office Produkt auftretende Sicherheitsproblematik. Keines der anderen Office Produkte weist ein ähnlich hohes Gefährdungspotenzial durch Makroviren und andere Angriffsvarianten auf.

### 4.3.5 Datenaustausch im Office Bereich – Anforderungen und Lösungen

Wie oben bereits dargestellt, ergibt sich in Praxis oft eine vergleichsweise starke Notwendigkeit zur Verarbeitbarkeit von Dokumenten im Microsoft Office Format. Dies wird vor allem durch die Projektarbeit bedingt, in denen Mitarbeiter unterschiedlichster Organisationen mit der Fraunhofer-Gesellschaft zusammenarbeiten und dabei Dokumente austauschen. Verstärkt wird diese Anforderung noch durch die Praxis verschiedener Institutionen, Dokumente im Microsoft Office Format zu übersenden.

Dieser Anforderung kann natürlich durch den Einsatz von Microsoft Office genügt werden. Sollte aber die Entscheidung zukünftig zugunsten einer anderen Office Software fallen, müsste für die erforderlichen Fälle das Microsoft Format erstellt bzw. gelesen werden können. Denn es ist auch für die Zukunft nicht davon auszugehen, dass Microsoft an einer Interoperabilität seiner Office Software mit Konkurrenzprodukten insbesondere aus dem Open Source Bereich interessiert sein wird. Die Einführung eines XML basierten Formates in Microsoft Office wird diese Situation wohl nicht ändern – es wird lediglich ein proprietäres Binärformat durch ein proprietäres XML Format ersetzt.

Die Gegenüberstellung ergab, dass die einzige alternative Office Software die derzeit empfohlen werden kann, OpenOffice ist. Es wurde daher ein Praxistest durchgeführt, der eine tendenzielle Aussage bezüglich der Interoperabilität von Microsoft Office und OpenOffice zum Ziel hatte. Als Testdokumente wurden für eine große Organisation typische Dokumente gewählt:

- Einfacher, einseitiger Brief im Unternehmens-Cl
- 6-seitiges Angebot (nur Text)
- Projekt-Abschlussbericht (komplexes Dokument mit zahlreichen Grafiken, Tabellen sowie einem Inhalts- und Abbildungsverzeichnis)
- Einfache Tabellenkalkulation (ein einzelnes Sheet; einfache Formeln)
- Komplexe Tabellenkalkulation (mehrere verknüpfte Sheets mit komplexen Berechnungen, Filtern, und Diagrammen)
- Einfache Präsentation (wenige Folien; überwiegend Text; ein Logo pro Folie)
- Komplexe Präsentation (viele Folien mit Überblendungs- und Animationseffekten; zahlreiche und komplexe Grafiken und Diagramme)

Die Dokumente waren allesamt tatsächlich verwendete Dokumente, die nicht speziell für die Studie hergestellt wurden.

#### 4.3.5.1 Szenario 1: Lesender Zugriff auf Microsoft Office Dokumente

Dieses Szenario würde beispielsweise nach einer Migration von Microsoft Office nach OpenOffice eintreten. Viele bestehende Altdokumente müssten noch gelesen, aber nicht mehr verändert werden. Des Weiteren deckt dieses Szenario

Fälle ab, in denen Anwender mit OpenOffice Arbeitsplätzen von externer Seite Dokumente im MS Office Format zugesendet bekommen die sie zwar lesen und drucken, aber nicht verändern müssten.

Die oben genannten Dokumente wurden mit MS Office 2003 gespeichert und anschließend in OpenOffice 2.0 eingelesen. Das Ergebnis wurde zunächst am Bildschirm begutachtet. Anschließend wurden die Dokumente sowohl aus MS Office als auch aus OpenOffice heraus ausgedruckt und verglichen. Bei der Präsentation wurde zusätzlich in beiden Office Produkten eine Bildschirmpräsentation durchgeführt.

Die folgende Tabelle stellt das Ergebnis des Tests dar.

	Lesbarkeit	Layouttreue	Drucktreue	Bildschirmpräsentation
CI Brief	●	●	●	-
Projektangebot (einfacher CI Text)	●	◐	◐	-
Abschlussbericht (komplexer CI Text)	●	◐	◐	-
Einfache Kalkulation	●	●	●	-
Komplexe Kalkulation	●	◐	◐	-
Einfache Präsentation	●	●	●	●
Komplexe Präsentation	●	●	●	●

Tabelle 19: Datenaustausch zwischen MS Office und OpenOffice.org (legend)  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung)

In allen Fällen konnten die Dokumente erfolgreich nach OpenOffice importiert werden. Verluste von Inhalten traten nicht auf. Eine 100-prozentige Lesbarkeit der Dokumente war somit gegeben. Das Layout wurde bei den einfachen Dokumenten praktisch unverändert übernommen, die Abweichungen vom Original waren marginaler Art. Mit steigender Komplexität der Dokumente nahm aber auch die Zahl der Abweichungen vom Layout zu. Insbesondere beim komplexen Textdokument wurden Seitenumbrüche, Seitenränder und Inhaltsverzeichnisse teilweise erheblich verändert, so dass das importierte Dokument nicht mehr den CI Anforderungen genügte. Hier muss mit Nachbearbeitungsaufwand beim Datenaustausch gerechnet werden.

Gut funktioniert der Datenaustausch bei der Präsentationskomponente. Nach dem Import waren kaum Unterschiede zum Original sichtbar. Die Bildschirmpräsentation stellte Folienübergänge und Animationen auch in der komplexen Präsentation korrekt dar.

In der Tabellenkalkulation wurden die Tabellen reibungslos übernommen. In den Tabellenblättern selbst war nach dem Import auch kaum ein Unterschied zum Original feststellbar. Die Diagramme wiesen allerdings teilweise erhebliche optische Abweichungen zum Original auf.

Zusammengefasst kann festgestellt werden, dass die OpenOffice Importfilter einen lesenden Zugriff auf Microsoft Office Dokumente relativ problemlos ermöglichen. Schwierigkeiten gab es insbesondere bei der Textverarbeitung, da der Importfilter die Seitenränder veränderte. Damit wird ein originalgetreuer Druck unmöglich. Es muss allerdings festgestellt werden, dass dieses Problem sogar bei Word selbst auftritt. Auch hier treten beim Datenaustausch zwischen Word Anwendern des Öfteren erhebliche Probleme auf.

#### **4.3.5.2 Szenario 2: Lesender und schreibender Zugriff auf Microsoft Office Dokumente in OpenOffice**

In diesem Szenario wurden mit Microsoft Office Dokumente erzeugt, anschließend in OpenOffice eingelesen und im OpenOffice Dateiformat abgespeichert. Danach wurden die Dokumente wieder geöffnet, einige kleinere Änderungen vorgenommen, im Microsoft Office Format gespeichert und wieder in MS Office eingelesen. Das Resultat wurde anschließend mit den Ausgangsdokumenten verglichen. Die verwendeten Dokumente waren dieselben wie in Szenario 1. Dieses Szenario simulierte ein gemeinsames Arbeiten am selben Dokument mit verschiedenen Office Paketen.

Die einfachen Dokumente konnten auf diese Weise alle gut bearbeitet werden. Es waren keine größeren Abweichungen zum Original zu beobachten. Der mittelkomplexe Text bestand das Testszenario ebenfalls recht ordentlich, mit lediglich kleineren Abweichungen. Diese konnten mit wenigen Handgriffen behoben werden.

Bei den komplexen Dokumenten schnitt wieder die Präsentationskomponente ab besten ab. Es traten hier keine nennenswerten Abweichungen gegenüber dem Original auf.

Die Tabellenkalkulation selbst ließ sich ebenfalls problemlos austauschen, für die Diagramme gilt dies jedoch mit Einschränkungen. Hier waren einige Abweichungen zum Original sichtbar, insbesondere bei Achsenbeschriftungen.

Nicht gut funktionierte der Datenaustausch mit dem komplexen Text. Wie schon beim reinen Lesevorgang waren auch bei schreibendem Zugriff erhebliche Abweichungen gegenüber dem Original zu beobachten. Besonders betroffen waren wieder Seitenränder und Inhaltsverzeichnisse.

	Lesbarkeit	Layouttreue	Drucktreue	Bildschirmpräsentation
CI Brief	●	●	●	-
Projektangebot (einfacher CI Text)	●	◐	◐	-
Abschlussbericht (komplexer CI Text)	●	○	○	-
Einfache Kalkulation	●	●	●	-
Komplexe Kalkulation	●	◐	◐	-
Einfache Präsentation	●	●	●	●
Komplexe Präsentation	●	●	●	●

Tabelle 20: Datenaustausch zwischen MS Office und OpenOffice.org (lesend und bearbeitend)  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung)

Zusammenfassend kann festgestellt werden, dass ein Datenaustausch zwischen Microsoft Office und OpenOffice zwar weitgehend möglich ist, es in der Praxis aber zu Problemen insbesondere bei Texten kommen kann. Hier ist mit Nachbearbeitungsaufwand zu rechnen, der bei komplexen Texten nicht unerheblich sein kann.

#### 4.3.5.3 Wertung

Reibungslos funktioniert der Austausch von Dokumenten zwischen Microsoft Office und OpenOffice nicht, auch wenn die Ergebnisse angesichts der Tatsache, dass Microsoft seine Dateiformate nicht offen legt, erstaunlich gut sind. Die Importfilter von OpenOffice lassen einen lesenden Zugriff auf Altdokumente auf jeden Fall zu. Auch von externer Quelle zugesendete Microsoft Dokumente können somit gelesen werden.

Im umgekehrten Fall, nämlich beim Versenden von Dokumenten an Dritte ist der Normalfall eher darin zu sehen, dass Office Dokumente nicht im nativen Format, sondern im Austauschformat PDF versendet werden. In diesem Fall ist

OpenOffice sogar überlegen, denn es beherrscht im Gegensatz zu MS Office den PDF Export schon von Haus aus.

Es gibt jedoch auch Fälle, in denen verschiedene Mitarbeiter eines Projektteams am selben Dokument arbeiten müssen. Bei externen Mitarbeitern muss im Licht der derzeitigen Marktsituation damit gerechnet werden, dass diese Microsoft Office einsetzen. Es konnte gezeigt werden, dass in diesen Fällen (insbesondere bei Textdokumenten) mit Problemen gerechnet werden muss, da OpenOffice das Microsoft Word Format zwar mit guter, aber nicht perfekter Qualität verarbeiten kann. Trotzdem muss festgehalten werden, dass – obwohl das verteilte Arbeiten an Dokumenten zweifelsohne vorkommt – dies wohl nur eine kleine Minderheit aller Dokumente betrifft, die in einem Unternehmen erzeugt und verarbeitet werden.

Es kann angesichts der technisch sehr starken Alternativen zu Microsoft Office nicht empfohlen werden, die Entscheidung für ein bestimmtes Office Paket ausschließlich von der Frage abhängig zu machen, ob dieses auch von der Mehrheit aller Organisationen eingesetzt wird, oder nicht.

In vielen Fällen ist ein Datenaustausch zwischen Microsoft Office und OpenOffice ohnehin möglich. In den Fällen in denen dies nicht zweckmäßig ist, kann das Problem teilweise auch mit organisatorischen Mitteln angegangen werden (z.B. Aufteilen des Dokuments in unabhängige Einzelteile, Verfassen des Dokuments in der jeweils verwendeten Office Anwendung, als letzter Schritt erfolgt Zusammenführung des Dokuments in einer vorab definierten Leitanwendung mittels Kopieren/Einfügen und Nachformatierung). Mittelfristig ist auch zu erwarten, dass die Qualität der Import und Exportfilter von OpenOffice weiter steigen wird.

#### 4.4 Wissenschaftlich-technische Software

Die Anwendungsbereiche von wissenschaftlich-technischer Software sind vielfältig. So existiert z.B. CAD-Software für Konstruktionen im Baubereich, in der Elektrotechnik aber auch im Ingenieurwesen. Jeder dieser Bereiche stellt unterschiedliche Anforderungen an die Funktionalitäten einer CAD-Software.

Im Rahmen der Studie konnten insgesamt fünf bedeutende Kategorien von wissenschaftlich-technischer Software identifiziert werden:

- **Mathematische Software:**  
Schwerpunkt bildet die Programmierung mathematischer Berechnungsverfahren zur Lösungen von Problemstellungen diverser Disziplinen, wie z.B. aus dem Ingenieurwesen, Finanzen, Versicherungen und Naturwissenschaften;
- **Analyse- und Visualisierungssoftware:**  
Kern ist die Datenanalyse mit Hilfe unterschiedlicher Methoden der Auswertung und die Darstellung der ermittelten Ergebnisse. Einige Programme unterstützen ebenso die Datenerfassung;
- **Simulationen:**  
Die Software unterstützt bei der Aufstellung von Modellen und Simulationen;
- **Konstruktionssoftware:**  
Unterstützung bei der Erstellung von Entwürfen, Plänen, Konstruktionszeichnungen, etc. hauptsächlich im Ingenieurwesen;
- **Referenzverwaltung:**  
Verwaltung von Literaturquellen eines einzelnen Mitarbeiters oder mehrerer Benutzer eines Forschungsteams.

Aufgrund der unterschiedlichen Einsatzgebiete der wissenschaftlich-technischen Software und der Vielfalt von angebotenen Softwareprodukten zu diesem Bereich fällt eine Bewertung der Software nach FURPS wie in den anderen Anwendungsfeldern schwer und war im Rahmen dieser Studie nicht möglich.

Um dennoch einen Überblick über die Open Source Software in diesem Bereich zu geben, wird nachfolgend eine Auswahl an Softwareprodukten zu diesem Bereich aufgelistet und kurz beschrieben. Ziel ist es dabei einen kurzen Überblick über die Softwareprodukte der fünf Kategorien zu geben.

Je nach Anwendungsgebiet und Anforderungen stellen einige der Open Source Softwareprodukte Alternativen zu den kommerziellen Softwareprodukten dar. Ob Alternativen für einen konkreten Anwendungsfall existieren, muss jedoch im Einzelfall überprüft werden.

#### 4.4.1 Mathematische Software

- **Mathematica (kommerzielle Software):**<sup>50</sup> Mathematica ist ein Softwaresystem zur Lösung von Problemstellungen in denen Berechnungen und Simulationen aller Art notwendig sind. Dem Benutzer werden mathematische Funktionen zur Lösung von Aufgaben aus dem Ingenieurwesen, der Finanz- und Versicherungsmathematik, der Mathematik, der Statistik, der Physik, dem »Chemical Computing« und weiteren Naturwissenschaften.
- **GNU Octave (Open Source Software):**<sup>51</sup> GNU Octave ist ein Programm zur Lösung linearer und nicht-linearer Problemstellungen. Die Programmiersprache ist größten Teils kompatibel zu Matlab.
- **SciLab (Open Source Software):**<sup>52</sup> Scilab ist ein wissenschaftliches Software Paket für Berechnungen in den Bereichen des Ingenieurwesens sowie der Naturwissenschaften, wie z.B. lineare Algebra, Matrizen, Signalverarbeitung. Die Entwicklung begann 1990.
- **Maxima (Open Source Software):**<sup>53</sup> Macysma ist seit 1982 die Grundlage für Programme wie Maple und Mathematica und wurde bis 2001 entwickelt. Seit 2001 wurde der Source Code unter dem Namen Maxima weiterentwickelt. Maxima ist ein Berechnungsprogramm mit Funktionen zur Berechnung von Polynomen, Matrizen, rationale Funktionen, Integrale, etc.
- **IT++ (Open Source Software):**<sup>54</sup> Der Kernel von IT++ ähnelt dem von Matlab. IT++ ist eine C++ Library mit Klassen und Funktionen für Mathematik, Signalverarbeitung, Sprachbearbeitung und Kommunikation. Seit 2004 wird IT++ als Teil des European network of excellence NEWCOM betrachtet<sup>55</sup>.
- **Yacas (Open Source Software):**<sup>56</sup> Yacas ist ein Computer Algebra System (CAS).
- **GAP (Open Source Software):**<sup>57</sup> Groups, Algorithms, Programming (GAP) unterstützt bei Lösungen mit diskreter Algebra.

<sup>50</sup> <http://www.additive-net.de/software/mathematica/index.shtml>

<sup>51</sup> <http://www.octave.org/>

<sup>52</sup> <http://scilabsoft.inria.fr/>

<sup>53</sup> <http://maxima.sourceforge.net/>

<sup>54</sup> <http://itpp.sourceforge.net/latest/>

<sup>55</sup> <http://newcom.ismb.it/>

<sup>56</sup> <http://yacas.sourceforge.net/infoindex.html>

<sup>57</sup> <http://www.gap-system.org/>

#### 4.4.2 Analyse- und Visualisierungssoftware

- **Origin (kommerzielle Software):**<sup>58</sup> Visualisierungs- und Datenanalyse-Software für Wissenschaftler und Ingenieure. Neben Funktionen zur Integration verschiedener Datenquellen besitzt das Programm Funktionen zur Datenanalyse, zur Visualisierung von Daten und zur individuellen Programmierung der Ein- und Ausgabe am Bildschirm.
- **MATLAB (kommerzielle Software):**<sup>59</sup> MATLAB ist eine Sprache für wissenschaftlich-technische Berechnungen und eine interaktive Umgebung zur Algorithmen-Entwicklung, zur Visualisierung und Analyse von Daten sowie für numerische Berechnungen. Mit MATLAB lassen sich technische Probleme schneller lösen als mit herkömmlichen Programmiersprachen wie C, C++ und Fortran.
- **LabVIEW (kommerzielle Software):**<sup>60</sup> LabVIEW ist eine grafische Programmiersprache, die hauptsächlich für messtechnische Aufgaben eingesetzt wird - von der einfachen Datenerfassung bis zur komplexen Prüfstandssteuerung. LabVIEW ist auch zur Auswertung, Analyse und Dokumentation von (Mess-) Daten geeignet.
- **SPSS (kommerzielle Software):**<sup>61</sup> Datenanalyse Software u.a. für die Bereiche Befragungen, Electronic Business und Web Analysen, Analyse von Kundenbedürfnissen, Marktforschung und Datenerhebung.
- **Scicraft (Open Source Software):**<sup>62</sup> Scicraft ist speziell für die Integration und Anwendung von Datenanalyse-Methoden für die Anwendungsfelder Bioinformatik, Statistik und künstliche Intelligenz erschaffen worden.
- **GNU Plot (Open Source Software):**<sup>63</sup> GNU Plot unterstützt interaktiv das 2D- oder 3D-Plotten von Daten und Funktionen. Das Programm wird seit 1986 entwickelt.
- **R (Open Source Software):**<sup>64</sup> R ist eine Software-Umgebung für statistische Berechnungen und grafische Darstellung der Ergebnisse.

<sup>58</sup> <http://www.originlab.com/>

<sup>59</sup> <http://www.mathworks.de/>

<sup>60</sup> <http://www.labview-service.de/>;

<sup>61</sup> <http://www.spss.com/de/>

<sup>62</sup> <http://www.scicraft.org/>

<sup>63</sup> <http://www.gnuplot.info/>

<sup>64</sup> <http://www.r-project.org/>

Dazugehörige GUI sind z.B. R Commander<sup>65</sup>, Java GUI for R (JGR)<sup>66</sup> oder SciViews R<sup>67</sup>.

- **SciPy (Open Source Software):**<sup>68</sup> Open Source Library mit Software-Werkzeugen für Python - nützlich für den naturwissenschaftlichen Bereich und das Ingenieurwesen.
- **MayaVi Data Visualizer (Open Source Software):**<sup>69</sup> MayaVi ist ein Programm zur Darstellung wissenschaftlicher Daten.
- **PARI/GP (Open Source Software):**<sup>70</sup> PARI/GP ist ein weit verbreitetes Computeralgebrasystem. Ausgelegt für schnelle Berechnungen aus der Zahlentheorie (Faktorisierungen, Algebraische Zahlentheorie, Elliptische Kurven, etc.), enthält es auch eine große Anzahl anderer nützlicher Funktionen zur Manipulation mathematischer Objekte wie Matrizen, Polynome, Potenzreihen, algebraische Zahlen usw. sowie eine Reihe transzendenter Funktionen. PARI steht auch als C-Bibliothek zur Implementierung schneller Berechnungen zur Verfügung.
- **SalStat (Open Source Software):**<sup>71</sup> SalStat unterstützt bei der statistischen Analyse von wissenschaftlichen Daten.
- **MacAnova (Open Source Software):**<sup>72</sup> MacAnova unterstützt bei der statistischen Analyse von wissenschaftlichen Daten und bei der Berechnung von Matrizen.

<sup>65</sup> <http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>

<sup>66</sup> <http://www.rosuda.org/JGR/>

<sup>67</sup> <http://www.sciviews.org/SciViews-R/>

<sup>68</sup> <http://www.scipy.org/>

<sup>69</sup> <http://mayavi.sourceforge.net/>

<sup>70</sup> <http://pari.math.u-bordeaux.fr/>

<sup>71</sup> <http://salstat.sourceforge.net/>

<sup>72</sup> <http://www.stat.umn.edu/macanova/>

### 4.4.3 Simulationen

- **SNNS (Open Source Software):**<sup>73</sup> Der Stuttgart Neural Network Simulator (SNNS) ist ein Softwaresimulator für neuronale Netzwerke. SNNS wurde an der Universität Stuttgart mit dem Ziel entwickelt, effiziente und flexible Simulationen von Anwendungen neuronaler Netze zu ermöglichen. Die Graphiken werden in 2D oder 3D angezeigt.
- **ANSYS (kommerzielle Software):**<sup>74</sup> ANSYS entwickelt Simulationslösungen für den Ingenieurbereich, z.B. um die Eignung von Produktentwicklungen vor dem Prototyping zu prüfen.

### 4.4.4 Konstruktionssoftware

- **Cadence (kommerzielle Software):**<sup>75</sup> Cadence ist eine Plattform mit Entwicklungs- und Konstruktionstechnologien und –methoden für die Elektrotechnik im Nanometer Bereich.
- **Systematics (kommerzielle Software):**<sup>76</sup> 3D-Entwicklungs- und Konstruktionssoftware.
- **AutoCAD (kommerzielle Software):**<sup>77</sup> AutoCAD ist eine 2D-/3D-CAD-Software zur Erstellung und Dokumentation von Entwürfen, Plänen, Detail- und Konstruktionszeichnungen mit zahlreichen Anpassungs- und Erweiterungsmöglichkeiten.
- **BRL-CAD (Open Source Software):**<sup>78</sup> BRL-CAD ist ein 3D-Modelling System mit Funktionen wie interaktiver Geometry Editor, Ray Tracing Unterstützung für Rendering und geometrische Analyse und Bildbearbeitung.
- **PythonCAD (Open Source Software):**<sup>79</sup> PythonCAD ist eine CAD-Software programmiert in Python. Die erste Version kam im Dezember 2002 heraus. Seitdem gab es zahlreiche weitere Releases.

<sup>73</sup> <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

<sup>74</sup> <http://www.ansys.com/>

<sup>75</sup> <http://www.cadence.com/>

<sup>76</sup> <http://www.systematics.co.il/English/Solutions.html>

<sup>77</sup> <http://www.autodesk.de/>

<sup>78</sup> <http://brlcad.org/>

<sup>79</sup> <http://www.pythoncad.org/>

- **XCircuit (Open Source Software):**<sup>80</sup> Xcircuit ist ein Programm für die Elektrotechnik. Es dient zur Darstellung von qualitativ-hochwertiger, schematischer Diagramme von elektronischen Schaltkreisen.
- **gEDA (Open Source Software):**<sup>81</sup> gEDA ist eine Konstruktionssoftware für den elektrotechnischen Bereich. Sie umfasst Funktionen zur Entwicklung elektronischer Schaltkreise, Simulationen, Prototyping und Produktion. gEDA läuft auf Unix-Basis.

#### 4.4.5 Referenzverwaltung

- **Endnote (kommerzielle Software):**<sup>82</sup> Literaturquellenverwaltung mit Abgleich bibliografischer Online-Datenbanken.
- **Reference Manager (kommerzielle Software):**<sup>83</sup> Literaturquellenverwaltung mit Abgleich bibliografischer Online-Datenbanken.
- **Litw<sup>3</sup> (Open Source Software):**<sup>84</sup> litw3 macht Literaturangaben für die wissenschaftliche Recherche im Internet weltweit verfügbar (von der Instituts- und Museumsbibliothek über Seminar-Literatur bis hin zur individuellen Publikationsliste). Es wird seit März 2002 bereits von mehreren wissenschaftlichen Einrichtungen erfolgreich genutzt. Ein ausgefeiltes Rechte-System regelt über individuelle Passwörter den Zugriff. litw3 ist daher ein brauchbares Tool für Forschungs-Teams oder DozentInnen an Universitäten, für Seminare und StudentInnen, die über eine Internet-Plattform gemeinsam und ortsunabhängig eine Literaturdatenbank erarbeiten wollen.
- **LiteRate (Freeware):**<sup>85</sup> Literaturverwaltung für Einzelplatz-PC mit Möglichkeiten zum Austausch von Literaturdaten. LiteRate ist eine sehr erfolgreiche nicht kommerzielle Literaturverwaltung und mittlerweile schon einige Zeit erhältlich.
- **JabRef (Open Source Software):**<sup>86</sup> JabRef ist ein grafisches Front End zur Verwaltung von Literaturquellen in BibTeX Datenbanken, einem standardisierten LaTeX Format für die Darstellung und Verwaltung von Literaturquellen.

<sup>80</sup> <http://xcircuit.ece.jhu.edu/>

<sup>81</sup> <http://www.geda.seul.org/>

<sup>82</sup> <http://www.endnote.com/>

<sup>83</sup> <http://www.isiresearchsoft.com/>

<sup>84</sup> <http://litw3.uni-muenster.de/litw3/>

<sup>85</sup> <http://www.literate.net/>

<sup>86</sup> <http://jabref.sourceforge.net/>

- **Wikindx (Open Source Software):**<sup>87</sup> Literaturquellenverwaltung für Einzelplatz-PC sowie zum Einsatz mit mehreren Benutzern.
- **Bibliograph (Open Source Software):**<sup>88</sup> Bibliograph ist eine serverbasierte (PHP, MySQL) Literaturquellenverwaltung und benutzt Mozilla bzw. Firefox als Front End. Das Programm ist (vergleichsweise) noch wenig ausgereift und die Demoversion ist sehr langsam. Bibliograph ist interessant für Einzelplatz-PCs sowie für mehrere Benutzer, die gemeinsam eine bibliografische Datenbank aufbauen wollen.
- **synapsen (Shareware):**<sup>89</sup> Elektronische Literaturverwaltung auf Java-Basis. synapsen ist eine Literaturverwaltung, die sich von den handelsüblichen Programmen wie ask-sam, Lidos, LiMan, EndNotes etc. in spezifischer Weise unterscheidet. Denn über die Verwaltung von bibliografischen Daten hinaus bietet synapsen eine Informations-Architektur, die dem Zettelkasten in spezifischer Weise selbst die Rolle eines Autors zuschreibt. Jeder Datensatz, der die bibliografischen Daten eines Textes ebenso erfasst wie einen mitunter sehr umfangreichen Lektürebericht, wird durch eine Liste von Schlagworten charakterisiert. Diese sind vom Anwender bei der jeweiligen Eingabe zu vergeben. Anhand umfangreicher interner Vergleiche fügt synapsen daraufhin eine Liste der Datensätze/Zettel an, die ebenfalls mit diesen Schlagworten belegt sind. Jeder Zettel schreibt sich damit automatisch in ein Netzwerk des persönlichen Wissens ein, das der Benutzer auf leichte Weise per hypertextuellem Mausklick verfolgen kann. Der Zettelkasten liefert auf diese Weise überraschende Verbindungen und assoziiert neue Argumentationslinien über Begriffe und die dazugehörigen Texte/Stellen, die vom Benutzer unter Umständen gar nicht gesehen, geahnt oder vergessen wurden.

#### 4.4.6 Einsatzerfahrungen

Der Reifegrad von freier Software im Bereich Wissenschaftlich-Technische Software wurde wie folgt bewertet:

- Funktionalität: 3,6
- Benutzerfreundlichkeit: 3,4
- Zuverlässigkeit, Sicherheit: 3,5
- Leistungsfähigkeit: 3,7
- Support: 3,2

Insgesamt wird Open Source Software in diesem Bereich somit eine recht hohe Ausgereiftheit bescheinigt.

<sup>87</sup> <http://wikindx.sourceforge.net/>

<sup>88</sup> <http://bibliograph.panya.de/>

<sup>89</sup> <http://www.verzetteln.de/synapsen/synapsen.html>

## 4.5 Groupware

Im weitesten Sinne ist Groupware definiert als Software zur Kollaboration von Individuen. In der Regel wird Groupware eingesetzt, um gemeinsam Daten bearbeiten zu können. Im engeren Sinn ist Groupware weitgehend ein Synonym für Personal Information Management (PIM) Software. PIM Software verwaltet üblicherweise:

- E-Mail
- Kontakte
- Termine
- Aufgaben
- Notizen

PIM Software kann sowohl als Einzelsystem als auch als verteiltes System eingesetzt werden. Erst der Einsatz als verteiltes System macht PIM Software zu Groupware. Es existieren integrierte Groupware Lösungen, die alle oder viele der genannten Funktionen in einem Produkt zusammenfassen, wie z.B. Microsoft Exchange oder Lotus Notes. Teilweise werden aber auch mehrere an sich unabhängige Softwarekomponenten zu einer Groupware Lösung kombiniert, z.B. ein Mailserver mit einer Kalenderlösung und einer Adressdatenbank.

### 4.5.1 Merkmale von Groupware

#### Funktionalität

Neben dem Senden und Empfangen von E-Mail beherrschen moderne Mail Clients in der Regel noch weitergehende Funktionalitäten. Dazu gehören insbesondere

- Funktionen zur Verwaltung mehrerer E-Mail Konten
- Adressbuchverwaltung
- Sicherheitsfunktionen (Verschlüsselung, Signaturen)
- Sortier- und Filterfunktionen
- Spamfilter
- Rechtschreibprüfung
- Layoutfunktionen (RTF oder HTML)

Wichtige Funktionen einer Kontaktverwaltung sind

- Anlegen und Editieren von Datenfeldern
- Suchfunktionalitäten und Filter
- Anlegen von Kommentaren
- Anlegen von Stichwörtern
- Verwaltung von Benutzerrechten

Eine Kalenderanwendung umfasst insbesondere

- Anlegen von Einzel- und Serienterminen
- Gruppenterminplanung / Terminkoordination
- Verwaltung von Ressourcen
- Verwaltung von Aufgaben
- Vergabe von Benutzerrechten

Die einzelnen Komponenten sind bei Groupware Anwendungen in unterschiedlichem Ausmaß integriert. Oft wird das Versenden von E-Mail aus der Kontaktverwaltung oder der Kalenderanwendung heraus benötigt, teilweise an mehrere Kontakte gleichzeitig (z.B. Besprechungsplanung oder Mailingaktion). Weiterhin ist die Möglichkeit einer Synchronisation mit PDAs von Wichtigkeit.

Die Gruppenfunktionalitäten werden in der Regel über eine Groupware Server Komponente bereitgestellt. Aus diesem Grund werden Client- und Serverkomponente hier zusammen bewertet.

### **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit einer Groupware Anwendung kann analog zu einem Betriebssystem in den administrativen Bereich und den Anwendungsbereich unterteilt werden. Unter den administrativen Bereich fallen Aufgaben, die mit der Einrichtung, Wartung und Verwaltung zu tun haben. Dazu gehören beispielsweise

- Grundkonfiguration des Servers
- Einrichtung und Verwaltung von Benutzern und Benutzerrechten (dazu gehört auch das Einrichten von E-Mail Konten)
- Einrichtung von Systemrichtlinien
- Einrichtung von Verwaltung von Ressourcen

Für Anwender ist insbesondere die grafische Benutzeroberfläche und die möglichst weitgehende intuitive Bedienbarkeit der Anwendung von Wichtigkeit.

### **Zuverlässigkeit / Sicherheit**

Eine Groupware Anwendung ist oft eine der wichtigsten Systeme einer Organisation, die bei Ausfall eine Unterbrechung von Arbeitsprozessen nach sich ziehen kann. Die Stabilität einer Groupware Anwendung ist daher eines der zentralen Kriterien. Des Weiteren muss die Datenintegrität gewährleistet sein. Daten dürfen keinesfalls verloren gehen, weder im Speicherzustand (Datenbank) noch während des Transits (E-Mail).

Eine grundlegende Sicherheitsanforderung ist, dass nur berechtigte Benutzer Zugang zu den Daten haben dürfen. Daher müssen wirksame Mechanismen für die Zugriffskontrolle und Authentifikation von Benutzern vorhanden sein. Diese stellen auch die Vertraulichkeit der abgelegten Daten sicher.

Eine weitere Forderung ist der Schutz vor Sabotage. Dies beinhaltet eine Resistenz gegenüber böswilligen oder irrtümlichen Angriffsversuchen und Schadprogrammen (wie Viren und Würmer). Auch sollten Sicherheitslücken in der Software nicht zu Angriffen gegen die gesamte IT Infrastruktur genutzt werden können (z.B. durch Rechteerhöhung).

### **Leistungsfähigkeit**

Die Leistungsfähigkeit einer Groupwarelösung definiert sich zum einen in der Ablaufgeschwindigkeit der Anwendung, zum anderen über die Skalierbarkeit auf eine große Anzahl von Benutzern und Daten.

### **Support**

Unter Support fallen die Art und der Umfang der angebotenen Dienstleistungen für die Software sowie Art und Umfang des vorhandenen Know-how.

## **4.5.2 Gegenüberstellung von Groupware**

### **Microsoft Exchange Server / MS Outlook**

Auch im Bereich der Groupware ist das Produkt aus dem Hause Microsoft Marktführer. Das Unternehmen gibt den Marktanteil seines Groupware Servers mit ca. 60% an<sup>90</sup> – mit steigender Tendenz. Die Client Komponente Outlook wird mit Microsoft Office ausgeliefert und verfügt somit über ein sogar noch höheres Marktpotenzial.

MS Exchange verfügt über E-Mail, Adressverwaltungs- und Kalenderfunktionen und deckt damit alle geforderten Gebiete ab. Die E-Mail Komponente ist sehr mächtig und genügt den Anforderungen ebenso wie die Kalenderfunktion. Die Adressverwaltung kann jedoch nicht mit fortgeschrittenen Anforderungen mithalten und eignet sich nur für einfache Szenarien, in denen nur einige wenige Adressen verwaltet werden müssen. Fortgeschrittene Suchfunktionalitäten fehlen ebenso wie ein zufrieden stellender Umgang mit Schlag- und Stichwörtern.

<sup>90</sup> Vgl. [http://www.microsoft.com/austria/windowsserversystem/1jahr\\_exchangeserver2003.msp](http://www.microsoft.com/austria/windowsserversystem/1jahr_exchangeserver2003.msp)

Die Administration eines Exchange Servers ist sehr komplex und erfordert vergleichsweise umfangreiches Fachwissen. Teilweise ist die Verwaltung des Exchange Servers sehr stark mit der Verwaltung von Windows verwoben, so dass Konfigurationen an den unterschiedlichsten Stellen durchgeführt werden müssen. Die Administrationskomponente ist hochgradig komplex und bedient sich nicht immer intuitiv. Die Bedienung der Clientkomponente gestaltet sich im Gegensatz dazu relativ stringent – wobei aber mit der neueren Version Outlook 2003 ein Komplexitätssprung auch auf der Clientseite zu bemerken ist.

Der Exchange Server arbeitet zuverlässig. Neuere Versionen fallen nur selten durch Ausfälle auf und es existieren fortgeschrittene Mechanismen zur Sicherstellung von Datenintegrität. Mit den vorhandenen Sicherheitsmechanismen lassen sich Benutzerrechte sehr granular regeln. Der Outlook Client fällt hingegen immer wieder als gravierendes Sicherheitsloch negativ auf, über das sich Angreifer leicht Zugang zum System verschaffen können. Die unnötig mächtige und unzureichend abgesicherte Skriptsprache in Outlook ist zudem ein willkommenes Werkzeug für die Programmierer von Schadsoftware.

Die Leistungsfähigkeit des Exchange Servers ist gut. Er skaliert auch in sehr großen Umgebungen zufrieden stellend und kann mehrere tausend Mailboxen pro Server verwalten, so dass die gesamte Fraunhofer-Gesellschaft theoretisch mit nur ca. 3-5 Exchange Servern betrieben werden könnte.

Die Unterstützung durch Dienstleister ist ein klarer Pluspunkt des Exchange Servers. Ebenso ist sehr viel Know-how sowohl auf der Administrations- als auch der Anwenderseite vorhanden.

### **Lotus Notes**

Das Produkt Lotus Notes wurde mit der Akquisition der Firma Lotus durch IBM übernommen. Notes ist ein äußerst mächtiges Produkt, das neben den genannten Groupware Funktionalitäten auch über eine Schnittstelle zur Programmierung eigener Groupware Applikationen verfügt. Auf dem Markt werden derartige Anwendungen auch käuflich angeboten. Die Server Komponente (Domino) läuft auf unterschiedlichen Betriebssystemen (darunter Windows und Linux), der Client ist nur für Windows und Mac OS verfügbar. Auf die Groupware Applikationen kann auch über einen Web Browser zugegriffen werden. Lotus Notes enthält auch Collaboration Funktionalitäten für die Zusammenarbeit in Teams (z.B. gemeinsames Arbeiten an Dokumenten).

Die benötigten Groupware Funktionalitäten sind alle in Lotus Notes enthalten. Auch hier sollte nicht unerwähnt bleiben, dass die integrierte Kontaktverwaltung für komplexe Aufgaben nicht ausreichen wird.

Die Administration eines Domino Servers ist als ähnlich komplex wie bei MS Exchange einzuschätzen. Allerdings fehlt die weitgehende Integration in das Betriebssystem, so dass sich der Domino Server insgesamt etwas übersichtlicher gestaltet. In der Benutzerfreundlichkeit des Clients liegen beide Produkte in etwa gleich auf.

Die Zuverlässigkeit der Notes Lösung ist sehr hoch. Systemausfälle sind eher selten und Datenverluste können durch umfangreiche Sicherungsmechanismen vermieden werden. Derzeit ist Notes auch nicht in gleichem Umfang wie die Exchange/Outlook Kombination Ziel von Angreifern. Die Leistungsfähigkeit von Lotus Notes lässt einen Betrieb auch in extrem großen Umgebungen zu.

Dienstleistungen für Notes sind ausreichend vorhanden, auch wenn hier das Microsoft Produkt etwas vorne liegen dürfte. Anwender- und Administrations-Know-how ist ebenfalls gut verfügbar, allerdings etwas weniger verbreitet als für Exchange/Outlook.

### **OpenGroupware.org**

OpenGroupware.org<sup>91</sup> ist ein freier Groupware Server, der aus dem kommerziellen Produkt Skyrix hervorging, aber nun unter der GPL (Teile auch unter der LGPL) steht. OpenGroupware bietet Kontakt-, Kalender- und Aufgabenverwaltung sowie Unterstützung für Projekt- und Dokumentenmanagement. Ein E-Mail Server wurde jedoch nicht in das Produkt integriert<sup>92</sup> – wohl aber das Versenden von E-Mail aus der Groupware Anwendung heraus. Für Letzteres ist ein eigener Web E-Mail Client enthalten, das Produkt kann aber auch mit externen Mail Clients zusammenarbeiten. Insgesamt ist die Funktionsvielfalt zwar nicht ganz so ausgeprägt wie bei der kommerziellen Konkurrenz, aber die vorhandenen Funktionen sollten für die meisten anstehenden Aufgaben ausreichen. Die Aufgabenverwaltung geht teilweise sogar über die Funktionalität von Outlook und Notes hinaus.

Auf den OpenGroupware Server kann über ein Web Interface mittels eines herkömmlichen WWW Browsers zugegriffen werden. Die Verwendung anderer Groupware Clients wie Ximian Evolution ist ebenfalls möglich, für MS Outlook muss allerdings eine kostenpflichtige Zusatzkomponente verwendet werden.

Optisch macht die Benutzerführung einen einfachen aber funktionellen Eindruck. Die Bedienung ist insgesamt recht intuitiv erlernbar. Auch die Administration gestaltet sich aufgrund der nicht so stark ausgeprägten Komplexität des Produkts

<sup>91</sup> <http://www.opengroupware.org/>

<sup>92</sup> Die Entwickler empfehlen die Verwendung des freien Cyrus Mailservers als IMAP Server

eher einfach. Die Leistungsfähigkeit müsste für den Betrieb in der eher dezentral orientierten Struktur der Fraunhofer-Gesellschaft ausreichen. Vergleichsweise schwierig dürfte sich hier die Beschaffung von Dienstleistungen und Know-how gestalten, da die Verbreitung von OpenGroupware deutlich hinter die kommerziellen Konkurrenten zurückfällt.

### **PHPGroupware**

PHPGroupware<sup>93</sup> ist eine freie, auf Apache und PHP basierende Groupware Anwendung, die sich noch in der Entwicklung befindet. Die verwendete Lizenz ist die GPL. Wie bei OpenGroupware.org wurde auch hier auf eine Mailserver Komponente verzichtet. Ein Web E-Mail Client ist aber enthalten.

Die Funktionalität kann mit den anderen, hier betrachteten Groupware Produkten noch nicht mithalten. Die Kernfunktionen sind aber enthalten, so dass ein möglicher Einsatz von PHPGroupware letztendlich vom konkreten Einsatzszenario abhängt.

#### **4.5.3 Einsatzerfahrungen**

Der Reifegrad von freier Groupware wurde relativ zu den relevanten kommerziellen Produkten in diesem Bereich wie folgt bewertet:

- Funktionalität: 3,2
- Benutzerfreundlichkeit: 3,0
- Zuverlässigkeit, Sicherheit: 3,3
- Leistungsfähigkeit: 3,3
- Support: 2,8

Insgesamt wird Open Source Software in diesem Bereich somit eine mittlere Ausgereiftheit bescheinigt.

<sup>93</sup> <http://www.phpgroupware.org/>

#### 4.5.4 Bewertung und Empfehlung

Die folgende Tabelle fasst das Ergebnis der Gegenüberstellung zusammen:

Kriterium	MS Exchange/ Outlook	Lotus Domino/ Notes	OpenGroupware	PHPGroupware
<b>Lizenz</b>	Kommerziell	Kommerziell	OSS (GPL, LGPL)	OSS (GPL)
<b>Funktionalität E-Mail</b>	◐	◐	○	○
Bearbeiten/Layout von E-Mail	●	●	◐	◐
Verwaltung von Mailadressen	◐	◐	◐	◐
Verwaltung mehrerer Mail Konten	◐	◐	◐	◐
Mailverschlüsselung / Signaturen	◐	●	○	○
Sortier- und Filterfunktionen	◐	◐	○	○
Rechtschreibprüfung	◐	◐	○	○
Spamfilter	●	●	○	○
<b>Funktionalität Kontaktverwaltung</b>	◐	◐	◐	○
Anlegen / Editieren von Datenfeldern	◐	◐	◐	○
Suchfunktionen / Filter	◐	◐	◐	○
Kommentare	◐	◐	◐	◐
Stichwörter	○	◐	◐	◐
Rechteverwaltung	◐	◐	◐	○
<b>Funktionalität Kalender</b>	●	●	●	◐
Einzel- / Serienterminplanung	●	●	●	◐

Kriterium	MS Exchange/ Outlook	Lotus Domino/ Notes	OpenGroupware	PHPGroupware
Terminkoordination	●	●	●	◐
Ressourcenverwaltung	●	●	●	○
Aufgabenplanung	◐	◐	●	◐
Rechteverwaltung	●	●	◐	◐
<b>Benutzerfreundlichkeit</b>	◐	◐	●	●
Anwenderseite	◐	◐	●	●
Administration	○	◐	●	●
<b>Zuverlässigkeit / Sicherheit</b>	◐	◐	●	◐
Stabilität	◐	◐	◐	◐
Datenintegrität	●	●	●	◐
Schutz vor Angriffen	○	◐	●	●
<b>Leistungsfähigkeit</b>	●	●	◐	◐
Ablaufgeschwindigkeit	◐	●	●	◐
Skalierbarkeit	●	●	◐	○
<b>Support</b>	●	◐	○	○
Angebot von Dienstleistungen	●	◐	○	○
Vorhandenes Know-how	●	◐	○	○

Tabelle 21: Gegenüberstellung von Groupware  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung)

Die Entscheidung, welche Groupware zum Einsatz kommen soll, hängt letztendlich mit dem konkreten Einsatzszenario zusammen. Sofern lediglich E-Mail genutzt werden soll, ist der Einsatz komplexer Groupware in der Regel nicht sinnvoll. Hier kann mit einem spezialisierten E-Mail Server und stand alone Clients eine mindestens genauso leistungsfähige und mit weniger Aufwand zu realisierende Lösung gefunden werden. Die Open Source Welt stellt hier mit Sendmail oder Cyrus äußerst leistungsfähige Produkte zur Verfügung. Freie Mail Clients wie z.B. Mozilla Thunderbird brauchen sich hinter der Mail Funktionalität der etablierten Groupware Produkte ebenfalls sicher nicht zu verstecken.

Die Kontaktverwaltungen der Groupware Produkte lassen in der Regel so manche Wünsche unerfüllt. Auch hier existieren deutlich leistungsfähigere, spezialisierte Adressverwaltungen, deren Einsatz in Betracht gezogen werden kann. Jedenfalls ist der Wunsch nach einer Kontaktverwaltung noch kein hinreichender Grund für den Einsatz von Groupware.

Dies gilt auch für die letzte hier betrachtete Groupware Komponente, den Gruppenkalender. Auch hier existieren spezialisierte Tools, die diese Aufgabe verrichten können – auch als Open Source (z.B. PHProjekt).

Groupware Produkte sind insbesondere in großen Szenarien mit vielen Benutzern sinnvoll, oder wenn die integrativen Funktionen dieser Software genutzt werden sollen. Mit der Aufgabe, an mehrere Adressen Terminvorschläge über E-Mail zu versenden, sind die spezialisierten Tools eher überfordert, da sie - wenn überhaupt - eher lose zusammenarbeiten.

PHPGroupware eignet sich derzeit eher für den Einsatz im kleineren und mittleren Rahmen. Mit der Aufgabe, eine einheitliche Groupware Lösung für ein verteilt arbeitendes Großunternehmen bereitzustellen, wäre das Produkt im derzeitigen Entwicklungsstadium wohl eher überfordert. Für eine unkomplizierte und einfach zu bedienende Lösung in kleinerem Rahmen kann das Produkt aber durchaus in Betracht gezogen werden.

OpenGroupware.org ist eine durchaus mächtige Groupware Applikation, die auch in größeren Szenarien zum Einsatz gebracht werden kann. Für fortgeschrittene Anforderungen an den Mail Client kann der Einsatz eines spezialisierten Mail Clients in Verbindung mit diesem Produkt überlegt werden.

Die Komplexität der beiden kommerziellen Groupware Lösungen Microsoft Exchange sowie Lotus Notes machen diese Produkte für den Einsatz in kleineren und mittleren Umgebungen eher unattraktiv, da der Administrationsaufwand als relativ hoch anzusehen ist. Microsoft Exchange setzt weiterhin zwingend einen Windows Server mit Active Directory voraus, so dass sich dieses Produkt eher für reine Windows Umgebungen eignet. Lotus Notes ist die mit Abstand mächtigste Groupware Anwendung und eignet sich daher insbesondere für hochkomplexe Einsatzszenarien mit vielen Benutzern.

## 4.6 Datenbanken

Datenbanksoftware ermöglicht die Verwaltung von Daten. Dazu gehört insbesondere das Speichern und Abrufen von Daten unter verschiedenen Sichten. Häufig sind auch Benutzer- und Rechteverwaltung sowie Transaktionsmanagement Teil einer Datenbankanwendung. Die meisten eingesetzten Datenbanksysteme sind relationale Datenbanken, in denen Daten in Tabellen gehalten werden, die untereinander in Beziehungen stehen können. Zunehmend werden aber auch objektorientierte und XML Datenbanken eingesetzt. Betrachtet werden hier nur relationale Datenbanken.

Datenbanksoftware ist sehr heterogen. Office Pakete enthalten häufig auch eine Datenbankkomponente. Diese eignen sich vor allem zur Verwaltung kleinerer Datenmengen in einer Einzelplatzumgebung. Große Datenbanksysteme enthalten weitergehende Funktionalitäten für den Umgang mit großen Datenmengen oder vielen Benutzern. In der Fraunhofer-Gesellschaft werden Datenbanken für die unterschiedlichsten Einsatzzwecke benötigt.

### 4.6.1 Merkmale von Datenbanken

#### Funktionalität

Die Funktionalität von Datenbanken wird in erster Linie vom Grad der Unterstützung der Abfragesprache SQL bestimmt. Nicht alle Datenbanken unterstützen sämtliche vorgesehenen Funktionen. Des Weiteren sind die vorhandenen Funktionen für die Benutzer- und Rechteverwaltung von Wichtigkeit. Größere Datenbanken sollten über Mechanismen zur Sicherstellung der Datenintegrität verfügen. Dazu gehören die Unterstützung von Transaktionen nach dem ACID Prinzip sowie Mechanismen für die Erstellung von Backups. In großen Einsatzszenarien sind ferner Funktionalitäten wie Clustering und Replikation nötig. Schließlich ist auch die Art und Anzahl der angebotenen Schnittstellen (z.B. ODBC, JDBC) von Wichtigkeit (letzteres Kriterium wird aber von allen hier betrachteten Datenbanken zu einem sehr hohen Grad erfüllt, so dass im Weiteren nicht mehr darauf eingegangen wird).

#### Benutzerfreundlichkeit

Anwender kommen nur selten direkt mit der Datenbanksoftware in Berührung. Die jeweiligen Funktionen werden von der darauf beruhenden Applikation zur Verfügung gestellt. Die Benutzerfreundlichkeit bezieht sich bei Datenbanksystemen also insbesondere auf die Administration.

### **Zuverlässigkeit/Sicherheit**

Hier ist insbesondere die Ausfallsicherheit der Datenbankanwendung von Bedeutung, da viele Datenbanksysteme als »mission critical« einzustufen sind. Sicherheit ist sowohl in Bezug auf die Datenintegrität als auch in Bezug auf die Sicherheit des Systems vor Angreifern und unbefugten Zugriffen zu verstehen.

### **Leistungsfähigkeit**

Die Leistungsfähigkeit einer Datenbank bezieht sich insbesondere auf die Verarbeitungsgeschwindigkeit und die Skalierbarkeit des Systems in großen Einsatzszenarien.

### **Support**

Auch für Datenbanken wird die Güte des Supports in erster Linie durch die Unterstützung durch Dienstleister sowie in der Art und dem Umfang des vorhandenen Know-hows gemessen.

## **4.6.2 Gegenüberstellung von Datenbanksystemen**

### **Microsoft Access**

MS Access ist eine Desktopdatenbank die als Teil bestimmter Varianten von Microsoft Office vertrieben wird. Die Datenbank selbst wird im Unterschied zu den meisten größeren Datenbanksystemen in einer einzelnen Datei gehalten. MS Access unterstützt SQL, bietet aber keine Funktionen zur Unterstützung von vielen gleichzeitigen Benutzern. Dafür sind recht komfortable Werkzeuge für den Entwurf von Datenbanken vorhanden. Generell kann festgehalten werden, dass die Benutzerfreundlichkeit von Access als sehr hoch anzusehen ist. Durch den hohen Verbreitungsgrad von MS Office werden auf dem Markt auch viele Dienstleistungen (z.B. Schulungen) für dieses Produkt angeboten. Die Leistungsfähigkeit dieser Datenbank ist aber für größere Anwendungen definitiv nicht ausreichend.

### **Oracle**

Oracle ist der Marktführer im Datenbanksegment. Oftmals wird diesem System auch die Technologieführereigenschaft zugeschrieben. Es gibt wohl keine benötigte Funktionalität, die von Oracle nicht bereitgestellt wird. Oracle ist allerdings auch hochgradig komplex. Die Unterstützung durch Dienstleister ist sehr gut, ebenso wie die Leistungsfähigkeit dieses Systems. Know-how über diese Datenbank ist sehr verbreitet. Nicht zuletzt gilt Oracle als äußerst zuverlässig und sicher.

## Microsoft SQL Server

Der SQL Server wird von Microsoft als Datenbank für große Einsatzszenarien positioniert, in denen Access nicht mehr ausreicht. Es sind daher die meisten diesbezüglich benötigten Funktionalitäten vorhanden. Es stehen benutzerfreundliche Werkzeuge für die Administration zur Verfügung. Die Leistungsfähigkeit dürfte für die allermeisten Einsatzgebiete ausreichen. SQL Server arbeitet zuverlässig und ist relativ sicher. Leider arbeitet er, wie die meisten Microsoft Produkte, nur mit Windows Servern zusammen.

## MySQL

MySQL<sup>94</sup> ist die meisteingesetzte Open Source Datenbank und sowohl unter der GPL als auch unter einer kommerziellen Lizenz erhältlich. Sie zeichnet sich durch eine extrem hohe Performanz aus, welche allerdings teilweise durch Kompromisse bei der Funktionalität erkaufte wurde. Nicht alle SQL Features werden unterstützt und nicht alle enthaltenen Funktionen verhalten sich SQL standardgetreu. In neueren Versionen sind die wichtigsten Funktionalitäten (z.B. Transaktionen, Sub Selects, Stored Procedures) aber vorhanden. Trigger und Views fehlen nach wie vor noch. Die Administration kann entweder über eine textbasierte Kommandozeile oder auch über zusätzliche grafische Front Ends bewerkstelligt werden<sup>95</sup>. MySQL steht in Bezug auf Zuverlässigkeit und Sicherheit den kommerziellen Produkten nicht nach. Die Firma MySQL AB (die den Code der Datenbank pflegt und weiterentwickelt) bietet auch kommerziellen Support für MySQL an.

## PostgreSQL

PostgreSQL<sup>96</sup> ging aus einem Projekt der Universität Berkeley hervor und wurde später unter der BSD Lizenz freigegeben. PostgreSQL bietet die meisten benötigten Funktionalitäten mit Ausnahme der Replikation. Benutzerfreundliche Administrationswerkzeuge werden bereitgestellt. Die Leistungsfähigkeit der Datenbank wurde vor allem in neueren Versionen deutlich gesteigert. Der Verbreitungsgrad von PostgreSQL reicht nicht ganz an MySQL heran, trotzdem ist Know-how in durchaus ausreichender Menge vorhanden. Kommerzieller Support ist jedoch eher selten. Dafür arbeitet PostgreSQL zuverlässig und sicher.

<sup>94</sup> <http://www.mysql.com/>

<sup>95</sup> Beliebt ist beispielsweise PHPMyAdmin ([http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php))

<sup>96</sup> <http://www.postgresql.org/>

### **SAP DB / MaxDB**

Die aus ADABAS D hervorgegangene SAP DB wurde von SAP im Jahr 2000 unter die GPL gestellt. Wenig später wurde zusammen mit MySQL AB ein Kooperationsprojekt gestartet, um die nun in MaxDB<sup>97</sup> umbenannte Datenbank für den Unternehmenseinsatz zu positionieren.

MaxDB ist für den Betrieb in komplexen SAP Anwendungen ausgelegt und bietet eine dementsprechend entwickelte Palette an Funktionen und eine dafür ausgelegte Leistungsfähigkeit. Ein mächtiges und komfortables Administrations-tool ist verfügbar – allerdings derzeit leider nur für Windows. Support wird sowohl von SAP als auch von MySQL AB angeboten.

#### **4.6.3 Einsatzerfahrungen**

Der Reifegrad von freier Datenbanksoftware wurde wie folgt bewertet:

- Funktionalität: 4,0
- Benutzerfreundlichkeit: 3,7
- Zuverlässigkeit, Sicherheit: 4,4
- Leistungsfähigkeit: 4,0
- Support: 3,4

Insgesamt wird Open Source Datenbanksoftware somit eine sehr hohe Ausgereiftheit bescheinigt. Datenbanken gehören zum dem mit am höchsten bewerteten Bereich.

<sup>97</sup> <http://www.mysql.com/products/maxdb/>

#### 4.6.4 Bewertung und Empfehlung

Die folgende Tabelle fasst das Ergebnis der Gegenüberstellung zusammen.

Kriterium	MS Access	MS SQL Server	Oracle	PostgreSQL	MySQL	MaxDB
<b>Lizenz</b>	Kommerziell	Kommerziell	Kommerziell	OSS (BSD)	OSS (GPL)	OSS (GPL)
<b>Funktionalität</b>	●	●	●	●	◐	●
Transaktionen	+	+	+	+	+	+
Joins	+	+	+	+	+	+
SubSelects	+	+	+	+	+	+
Stored Procedures	+	+	+	+	+	+
Triggers	+	+	+	+	-	+
Views	+	+	+	+	-	+
<b>Benutzerfreundlichkeit</b>	●	◐	○	◐	◐	◐
<b>Zuverlässigkeit / Sicherheit</b>	○	◐	●	●	●	●
<b>Leistungsfähigkeit</b>	○	◐	●	◐	●	◐
<b>Support</b>	●	●	●	○	●	◐

Tabelle 22: Gegenüberstellung von Datenbanksoftware  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung; + = ist vorhanden)

Aufgrund der stark unterschiedlichen Funktionalitäten der Datenbanksysteme ist eine Empfehlung auch hier stark vom jeweiligen Einsatzszenario abhängig. Für einfache Adressverwaltungen sind die großen Datenbanksysteme sicherlich zu komplex. Hier ist der Anwender mit einer Desktopdatenbank gut bedient. Derzeit ist Microsoft mit dem Produkt Access noch relativ alleine in diesem Marktsegment, doch OpenOffice 2.0 enthält jetzt ebenfalls eine integrierte Datenbankkomponente (Base) und bietet mit dieser eine vergleichbare Funktionalität. Die Wahl der Datenbankkomponente wird daher in diesen Fällen stark mit dem eingesetzten Office Produkt zusammenhängen. Des Weiteren hat das

bereits fortgeschrittene Open Source Projekt »Rekall«<sup>98</sup> die Entwicklung einer Datenbankkomponente im Access Segment zum Ziel.

Für Webanwendungen ist die freie Datenbank MySQL in vielen Fällen die erste Wahl. Die hervorragende Performanz dieser Datenbank kann in diesem Bereich von keinem anderen Produkt erreicht werden. Gerade für Webapplikationen sind auch viele Zusatzkomponenten für MySQL erhältlich. Komplexere Anwendungen, die in MySQL nicht vorhandene Funktionalitäten benötigen, sind unter Umständen mit PostgreSQL gut bedient. Diese Datenbank bietet viele SQL Funktionalitäten, die in MySQL derzeit noch nicht implementiert sind. Die BSD Lizenzierung macht PostgreSQL auch für kommerzielle Einsätze geeignet. Für sehr komplexe Anwendungen ist unter Umständen MAX DB eine gute Wahl. Kaum eine Funktionalität wird von dieser Datenbank nicht unterstützt.

Insgesamt sind die freien Datenbanken als äußerst stark einzuschätzen. Für die meisten denkbaren Einsatzgebiete kann eine geeignete Open Source Datenbank gefunden werden. Die kommerziellen Datenbankserver sind letztendlich nur in Bezug auf den Einsatz in extrem großen, verteilten Umgebungen mit »mission critical« Charakter (derzeit noch) stärker. Auch bieten diese teilweise Funktionalitäten, die über einen reinen Datenbankserver hinausgehen (z.B. Unternehmensreporting). In den vielen Fällen, in denen die genannten Einschränkungen nicht zutreffen, kann der Einsatz einer Open Source Datenbank sehr empfohlen werden.

## 4.7 Entwicklungssysteme

Entwicklungssysteme sind Programmpakete, die grundlegende Werkzeuge zur Erstellung von Software umfassen. Nachdem diese Tools früher im Wesentlichen nur für die Implementierung genutzt wurden, können heute alle Software-Engineering-Phasen durch Programme unterstützt werden. Für die Phasen Analyse und Design gibt es beispielsweise kommerzielle Tools zur XML- oder UML-Modellierung. In der Test-Phase können nicht nur Funktionalitäten sondern auch Qualitätsaspekte wie Leistung und Wartbarkeit mit Hilfe von Software-Werkzeugen geprüft werden. Hier existiert bereits eine Vielzahl von beliebten Open Source Werkzeugen wie JUnit oder Checkstyle.

Für die Implementierungs-Phase ist das Angebot an Werkzeugen am größten. Neben unverzichtbaren Tools wie Compilern und Laufzeitumgebungen für verschiedene Programmiersprachen gibt es Werkzeuge, die die Entwicklung vereinfachen. Dazu gehören sowohl Anwendungen wie integrierte Entwicklungsumgebungen, die nur während der Implementierung benötigt werden, als

<sup>98</sup> <http://www.thekompany.com/projects/rekall/>

auch Software, die zur Laufzeit eingesetzt wird (siehe dazu auch Abschnitt 4.9 Application Server). Mit Rahmenwerken (Frameworks) kann gutes Design wieder verwendet werden und Komponenten beziehungsweise Bibliotheken erleichtern die Programmierung von besonderen Funktionalitäten (z.B. XML-Parser, Logging oder objekt-relationales Mapping). Die Open Source Community stellt hier viele ausgereifte Lösungen wie Struts, Spring, Xerces, Hibernate und Log4j zur Verfügung. Eine weitere Gruppe von Werkzeugen dient dazu, die Entwicklung im Team zu unterstützen. Häufig verwendete Open Source Programme sind CVS für die Versionsverwaltung oder Bugzilla zum Bugtracking.

Diese Studie fokussiert sich auf die wichtigsten Basiswerkzeuge für die Implementierungs-Phase. Dies umfasst die bereits genannten Compiler und Laufzeitumgebungen sowie integrierte Entwicklungsumgebungen (IDE - Integrated Development Environment), die andere Werkzeuge zusammenfassen und leichter bedienbar machen. Die Verwendbarkeit dieser Werkzeuge ist wesentlich von der in einem Projekt eingesetzten Programmiersprache bestimmt. Die Auswahl einer Programmiersprache hängt von vielen Faktoren wie der zu implementierenden Software und deren Umgebung sowie den Kenntnissen und Vorlieben der Programmierer ab und ist nicht Gegenstand dieser Betrachtung.

#### 4.7.1 Merkmale von Entwicklungssystemen

##### Funktionalität

Die Notwendigkeit der im Folgenden aufgeführten Funktionalitäten und Werkzeuge hängt von der eingesetzten Programmiersprache ab. Die wichtigsten Basiswerkzeuge für die Implementierungs-Phase sind:

##### Editor

Prinzipiell kann jeder Texteditor zur Programmierung genutzt werden. Einige spezielle Funktionalitäten erleichtern jedoch die Arbeit. Die farbliche Markierung der Syntax verbessert die Übersicht. Quellcode-Formatierung sorgt für ein einheitliches Aussehen des Codes. Code-Vervollständigung schlägt passende Textbausteine vor. Refactoring-Unterstützung hilft beim Umbenennen und Umstrukturieren von Programmteilen. Die einfache Suche nach zusammenhängenden Code-Teilen wie aufrufende Methoden verbessert das Verständnis von fremdem Code.

##### Bibliotheken

Ausgereifte Standard-Bibliotheken für viele Anwendungskomponenten wie GUI, Internettechnologien, Datenbankanbindung, etc. verhelfen zu kürzeren Entwicklungszeiten mit qualitativ besseren Ergebnissen und leichter portierbaren Anwendungen.

### Compiler

Gute Compiler bieten unterschiedliche Optimierungsoptionen, um den generierten Code an die Anforderungen anpassen zu können. Mit einem Cross-Compiler ist es möglich, Maschinencode für andere Zielumgebungen zu erzeugen. Oft genügt der Compiler um Bibliotheken einzubinden, aber auf manchen Systemen ist ein separater Linker erforderlich. Bei der Erstellung von komplexen Softwaresystemen ist ein zusätzliches Build-Tool hilfreich, um nur die geänderten Programmteile neu zu übersetzen und alle Dateien in die Zielumgebung zu kopieren.

### Interpreter

Ein Interpreter beziehungsweise eine Virtuelle Maschine sollte Programme möglichst schnell und zuverlässig ausführen. Mit Just-in-time (JIT) Compilern kann die Abarbeitung von häufig genutzten Codeteilen zur Laufzeit beschleunigt und optimiert werden. Wenn eine automatische Speicherverwaltung (Garbage Collection) genutzt wird, sollte diese die Ausführung des Codes nicht merkbar blockieren.

### Debugger

Neben den Standardfunktionalitäten wie Breakpoints und der Anzeige von Variableninhalten erlauben manche Debugger die Änderung des Codes während einem Debugging-Lauf. Bei verteilten Anwendungen wird Remote-Debugging auf einem anderen Rechner benötigt.

### Integrierte Entwicklungsumgebung

Viele integrierte Entwicklungsumgebungen (IDE - Integrated Development Environment) unterstützen nur eine Programmiersprache. Es existieren aber auch Anwendungen, die mehrere spezielle IDEs unter einer gemeinsamen Benutzungsoberfläche zusammenfassen. Umfangreichere IDEs integrieren neben den oben genannten Basiswerkzeugen oft weitere hilfreiche Komponenten wie Code-Browser, GUIs zur Konfiguration von Komponenten oder die Möglichkeit der einfachen Erstellung von grafischen Benutzungsschnittstellen (Visual GUI Designer). Am flexibelsten ist eine IDE, wenn sie eine offene Schnittstelle für Plug-Ins enthält. So können weitere Werkzeuge mit zusätzlicher oder besserer Funktionalität bei Bedarf integriert werden.

Generell sollten die Werkzeuge Unicode unterstützen, um Anwendungen in verschiedenen Sprachen erstellen zu können. Wenn ein Entwicklungssystem auf vielen Plattformen verfügbar ist, können Entwickler bei unterschiedlichen Projekten mit der gewohnten Umgebung arbeiten. Der interpretierte Quellbeziehungsweise Bytecode einer Programmiersprache sollte möglichst ohne Modifikationen auf vielen Plattformen ausführbar sein

## **Benutzerfreundlichkeit**

Für Programmierer ist die Entwicklung am einfachsten, wenn alle Werkzeuge eine grafische Oberfläche (GUI) haben und integriert sind. Dies gilt insbesondere für Entwickler, die nur gelegentlich programmieren oder mehrere unterschiedliche Programmiersprachen nutzen und daher nicht alle notwendigen Kommandozeilenbefehle auswendig können.

Erfahrene Entwickler können mit Konsolen manchmal schneller arbeiten. Unter Unix sind diese Werkzeuge häufig schon vorkonfiguriert und müssen nicht erst installiert werden. Auch zur Automatisierung von Aufgaben wie dem Build-Prozess sind grafische Oberflächen weniger geeignet. Idealerweise bietet eine Entwicklungsumgebung also beide Möglichkeiten zur Bedienung an.

## **Zuverlässigkeit / Sicherheit**

Für einen einzelnen Programmierer sind höchste Zuverlässigkeit und Sicherheit der Werkzeuge während der Entwicklung nicht von entscheidender Bedeutung. Bei der Entwicklung im Team mit verteilten Standorten ist ein zuverlässiger Zugriff auf ein Code-Repository erforderlich. Wenn das Projekt kommerzielle Ziele verfolgt oder Geheimhaltung notwendig ist, sollte der Zugriff mit Hilfe von verschlüsselten Protokollen gemacht werden.

Laufzeitumgebungen sollten Programme zuverlässig ohne Absturz abarbeiten. Außerdem sollten sie mit Funktionen wie Zertifikatprüfungen und eingeschränkten Rechten für bestimmte Programme die Ausführung schädlicher Aktionen von fremdem Code verhindern.

## **Leistungsfähigkeit**

Rechenintensive Bearbeitungsschritte bei der Entwicklung wie die Compilierung sollten möglichst schnell ausgeführt werden. Für eine schnelle Abarbeitung des Programmes zur Laufzeit sind je nach Sprache optimierende Compiler, Just-in-time (JIT) Compiler, schnelle Interpreter bzw. Virtuelle Maschinen sowie Bibliotheken notwendig. Die Geschwindigkeit kann bis zu einem gewissen Grad durch Nutzung von mehreren Prozessoren oder größerem Speicher gesteigert werden. Die Hardwareanforderungen sollten jedoch besonders für die Laufzeitumgebung konfigurierbar und nicht zu hoch sein. Dies gilt vor allem bei der Software-Entwicklung für Embedded Systems und Konsumenten-Elektronikgeräte.

## **Support**

In der Regel sollten Programmiersprachen und Entwicklungsumgebungen gewählt werden, für die viel Know-how am Markt vorhanden ist. Es ist dann leichter, Programmierer für Entwicklung und Wartung zu finden. Die Qualität des Ergeb-

nisses und die Effizienz von erfahrenen Entwicklern sind deutlich höher. Bei der Lösung von Problemen sind entweder professionelle Dienstleister oder gute Literatur und Tipps von anderen Programmierern in Internet-Foren wichtig.

#### 4.7.2 Gegenüberstellung von Entwicklungssystemen

Die hier aufgeführten Entwicklungssysteme sind nur bedingt miteinander vergleichbar, da sie verschiedene Programmiersprachen und Werkzeuge wie Compiler und IDEs umfassen. Es ist auch möglich, unterschiedliche Kombinationen von diesen Tools zu verwenden.

##### Java

Java<sup>99</sup> ist eine noch relativ junge, objektorientierte Programmiersprache, die von Sun Microsystems 1994 veröffentlicht wurde. Sie wurde als Nachfolger von C++ entworfen und beseitigt mit Eigenschaften wie automatischer Garbage Collection und dem Verzicht auf Zeiger einige Probleme dieser Sprache. Ein weiterer wichtiger Grund für die Popularität von Java ist ihre Plattformunabhängigkeit, die durch dadurch erreicht wird, dass der Quellcode in einen Bytecode übersetzt wird, der auf einer virtuellen Maschine (JVM) ausgeführt werden kann. Eine JVM ist auf sehr vielen Plattformen und sogar als Chip verfügbar. Am Anfang wurde Java hauptsächlich dazu benutzt, so genannte Applets unter Verwendung eines Plugins in Web-Browsern auszuführen. Heute ist die Sprache in drei Varianten für unterschiedliche Zielumgebungen verfügbar: Desktops (J2SE), Server (J2EE) und Elektronikgeräte (J2ME). Neuere Java-Versionen erreichen mit Optimierungen wie einem Just-in-time (JIT) Compiler eine ähnliche Performanz wie C++. Es existieren auch diverse Compiler, die andere Sprachen wie Ada, Python oder Scheme in Java-Bytecode übersetzen.

Die Sun Java-Laufzeitumgebung (JRE) besteht aus der Virtuellen Maschine und den umfangreichen Standard-Klassenbibliotheken mit über 3000 Klassen. In der Entwicklungsumgebung (JDK / SDK) von Sun sind noch diverse Kommandozeilenprogramme wie Compiler, Debugger, CORBA-Nameservice, Appletviewer, JavaDoc-Generator, etc. enthalten. Neben Sun gibt es noch andere kommerzielle Anbieter von Java-Entwicklungsumgebungen wie BEA mit JRockit<sup>100</sup>. Java kann von Programmierern zwar kostenlos verwendet werden<sup>101</sup>, aber es gibt noch keine OSI-Open Source Lizenz für die Original-Software von Sun sondern nur die Java Research License (JRL)<sup>102</sup>. Für die wichtigsten Java-Werkzeuge wie Compiler, Virtuelle Maschine und Standard-Klassenbibliotheken existieren auch

<sup>99</sup> <http://java.sun.com>

<sup>100</sup> <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/jrockit>

<sup>101</sup> <http://java.sun.com/j2se/1.5.0/docs/relnotes/license.html>

<sup>102</sup> [http://java.sun.com/j2se/1.5.0/source\\_license.html](http://java.sun.com/j2se/1.5.0/source_license.html)

mehrere Open Source Implementierungen. Allerdings besitzt Sun das Trademark für Java, so dass die Open Source Versionen offiziell nicht Java heißen<sup>103</sup>.

GNU Classpath<sup>104</sup> ist ein Projekt der Free Software Foundation (FSF) und hat das Ziel die Standard-Klassenbibliotheken von Java zu implementieren. Da diese sehr umfangreich sind, gibt es noch mehrere Pakete wie Swing and AWT, die unvollständig implementiert sind. Classpath ist unter der LGPL veröffentlicht.

Das Kaffe Projekt<sup>105</sup> des Unternehmens Transvirtual Technologies umfasst eine JVM und Klassenbibliotheken und ist unter der GNU General Public License verfügbar. Kaffe ist noch in der Entwicklung und es fehlen noch viele Funktionalitäten wie beispielsweise ein Bytecode-Verifier zur vollen Kompatibilität mit der Java-Spezifikation. Die Klassenbibliotheken basieren auf GNU Classpath und haben eine hohe Kompatibilität zur Java Version 1.1, aber bei den neueren Versionen bis 1.5 gibt es noch größere Lücken. Kaffe wird mit vielen Linux und BSD-Distributionen ausgeliefert. Zusätzlich zu Suns Java kann Kaffe auch Erweiterungen von Microsoft wie J/Direct abarbeiten.

JamVM<sup>106</sup> ist eine Java Virtual Machine, die die JVM Spezifikation Version 2 erfüllt und sehr kompakt ist. Sie ist auf Unix-Systemen und unter der GPL nutzbar.

SableVM<sup>107</sup> ist eine weitere JVM, die Classpath Bibliotheken nutzt und unter der LPGL verfügbar ist. SableVM unterstützt die JVMDI und JDWP Schnittstellen und ermöglicht damit Debugging in IDEs wie Eclipse.

Jikes RVM (Research Virtual Machine)<sup>108</sup> ist eine JVM von IBM, die im Gegensatz zu den meisten anderen auch in Java geschrieben ist. Jikes RVM nutzt die GNU Classpath Bibliotheken. Jikes RVM implementiert noch nicht alle Funktionen der JVM-Spezifikation, aber selbst umfangreiche Anwendungen wie die IDE Eclipse laufen mit kleinen Einschränkungen darauf. Jikes RVM ist unter der Common Public License verfügbar und läuft auf unterschiedlichen Unix-Derivaten.

Jikes<sup>109</sup> ist ein schneller Java-Compiler, der Bytecode generiert. Er benötigt zusätzlich die Klassenbibliotheken einer Java-Laufzeitumgebung. Jikes ist unter

<sup>103</sup> <http://www.sun.com/policies/trademarks/>

<sup>104</sup> <http://www.gnu.org/software/classpath/>

<sup>105</sup> <http://www.kaffe.org>

<sup>106</sup> <http://jamvm.sourceforge.net/>

<sup>107</sup> <http://sablevm.org>

<sup>108</sup> <http://jikesvm.sourceforge.net>

<sup>109</sup> <http://jikes.sourceforge.net/>

der OSI-zertifizierten IBM's Public License verfügbar und wird in vielen Open Source Betriebssystem-Distributionen mitgeliefert..

GCJ<sup>110</sup> ist ein Java Compiler, der zur GNU Compiler Collection (GCC) gehört und den sprachunabhängigen Teil der GCC nutzt. Im Gegensatz zum Java-Compiler von Sun kann GCJ auch Java oder Bytecode in Maschinencode übersetzen. Er dient auch als Cross-Compiler für andere Zielumgebungen. Die Laufzeitumgebung libgcj enthält eine Virtuelle Maschine und Klassenbibliotheken. Damit kann Maschinencode gemischt mit Bytecode ausgeführt werden. Die Klassenbibliotheken von libgcj werden zurzeit mit denen von GNU Classpath zusammengeführt. Die Java APIs werden noch nicht vollständig unterstützt. Debugging von compiliertem Java-Code ist mit aktuellen Versionen des GNU Debuggers GDB<sup>111</sup> möglich.

## **GCC**

GCC<sup>112</sup> bedeutet offiziell GNU Compiler Collection und stand früher für GNU C Compiler. GCC ist der Standard-Compiler unter vielen Linux/Unix-Varianten. GCC besteht aus sprachabhängigen Bibliotheken und Compiler-Frontends, generischen Optimierungsfunktionen und zahlreichen maschinenspezifischen Backends. Direkt integriert sind die Sprachen C, C++, Objective-C, Objective-C++, Java, Fortran, und Ada. Weitere separate Frontends für die GCC existieren für Sprachen wie Pascal und COBOL. Die Compiler der GCC fallen unter die GPL. Für die Bibliotheken gilt die Runtime GPL<sup>113</sup> mit einer »runtime exception« oder »libgcc exception« genannten Ausnahme, die es erlaubt, die Bibliothek auch in proprietäre Software einzubinden.

Neben den Compilern existiert eine Vielzahl von Entwicklerwerkzeugen im GNU-Projekt von denen hier nur einige exemplarisch erwähnt werden. GNU Make<sup>114</sup> unterstützt den Build-Prozess. GNU Emacs<sup>115</sup> ist ein mächtiger programmierbarer Editor für viele Programmiersprachen, der auch für Windows verfügbar ist. Für den GNU Debugger GDB sind mehrere grafische Benutzungsoberflächen verfügbar.

<sup>110</sup> <http://gcc.gnu.org/java/>

<sup>111</sup> <http://sources.redhat.com/gdb/>

<sup>112</sup> <http://gcc.gnu.org/>

<sup>113</sup> [http://gcc.gnu.org/onlinedocs/libstdc++/17\\_intro/license.html](http://gcc.gnu.org/onlinedocs/libstdc++/17_intro/license.html)

<sup>114</sup> <http://www.gnu.org/software/make/make.html>

<sup>115</sup> <http://www.gnu.org/software/emacs/emacs.html>

## Eclipse

Eclipse<sup>116</sup> ist ein größtenteils auf Java basierendes Rahmenwerk zur Entwicklung von Desktop- Anwendungen und ist daher auf vielen Plattformen verfügbar. Das bekannteste Projekt auf Basis von Eclipse ist das Java Development Toolkit (JDT), eine integrierte Entwicklungsumgebung. Weitere Eclipse Projekte umfassen neben IDEs für C/C++ oder COBOL zahlreiche Entwicklungsaufgaben wie Modellierung und Test. Im Unterschied zu vielen anderen IDEs gibt es kaum eigene Basiswerkzeuge sondern Plug-Ins zur Integration von existierenden Tools. Ein eigener inkrementeller Java Builder ist ebenso nutzbar wie externe Builder (z.B. ANT). JDT integriert alle Java-Basiswerkzeuge und hat einen sehr guten Editor mit sämtlichen Funktionen wie Code-Formatierung und Refactoring. Der Debugger unterstützt auch die fortgeschrittenen Funktionen der JVM wie Remote-Debugging und dynamisches Laden von veränderten Klassen. Weiterhin sind zahlreiche Views mit Code-Browser und CVS Integration vorhanden. Eclipse beinhaltet eine eigene GUI-Bibliothek, das Standard-Widget-Toolkit (SWT), die auch zur Erstellung von Anwendungen verwendet werden kann. Anders als Java Swing nutzt SWT die grafischen Komponenten des zugrunde liegenden Betriebssystems. Das Visual Editor Project ermöglicht die visuelle Erstellung von GUIs mit Swing oder SWT. Die Unterstützung für die neueste Java Version 1.5 ist leider nur im aktuellen Build enthalten und nicht in der Release Version. Eine Vielzahl von externen Open-Source Projekten bietet weitere Plugins an, die aber oft noch nicht ausgereift sind.

Eclipse wurde ursprünglich von IBM entwickelt und dann unter der Eclipse Public License als Open Source veröffentlicht. Seit 2001 wird es von der Eclipse Foundation, einem unabhängigen Konsortium von bedeutenden Software-Anbietern verwaltet. IBM bietet auf Basis von Eclipse eine kommerzielle Entwicklungsumgebung mit zusätzlicher Integration in den IBM Application Server den WebSphere Studio Application Developer (WSAD)<sup>117</sup> bzw. den Nachfolger Rational Application Developer for WebSphere Software<sup>118</sup> an.

Neben Eclipse existieren noch weitere gute Open Source IDEs für Java wie NetBeans<sup>119</sup>. NetBeans ist unter der Sun Public License verfügbar und bildet die Basis für das kommerzielle Sun ONE Studio (früher Forte for Java).

<sup>116</sup> <http://www.eclipse.org/>

<sup>117</sup> <http://www-306.ibm.com/software/awdtools/studioappdev/>

<sup>118</sup> <http://www-306.ibm.com/software/awdtools/developer/application/index.html>

<sup>119</sup> <http://www.netbeans.org/index.html>

## Borland IDEs

Eine der ersten erfolgreichen Text-basierten IDEs war Turbo Pascal von Borland<sup>120</sup>. Der objektorientierte Nachfolger davon ist seit 1995 Delphi, das aus einer grafischen IDE, der Delphi Programmiersprache und den VCL/CLX (Visual Component Library, Component Library for Cross-platform development) Bibliotheken besteht. Der Delphi-Compiler ist schnell, optimiert und kann Maschinencode für mehrere Plattformen erzeugen. Es gibt zwar auch Open Source Compiler, die den Pascal-Dialekt Delphi mehr oder weniger gut verarbeiten können, aber Delphi wird allein von Borland kontrolliert. Delphi kann seit 2003 auch .NET Code generieren. Der Architekt von Turbo Pascal und Delphi wechselte 1996 zu Microsoft und entwickelte dort C# und das .NET Framework maßgeblich mit.

Ähnlich wie mit Turbo Pascal brachte Borland, das sich vorübergehend Inprise nannte, mit C++Builder eine Entwicklungsumgebung heraus, die den Produkten des Marktführers Microsoft überlegen waren. C++Builder beinhaltet ähnlich wie Delphi eine gute IDE, einen schnellen Compiler und die CLX-Bibliothek. Damit kann Code für Windows, Linux oder Solaris erzeugt werden. Unter dem Namen Kylix sind die IDEs für MS-Windows Delphi und C++Builder, jedoch nicht in den neusten Versionen, auch für Linux verfügbar.

Die Java-basierte IDE JBuilder ist der Marktführer unter den kommerziellen Anbietern und unterstützt die neusten Java-Versionen J2SE 5.0 und J2EE 1.4. JBuilder nutzt neben einem eigenen Compiler das JDK von Sun und kann für jedes Projekt unterschiedliche JDK-Versionen verwenden. JBuilder ist für MS-Windows, Linux, Solaris und MaxOS erhältlich und auch mit J2EE Application Servern anderer Hersteller integriert. Die JBuilder 2005 Foundation Version ist kostenlos nutzbar, enthält aber nicht alle wichtigen Funktionalitäten und Werkzeuge der kostenpflichtigen Versionen. Oracles JDeveloper basiert auf JBuilder.

Alle Borland IDEs sind benutzerfreundlich und bieten neben den geforderten viele weitere Funktionalitäten. Optional können auch Open Source Werkzeuge wie Editoren integriert werden. Neben den IDEs für diverse Programmiersprachen bietet Borland eine breite Palette an integrierten Entwicklungswerkzeugen zur Modellierung, zum Test sowie Application Server an. Seit kurzem ist Borland auch Board-Mitglied der Eclipse Foundation.

<sup>120</sup> <http://www.borland.com>

## .NET

Das .NET Framework<sup>121</sup> ist die Entwicklungsplattform von Microsoft seit 2002. Ähnlich wie bei Java gibt es eine Virtuelle Maschine mit Just-in-time Compiler (common language runtime - CLR) und das Pendant zum Bytecode ist die Common Intermediate Language (CIL). Neben der CIL definiert die Common Language Infrastructure (CLI) auch ein Typsystem und eine Basisklassenbibliothek. Im Framework enthaltene Sprachen sind Visual Basic .NET, Managed C++, J# und C#. Von anderen Anbietern gibt es über 40 weitere Sprachen, die für .NET geeignet sind. In Anwendungen können in CIL übersetzte Bibliotheken aus mehreren Sprachen gemischt werden. Ein Teil der Spezifikation von .NET ist als ECMA- Standard veröffentlicht. Einige Bibliotheken wie ASP.NET und ADO.NET sind jedoch nicht standardisiert und durch Patente geschützt.

Visual Studio .NET<sup>122</sup> ist die IDE von Microsoft für MS-Windows .NET Plattformen inklusive Windows CE für portable Geräte. Das Studio enthält Komponenten für die Sprachen Visual Basic, C++, C# und J#. Die IDE enthält alle wichtigen Funktionen wie einen Editor mit Syntax-Markierung, einen remote Debugger, einen visuellen GUI Designer, optimierende Compiler und diverse Bibliotheken. Die Basiswerkzeuge können auch über Kommandozeile genutzt werden. Die IDE kann mit Add-Ins erweitert werden.

Mono<sup>123</sup> ist ein Open Source Projekt mit Unterstützung von Novell, das eine Reihe von .NET kompatiblen Werkzeugen entwickelt. Dazu gehören ein C#-Compiler, eine Common Language Runtime und auch nicht standardisierte .NET Bibliotheken. Die .NET Spezifikation wird jedoch noch nicht vollständig erfüllt. Mit Mono sind wesentliche Teile von .NET unter Windows, Linux, verschiedenen UNIX-Derivaten sowie Mac OS X verfügbar. Der Compiler ist unter der GPL, die Runtime unter der LGPL und die Bibliotheken unter der MIT X11 Lizenz nutzbar. Auch proprietäre Lizenzen sind erhältlich.

Im Rahmen des DotGNU Projects<sup>124</sup> entsteht mit Portable.NET eine weitere Open Source Implementation von .NET für diverse Plattformen. Portable.NET enthält die CLR und den modularen csc Compiler, der bereits Frontends für C# und C beinhaltet. Frontends für andere Sprachen wie Java und Visual Basic sind in der Entwicklung. Der Compiler soll ähnlich wie der GCC unterschiedliche Backends für verschiedene Virtuelle Maschinen erhalten. Portable.NET ist unter der GPL mit einer Linking-Ausnahme für die Bibliotheken erhältlich. Die Bibliotheken sind noch nicht vollständig implementiert.

<sup>121</sup> <http://msdn.microsoft.com/netframework/>

<sup>122</sup> <http://msdn.microsoft.com/vstudio/>

<sup>123</sup> <http://www.mono-project.com>

<sup>124</sup> <http://www.dotgnu.org/>

IKVM.NET<sup>125</sup> ist eine Implementierung von Java für Mono und das Microsoft .NET Framework. Es enthält eine virtuelle Maschine und einen Compiler, der Java-Bytecode ins .NET Format (CIL) übersetzt. Die als Ziel definierte Kompatibilität mit Java 1.4 ist noch nicht erreicht. Die Klassenbibliotheken enthalten eine kompilierte Version von GNU Classpath. IKVM.NET nutzt die OSI-zertifizierte zlib/libpng License.

SharpDevelop<sup>126</sup> ist eine freie (GPL) integrierte Entwicklungsumgebung für C# und VB.NET, die auf Microsofts .NET Plattform aufsetzt. MonoDevelop<sup>127</sup> ist eine IDE für Mono in einem frühen Entwicklungsstadium, für die SharpDevelop auf Gtk# portiert wurde.

## Perl

Perl (Backronym: Practical Extraction and Report Language)<sup>128</sup> ist eine interpretierte Programmiersprache mit automatischer Speicherverwaltung und Unicode-Unterstützung. Der Perl-Interpreter parst den Quellcode nur einmal und arbeitet dann auf einem optimierten Syntax-Baum. Der Interpreter kann auch als Bibliothek in andere Programme integriert werden. Alternativ kann auch der Perl-Compiler verwendet werden, der C-Code erzeugt. Die Distribution enthält etwa 500 Module (Bibliotheken) und einen Debugger. Perl wird als praktische, aber unschöne Sprache bezeichnet, da es unterschiedliche Programmierparadigmen vermischt und schwer lesbar ist. Perl wird hauptsächlich für Textmanipulationen, Systemadministration und Webanwendungen mit Datenbank verwendet. Perl ist für die meisten Betriebssysteme und unter der Artistic License oder der GNU General Public License verfügbar. Seit einigen Jahren wird Parrot, eine Virtuelle Maschine mit Just-in-time Compiler für die kommende Perl Version 6 und andere Skriptsprachen, entwickelt.

Für Perl existieren freie IDEs wie PAGE (Perl Application Generator and Editor) oder Open Perl IDE, die aber beide nicht weiterentwickelt werden und nur für Windows verfügbar sind.

## PHP

Die Sprache PHP<sup>129</sup> (rekursives Reironym "PHP: Hypertext Preprocessor", vorher: "Personal Home Page Tools") wird bevorzugt für dynamische Webanwendungen genutzt. Sie ist die am häufigsten verwendete Skriptsprache für Webserver, da

<sup>125</sup> <http://www.ikvm.net/>

<sup>126</sup> <http://www.icsharpcode.net/OpenSource/SD/>

<sup>127</sup> <http://www.monodevelop.com/>

<sup>128</sup> <http://www.perl.org/>

<sup>129</sup> <http://www.php.net/>

sie leicht erlernbar ist. Häufig werden die Open Source Programme Linux, Apache, MySQL und PHP oder Perl (LAMP) gemeinsam eingesetzt. Die aktuelle PHP-Distribution enthält zahlreiche Bibliotheken sowie eine performante und zuverlässige Skripting-Engine. Einige Programmierer von PHP haben das Unternehmen Zend<sup>130</sup> gegründet und bieten Support und Produkte wie die IDE Zend Studio für PHP an. PHP ist für die wichtigsten Betriebssysteme verfügbar und unter der Open Source PHP License veröffentlicht. Ein Compiler ist nicht in der PHP-Distribution enthalten, aber von kommerziellen Anbietern wie Roadsend<sup>131</sup> erhältlich.

Neben Perl und PHP existieren weitere populäre und mächtige Open Source Skriptsprachen für alle wichtigen Plattformen. Python<sup>132</sup> ist eine interpretierte Sprache mit automatischem Speichermanagement, die mehrere Paradigmen wie objektorientierte oder funktionale Programmierung unterstützt. Mit Python und seiner umfangreichen Standardbibliothek werden auch komplexe Anwendungen wie der Zope Application Server oder die MoinMoin Wiki Engine entwickelt. Tcl/Tk (Tool Command Language / widget toolkit)<sup>133</sup> ist eine interpretierte Sprache mit GUI-Bibliothek, mit der Rapid Prototyping oder Rapid Application Development (RAD) von grafischen Anwendungen durchgeführt werden kann.

### 4.7.3 Einsatzerfahrungen

Der Reifegrad von Open Source Entwicklungssystemen wurde relativ zu relevanten kommerziellen Softwareprodukten und auf einer fünf-stufigen Skala bewertet. Der Reifegrad wurde separat für die fünf Indikatoren für Softwarequalität erhoben:

- Funktionalität: 4,1
- Benutzerfreundlichkeit: 3,7
- Zuverlässigkeit, Sicherheit: 4,3
- Leistungsfähigkeit: 4,1
- Support: 3,4

Damit liegen Entwicklungssysteme unter den am höchsten bewerteten Open Source Softwarekategorien.

<sup>130</sup> <http://www.zend.com/>

<sup>131</sup> <http://www.roadsend.com/home/index.php?SMC=1&pageID=compiler>

<sup>132</sup> <http://www.python.org/>

<sup>133</sup> <http://www.tcl.tk/>

#### 4.7.4 Bewertung und Empfehlung

Die Ergebnisse werden in folgender Tabelle zusammengefasst dargestellt:

Kriterium	Eclipse	Borland	MS Visual Studio.NET
<b>Lizenz</b>	OSS: EPL	Kommerziell	Kommerziell
<b>Funktionalität</b>	●	●	●
Editor	●	●	●
Bibliotheken	◐ / ●	●	●
Compiler	- / ●	●	●
Interpreter / Virtuelle Maschine	- / ●	- / ●	●
Debugger	●	●	●
Integrierte Entwicklungsumgebung	●	●	●
Plattformunabhängigkeit	●	●	○
Unterstützung mehrerer Programmiersprachen	◐	●	●
<b>Benutzerfreundlichkeit</b>	●	●	●
Grafische Benutzungsoberfläche	●	●	●
Bedienung über Kommandozeile	- / ●	●	●
<b>Zuverlässigkeit / Sicherheit</b>	●	●	●
Entwicklung im Team	●	●	●
Laufzeitumgebung	- / ●	- / ●	●
<b>Leistungsfähigkeit</b>	●	●	●
Geschwindigkeit	- / ●	●	●
<b>Support</b>	◐	◐	●
Unterstützung durch Dienstleister	◐	●	●
Vorhandenes Know-how	◐	◐	●

Kriterium	Perl	PHP	GCC	Sun Java
<b>Lizenz</b>	OSS: AL+GPL	OSS: PHPL	OSS: (R)GPL	JRL
<b>Funktionalität</b>	◐	◐	●	●
Editor	-	-	-	-
Bibliotheken	●	●	●	●
Compiler	◐	-	●	◐
Interpreter / Virtuelle Maschine	◐	◐	◐	●
Debugger	◐	○	- / ●	●
Integrierte Entwicklungsumgebung	-	-	-	-
Plattformunabhängigkeit	●	●	◐	●
Unterstützung mehrerer Programmiersprachen	○	○	●	◐
<b>Benutzerfreundlichkeit</b>	●	●	●	●
Grafische Benutzungsoberfläche	-	-	-	-
Bedienung über Kommandozeile	●	●	●	●
<b>Zuverlässigkeit / Sicherheit</b>	◐	◐	◐	●
Entwicklung im Team	-	-	-	-
Laufzeitumgebung	◐	◐	◐	●
<b>Leistungsfähigkeit</b>	◐	◐	●	●
Geschwindigkeit	◐	◐	●	●
<b>Support</b>	●	●	●	●
Unterstützung durch Dienstleister	●	●	●	●
Vorhandenes Know-how	●	●	●	●

Tabelle 23: Gegenüberstellung von Entwicklungssystemen

(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung; - = nicht vorhanden  
- / ● = Im Entwicklungssystem enthalten, andere Werkzeuge werden genutzt)

Bei Entwicklungssystemen sind Open Source Tools in vielen Bereichen die erste Wahl. Je nachdem für welches Betriebssystem und mit welcher Architektur die Programme entwickelt werden, gibt es jedoch Situationen in denen kommerzielle oder kostenlose, aber nicht den Open Source Kriterien entsprechende, Werkzeuge noch besser geeignet sind.

Die Verbreitung der Open Source Programmiersprachen Perl, PHP und GNU C spricht für sich. Für Webanwendungen auf den gängigen Betriebssystemen ist PHP zu empfehlen. Perl ist ein geeignetes Werkzeug für Administrationskripte. GNU C beziehungsweise C++ ist für die Entwicklung von Systemdiensten und schnellen Anwendungen unter allen Unix-Derivaten am besten geeignet. Unter MS Windows ist die GNU Compiler Collection jedoch nur unter Verwendung von Unix-Emulationen wie Cygwin<sup>134</sup> nutzbar. Für Programme, die nur auf MS-Windows laufen, ist das kommerzielle MS Visual Studio.NET mit den verschiedenen .Net Programmiersprachen wegen der Integration ins Betriebssystem am besten geeignet. Die freien .NET Implementationen sind zwar bereits nutzbar, aber unterstützen die Spezifikation noch nicht vollständig.

Komplexe Enterprise-Anwendungen in heterogenen Systemumgebungen oder Anwendungen, die plattformunabhängig sein sollen, können gut mit der Programmiersprache Java entwickelt werden. Die Open Source Basiswerkzeuge für die Sprache Java können jedoch noch nicht mit dem Java Development Kit von Sun konkurrieren. Open Source Java Laufzeitumgebungen sind derzeit nur auf manchen Unix-Betriebssystemen und für Anwendungen, die mit älteren Java-Versionen oder mit dem eingeschränkten Funktionsumfang auskommen, zu empfehlen.

Die integrierte Entwicklungsumgebung (IDE) Eclipse macht enorme Fortschritte und unterscheidet sich nur noch in der Zahl der mitgelieferten Werkzeuge von den besten kommerziellen IDEs wie dem JBuilder von Borland. Auch NetBeans ist eine empfehlenswerte Open Source IDE für Java. Neben diesen Basiswerkzeugen existieren auch viele Spezialprogramme zur Entwicklung, wie Rahmenwerke und Komponenten, bei denen Open Source Lösungen insbesondere für die Sprache Java erste Wahl sind.

## 4.8 Grafiksoftware

Grafiksoftware wird üblicherweise nach der Art der zu bearbeitenden Grafik in die beiden Unterbereiche Vektorgrafik- und Bitmapgrafiksoftware unterteilt. Vektorgrafikprogramme werden hauptsächlich für die Erstellung von Illustrationen (z.B. Broschüren und Flyer) genutzt, während Bitmapgrafiksoftware für die Bearbeitung von Bildern (z.B. Fotos) eingesetzt wird.

<sup>134</sup> <http://www.redhat.com/software/cygwin/index.html?id=home>

## 4.8.1 Merkmale von Grafiksoftware

### **Funktionalität**

Bitmap- und Vektorgrafikprogramme weisen fundamentale Unterschiede in den Funktionalitäten auf. Bitmapsoftware hat ihren funktionalen Schwerpunkt in der Manipulation von Bildern (z.B. Änderung von Farbwerten, Bildkorrekturen oder Effekte). Mit Texttools können Beschriftungen angebracht und bearbeitet werden. Des Weiteren werden von Bitmapgrafikprogrammen oft umfangreiche Funktionen zum Import von Bildern aus externen Quellen (wie Scanner oder Digitalkamera) angeboten. Für den professionellen Druck sollte die Farbtrennung nach dem CYMK Modell enthalten sein.

Wesentliche Merkmale von Vektorgrafikprogrammen sind im Zeichnen von Linien, Kurven und Polygonen zu sehen. Dabei kommen auch Aspekte aus der Konstruktion zum Einsatz (z.B. Bemaßung und Ausrichtung). Des Weiteren sind Werkzeuge für die Gestaltung von Farbverläufen und das Ausfüllen von Flächen von Wichtigkeit. Ebenso werden Funktionen für die Bearbeitung von Text (z.B. Beschriftungen) benötigt.

### **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit von Grafiksoftware wird hauptsächlich über die Gestaltung der Benutzeroberfläche definiert, wobei wesentliche Kriterien eine einfache Bedienung sowie eine stringente und nachvollziehbare Anordnung der Funktionen sind. Auch die leichte und intuitive Erlernbarkeit trägt zur Benutzerfreundlichkeit bei.

### **Zuverlässigkeit / Sicherheit**

Die Zuverlässigkeit wird bei Grafiksoftware hauptsächlich in der Stabilität der Anwendung gemessen. Sofern Makrosprachen zum Einsatz kommen, dürfen diese die Systemsicherheit nicht gefährden.

### **Leistungsfähigkeit**

Unter dieses Kriterium fallen die Ablaufgeschwindigkeit der Software sowie die Fähigkeit bezüglich des Umfangs mit großen und komplexen Bildern und Illustrationen.

### **Support**

Das Kriterium Support misst bei Grafiksoftware insbesondere die Unterstützung der Software durch Dienstleister (Angebot von Schulungen und Literatur), aber auch das vorhandene Know-how bei Anwendern.

## 4.8.2 Gegenüberstellung von Grafiksoftware

Das Gebiet der Grafiksoftware wird im Folgenden aufgrund seiner Heterogenität in die beiden Unterkategorien Bildbearbeitungsprogramme und Illustrationsprogramme unterteilt.

### 4.8.2.1 Bildbearbeitungsprogramme

#### Adobe Photoshop

Adobe Photoshop ist der Marktführer der Bildbearbeitungsprogramme. Die Software deckt die benötigten Funktionen ab. Kein anderes Grafikprogramm bietet eine derart große Fülle an Effekt- und Korrekturfiltern. Die Farbtrennung nach dem CYMK Schema ist vorhanden. Photoshop ist in der Verarbeitungsgeschwindigkeit aber teilweise schwerfällig. Photoshop Know-how ist zumindest bei professionellen Grafikern sehr verbreitet, allerdings bedient sich die Software nicht wirklich intuitiv, so dass bei Bedarf auf Schulungsmaßnahmen zurückgegriffen werden muss. Letztere sind am Markt ausreichend verfügbar.

#### Corel Photo Paint

Corel Photo Paint wird vor allem als Teil des Corel Draw Paketes vermarktet. Die Effekt- und Korrekturfilter sind fast ebenso mächtig wie bei Adobe Photoshop, das Textwerkzeug ist jedoch etwas weniger stark. CYMK Farbtrennung ist vorhanden. Die Verarbeitungsgeschwindigkeit entspricht weitgehend der von Photoshop. Corel Photo Paint ist insgesamt deutlich benutzerfreundlicher als das Adobe Produkt, jedoch nicht ganz so verbreitet. Dennoch ist ausreichend Know-how verfügbar, auch Gelegenheitsanwender finden sich in dem Programm schnell zurecht.

#### GIMP

Das freie Produkt GIMP<sup>135</sup> ist das bedeutendste Open Source Bildbearbeitungsprogramm. GIMP unterliegt der GPL und kann funktional mit den kommerziellen Produkten weitestgehend mithalten. Es verfügt allerdings derzeit noch nicht über die für die Druckvorstufe benötigte CYMK Farbseparation (diese wird mit einer der kommenden Versionen eingeführt werden). Die Bedienung gestaltet sich für an Photoshop gewöhnte Anwender etwas kontraintuitiv, ist aber durchaus als durchdacht zu bezeichnen. Die zunehmende Verbreitung von GIMP hat zu einer recht großen Menge an verfügbarer Dokumentation und auch angebotenen Schulungen geführt – wobei die kommerziellen Produkte diesbezüglich aber noch vorne liegen.

<sup>135</sup> <http://www.gimp.org>

### 4.8.2.2 Illustrationsprogramme

#### **Corel Draw**

Wie Photo Paint ist auch diese Software Teil des Corel Draw Pakets. Es bringt alle für die Erstellung professioneller Illustrationen benötigten Funktionen mit. Die Benutzerfreundlichkeit kann aber nicht mit dem Schwesterprodukt Photo Paint mithalten.

#### **Adobe Illustrator**

Auch dieses Produkt bietet alle benötigten Funktionalitäten. Die Benutzerfreundlichkeit lässt aber auch bei diesem Produkt einige Wünsche offen.

#### **Inkscape**

Inkscape<sup>136</sup> wird manchmal als das am weitesten fortgeschrittene Open Source Illustrationsprogramm bezeichnet. Es unterliegt der GPL und befindet sich noch in der Entwicklung. Funktional kann es daher derzeit noch nicht mit der kommerziellen Konkurrenz mithalten, aber zur Gestaltung von nicht allzu komplexen Broschüren und Flyern dürfte die Funktionalität bereits ausreichen. Das Produkt bedient sich sehr intuitiv und überzeugt durch eine recht hohe Verarbeitungsgeschwindigkeit. Der Support beschränkt sich derzeit noch im Wesentlichen auf die auf der Projektseite verfügbare Dokumentation sowie auf das Userforum.

<sup>136</sup> <http://www.inkscape.org/>

### 4.8.3 Einsatzerfahrungen

Der Reifegrad von freier Grafiksoftware wurde wie folgt bewertet:

- Funktionalität: 3,7
- Benutzerfreundlichkeit: 3,5
- Zuverlässigkeit, Sicherheit: 3,8
- Leistungsfähigkeit: 3,8
- Support: 3,1

Insgesamt wird Open Source Software in diesem Bereich somit eine überdurchschnittliche Ausgereiftheit bescheinigt.

### 4.8.4 Bewertung und Empfehlung

Open Source Software weist in den beiden Gebieten Bildbearbeitung und Illustration einen sehr unterschiedlichen Reifegrad auf. Mit GIMP existiert ein sehr mächtiges Werkzeug für die Bildbearbeitung. Funktional braucht sich GIMP wenig bis gar nicht hinter der kommerziellen Konkurrenz zu verstecken. Das einzige nennenswerte Manko ist die noch fehlende CYMK Farbseparation. Da diese in absehbarer Zeit dem Produkt hinzugefügt werden wird, gibt es (vielleicht abgesehen von einigen Spezialanforderungen) kaum noch einen Grund, GIMP nicht den Vorzug gegenüber den kommerziellen Bildbearbeitungsprogrammen zu geben. Für die allermeisten Einsatzzwecke sollte das Programm ausreichen.

Ein wenig zurückhaltender muss derzeit die Lage bei den Illustrationsprogrammen bewertet werden. Inkscape ist ein sehr viel versprechendes Projekt, befindet sich allerdings noch eher in der Entwicklungsphase. Es kann durchaus für einfachere Illustrationen bereits genutzt werden, allerdings wird man für komplexere Aufgaben derzeit noch auf die kommerziellen Produkte angewiesen sein. Interessierte Anwender sollten die Entwicklung von Inkscape jedoch im Auge behalten.

## 4.9 Application Server / Portalsoftware

Als Application Server bezeichnet man Softwarepakete, die zum Betrieb von Anwendungen mit einer mehrschichtigen Architektur (mit den Schichten Präsentation, Geschäftslogik und Datenhaltung) dienen. Die Pakete stellen anwendungsunabhängige Funktionalitäten wie eine dynamische Generierung von Webseiten oder Protokolle zur Kommunikation in verteilten Systemen bereit. Der Umfang dieser Basisfunktionalitäten der existierenden Softwareprodukte ist sehr unterschiedlich. Er reicht von einfachen Web-Anwendungen mit Anbindung an eine Datenbank bis zu hochskalierbaren, verteilten Unternehmensanwendungen. Viele dieser Produkte sind auf eine bestimmte

Softwareplattform wie die Java 2 Enterprise Edition (J2EE)<sup>137</sup>, Microsoft .NET<sup>138</sup> oder die Common Object Request Broker Architecture (CORBA)<sup>139</sup> spezialisiert. Da für diese Plattformen unterschiedliche Spezifikationen gelten, sind Application Server für verschiedene Plattformen nicht direkt vergleichbar. Für Microsoft .NET sind hauptsächlich proprietäre Softwareprodukte verfügbar. Spezielle CORBA-Produkte mit zahlreichen Service-Komponenten werden nur noch selten, insbesondere zur Integration von heterogenen Unternehmensanwendungen, eingesetzt. Basiskomponenten von CORBA wie ein Object Request Broker und das Protokoll IIOP werden auch von der Java-Plattform unterstützt. Deshalb wird im Folgenden nur auf Application Server für die J2EE-Plattform genauer eingegangen.

#### 4.9.1 Merkmale von Application Servern

##### Funktionalität

Die Java 2 Enterprise Edition (J2EE) ist eine Menge von Spezifikationen zur Entwicklung und dem Betrieb von mehrschichtigen Anwendungen auf der Basis der Java 2 Standard Edition (J2SE). Die wichtigsten Technologien der J2EE Plattform sind:

- JavaServerPages (JSP), Servlets
- Enterprise JavaBean Komponenten (EJB)
- Directory Service (JNDI)
- Message Service (JMS)
- XML- Remote Procedure Call (JAX-RPC)
- CORBA
- Transaction Service (JTA)
- JDBC Zugriff auf relationale Datenbanken

J2EE Application Server sollten diese Spezifikationen möglichst vollständig und in der aktuellen Version implementieren.

##### Benutzerfreundlichkeit

Die Bedienung umfasst die einfache Installation und Konfiguration des Application Servers sowie das Deployment und Management der Anwendungssoftware. Hierfür sind gute Werkzeuge, sinnvolle Voreinstellungen und eine gute Dokumentation notwendig.

<sup>137</sup> <http://java.sun.com/j2ee/index.jsp>

<sup>138</sup> <http://www.microsoft.com/net/>

<sup>139</sup> <http://www.corba.org/>

Zur intuitiven Bedienung sollte neben Kommandozeilen-Werkzeugen auch eine grafische Benutzungsoberfläche (GUI) vorhanden sein. Für die Entwicklung von Anwendungen sollte entweder ein spezielles GUI mitgeliefert werden oder eine Integration in häufig verwendete Entwicklungsumgebungen (IDEs) verfügbar sein. Die Administration der Application Server wird häufig über eine Webanwendung ermöglicht.

### **Zuverlässigkeit / Sicherheit**

Für Anwendungen mit Geschäftstransaktionen ist eine Verschlüsselung bei der Übertragung von sensiblen Daten unbedingt erforderlich. Die Konsistenz von Daten muss durch IT-Transaktionen sichergestellt werden. Viele Anwendungen setzen auch eine hohe Verfügbarkeit des Application Servers voraus. Neben einer hohen Stabilität sind dafür Funktionen wie Hot-Deployment notwendig.

### **Leistungsfähigkeit**

Die Leistungsfähigkeit ist eines der wichtigsten Kriterien von Application Servern, da bei vielen Unternehmensanwendungen eine sehr große Anzahl von Benutzern bedient werden muss. Für den einzelnen Anwender sind kurze Antwortzeiten der Anwendung wichtig. Eine hohe Skalierbarkeit muss durch Verteilbarkeit der Anwendung auf mehrere Rechner und durch Unterstützung von leistungsfähiger Hardware gewährleistet sein.

### **Support**

Das Kriterium Support misst bei Application Servern inwiefern die Software durch Dienstleister unterstützt wird. Auch das vorhandene Know-how bei Entwicklern und Informationen zur Problemlösung im Internet werden hier berücksichtigt.

## **4.9.2 Gegenüberstellung von Application Servern**

### **J2EE Application Server**

Die führenden kommerziellen J2EE Application Server sind der BEA WebLogic Server<sup>140</sup> und der IBM WebSphere Application Server<sup>141</sup>. Daneben existiert noch eine Vielzahl weiterer kommerzieller Java-Application Server von großen Anbietern wie Sun und Oracle sowie spezielle Application Server, die nur Teile der J2EE-Spezifikation unterstützen. Die Application Server von BEA und IBM sind ausgereift, performant und mit kompletten Entwicklungsumgebungen

<sup>140</sup> [http://de.bea.com/produkte/weblogic\\_server\\_81.jsp](http://de.bea.com/produkte/weblogic_server_81.jsp)

<sup>141</sup> <http://www-306.ibm.com/software/webservers/appserv/was/>

ausgestattet. Die Entwicklungsumgebung von IBM basiert auf der Open Source Software Eclipse.

Im Open Source Bereich dominiert der JBoss Application Server. Auch ObjectWeb JOnAS implementiert die komplette J2EE-Spezifikation. Sun hat die noch in der Entwicklung befindliche nächste Version 9 (alias GlassFish<sup>142</sup>) seines bisher kommerziellen Application Servers unter der neuen Open Source Lizenz CDDL veröffentlicht. Apache Geronimo<sup>143</sup> ist ein viel versprechender J2EE Server der Apache Software Foundation, der auf diversen Apache Projekten wie Tomcat basiert. Die J2EE 1.4 Zertifizierung von Geronimo ist nahezu abgeschlossen, jedoch sind bisher lediglich Beta-Versionen verfügbar. Andere Java-Application Server unterstützen nur die wichtigsten Funktionalitäten (z.B. Servlet-Engine).

### **JBoss**

Der JBoss Application Server<sup>144</sup> ist zwar ein Open Source Projekt, aber die wichtigsten Entwickler werden durch das Unternehmen JBoss Inc. finanziert. Dadurch ist JBoss Inc. in der Lage, Support durch die Entwickler anzubieten. Es gibt auch einige lokale Service Partner, die JBoss Support in Deutschland anbieten. JBoss unterhält im Rahmen der JBoss Enterprise Middleware System (JEMS) Architektur eine enge Kooperation und Integration mit verwandten Projekten wie dem Servlet-Container Tomcat und Hibernate als Objekt-Relationalen-Mapping (ORM) Werkzeug. In der JBoss/JIOP Distribution wird JacORB mitgeliefert, kann aber gegen andere CORBA-Implementierungen ausgetauscht werden. Mit JBoss Eclipse IDE steht ein Plug-In für die Entwicklungsumgebung Eclipse zur Verfügung. JBoss Inc. ist auch Mitglied der Eclipse Foundation. JBoss AOP ist ein Java-basiertes, aspekt-orientiertes Rahmenwerk, das sowohl integriert in den Application Server als auch separat verwendbar ist. Durch das AOP Model ist es bei JBoss möglich, POJOs (Plain Old Java Objects) Enterprise-Bean-Funktionalitäten hinzuzufügen. JBoss Portal ist der in Kürze erscheinende Nachfolger der JBoss Nukes Portalsoftware, die auf dem Application Server basiert.

### **Jonas**

JOnAS<sup>145</sup> ist ein Projekt der Open Source Initiative ObjectWeb. Daran sind große französische Konzerne wie Bull und France Telecom und das Forschungsinstitut INRIA beteiligt. JOnAS wird daher hauptsächlich in Frankreich eingesetzt und in zahlreichen produktiven Anwendungen genutzt. Aber auch der amerikanische

<sup>142</sup> <https://glassfish.dev.java.net/>

<sup>143</sup> <http://geronimo.apache.org/>

<sup>144</sup> <http://www.jboss.org/products/jbossas>

<sup>145</sup> <http://jonas.objectweb.org/index.html>

Red Hat Application Server für Linux basiert auf JOnAS. Außerdem gibt es auch deutschsprachige Unternehmen, die Support für JOnAS anbieten. Für die web-basierte Administration von JOnAS wird das Werkzeug JonasAdmin mitgeliefert. Für die Entwicklung mit JOnAS stehen mehrere Eclipse-Plugins aus eigenständigen Open Source Projekten wie Lombok oder JOPE zur Verfügung.

### **CORBA-Application Server**

Es existieren mehrere Open Source Projekte für CORBA-Application Server. Zu den bekanntesten gehören MICO CORBA<sup>146</sup> und JacORB<sup>147</sup>. Diese sind aber den führenden kommerziellen Produkten (z.B. Iona Orbix<sup>148</sup>) noch nicht ebenbürtig für die Implementierung von ausfallsicheren Enterprise-Systemen.

#### **4.9.3 Einsatzerfahrungen**

Der Reifegrad von Open Source Application Servern wurde relativ zu relevanten kommerziellen Softwareprodukten auf einer fünf-stufigen Skala bewertet. Der Reifegrad wurde separat für die fünf Indikatoren für Softwarequalität erhoben:

- Funktionalität: 3,6
- Benutzerfreundlichkeit: 3,1
- Zuverlässigkeit, Sicherheit: 3,6
- Leistungsfähigkeit: 3,5
- Support: 2,8

Dies entspricht einer überdurchschnittlichen Bewertung der relativen Reife von freien Application Servern.

<sup>146</sup> <http://www.mico.org/>

<sup>147</sup> <http://www.jacorb.org/>

<sup>148</sup> <http://www.ionac.com/products/orbix/welcome.htm>

#### 4.9.4 Bewertung und Empfehlung

Kriterium	BEA /IBM	JBoss	JOnAS
<b>Lizenz</b>	Kommerziell	OSS: LGPL	OSS: LGPL
<b>Funktionalität</b>	●	●	●
JavaServerPages, Servlets	●	●	●
Enterprise JavaBeans	●	●	●
Directory Service	●	●	●
Message Service	●	●	●
XML- RPC	●	●	●
CORBA	◐	◐	◐
Transaction Service	●	●	●
JDBC Datenzugriff	●	●	●
<b>Benutzerfreundlichkeit</b>	●	◐	◐
Bedienung	●	●	●
Grafische Benutzungsoberfläche	●	◐	◐
<b>Zuverlässigkeit / Sicherheit</b>	●	●	◐
Verschlüsselung	●	●	●
Datenkonsistenz	●	●	●
Verfügbarkeit	●	●	◐
<b>Leistungsfähigkeit</b>	●	●	◐
Skalierbarkeit	●	●	◐
<b>Support</b>	●	◐	○
Unterstützung durch Dienstleister	●	◐	○
Vorhandenes Know-how	●	◐	○

Tabelle 24: Gegenüberstellung von Application Servern  
(Legende: ● = starke Ausprägung; ◐ = mittlere Ausprägung; ○ = schwache Ausprägung; + = ist vorhanden; - = nicht vorhanden)

Die hier betrachteten Application Server implementieren alle die komplette J2EE-Spezifikation. Im Gegensatz zum IBM WebSphere Application Server ist der BEA WebLogic Server 9.0 mit Unterstützung der J2EE Version 1.4 allerdings nur in einer Beta-Version verfügbar. Die Open Source Application Server sind beide bereits zertifiziert für J2EE Version 1.4. Die CORBA-Unterstützung beschränkt sich auf die grundlegenden Funktionen und beinhaltet nicht alle CORBA-Services.

Die Eclipse-Plugins von JBoss und JOnAS bieten nicht die gleiche Integration wie die Entwicklungswerkzeuge von BEA und IBM. Die webbasierte Administration ist bei JBoss nur rudimentär umgesetzt.

Im produktiven Einsatz sind die kommerziellen Application Server von BEA und IBM noch weiter verbreitet. Daher sind die Unterstützung durch Dienstleister und das vorhandene Know-how noch etwas besser als im Open Source Bereich. Der JBoss Application Server holt jedoch auf, er wird oft zu Testzwecken und immer häufiger auch für den produktiven Einsatz genutzt. Durch JBoss Inc. und andere wird professioneller Support angeboten. JOnAS stammt aus Frankreich und wird in Deutschland seltener verwendet. JOnAS wird aber ebenfalls in großen Unternehmensanwendungen eingesetzt.

Zusammenfassend kann festgestellt werden, dass sich die Open Source Application Server nur noch in Details und bei höchsten Anforderungen von den kommerziellen Produkten unterscheiden. Für viele Anwendungen stellt insbesondere JBoss eine ernst zu nehmende Alternative dar.

#### 4.9.5 Portalsoftware

Ein Portal ist ein Anwendungssystem, das durch folgende Eigenschaften gekennzeichnet ist:

- Basierend auf Webtechnologien,
- Integration von Anwendungen, Prozessen und Diensten,
- Bereitstellung von Funktionen zur Personalisierung, Sicherheit, Suche und Präsentation von Informationen.

Ein Webportal bietet einen zentralen Zugriff auf personalisierte Inhalte und Anwendungen mit einmaliger Anmeldung am System (Single Sign On). Portalsoftware dient zur Erstellung und zum Betrieb von Portalen. Da Portale eine mehrschichtige Architektur haben, basiert Portalsoftware oft auf einem Application Server. Andere Produkte dagegen erweitern Content-Management-Systeme mit Funktionalitäten zur Enterprise Application Integration (EAI).

JBoss Nukes<sup>149</sup> ist eine Portalsoftware, die auf den JBoss Applicationserver portiert wurde. Das Projekt entstand, weil die Open-Source-Implementierungen PHPNuke und PostNuke nicht skalierbar genug waren. Nukes besteht aus einem Content Management System (CMS) mit Benutzerverwaltung, Sicherheitsfunktionen und Komponentenmanagement. Zusätzlich gibt es eine Vielzahl von Komponenten für Portale, die dem System sogar zur Laufzeit hinzugefügt werden können. JBoss Portal 2.0<sup>150</sup> ist der noch in der Entwicklung befindliche Nachfolger von JBoss Nukes. Er unterstützt die JSR-168 Portlet-Spezifikation zur einfachen Integration von standardkonformen Portal-Modulen. JBoss Portal 2.0 ist unter der LGPL verfügbar, Support wird von JBoss Inc. angeboten.

Apache Jetspeed<sup>151</sup> ist eine weitere Open-Source-Software, mit der Webportale erstellt werden können. Da Module wie Cocoon oder WebMacro integriert sind, lassen sich Portale mit XML/XSL-Technologien entwickeln. Weiterhin werden emplate-basierte Layouts inklusive Java Server Pages unterstützt. Um Portale auch auf mobilen Endgeräten anbieten zu können, wird die XML-basierte Sprache Wireless Markup Language unterstützt. Jetspeed bietet noch weitere Funktionalitäten wie Caching, Sicherheitsmechanismen, Synchronisation und Portlets.

Zope<sup>152</sup> ist ein Web Application Server, der zur Erstellung von Content-Management-Systemen und Portalen ausgelegt ist. Der Open Source Server ist in der Programmiersprache Python implementiert. Unterstützt werden unter anderem dynamische Inhalte, eine Suchmaschine, ein Rechtssystem und eine Historie der angelegten Objekte. Es steht eine Vielzahl von Zusatzmodulen wie neue Objekttypen, Konnektoren zu Datenbanken und anderen externen Quellen sowie Werkzeuge für E-Commerce-Applikationen zur Verfügung.

Die meisten Open Source Content Management Systeme, die zur Implementierung von Portalen verwendet werden, haben jedoch nicht den Funktionsumfang von kommerzieller Portalsoftware. Manche kommerzielle Anbieter wie eXo platform<sup>153</sup> bieten ihre Software zusätzlich auch unter der GPL an.

<sup>149</sup> <http://www.jboss.org/products/nukesjboss>

<sup>150</sup> <http://www.jboss.org/products/jbossportal>

<sup>151</sup> <http://portals.apache.org/jetspeed-1/>

<sup>152</sup> <http://www.zope.org/>

<sup>153</sup> <http://www.exoplatform.com/portal/faces/public/exo>

#### 4.10 Dokumenten Management Systeme (DMS)

Eine Studie des Verbands Organisations- und Informationssysteme e. V. (VOI)<sup>154</sup> hat festgestellt, dass die DMS-Lösungen IXOS von Open Text und EASYWARE der EASY SOFTWARE AG besonders häufig zum Einsatz kommen. Unter den annähernd 200 Anbietern von DMS-Lösungen<sup>155</sup> konnte ein Marktführer, der den Markt von Dokumenten Management Systeme dominiert, nicht identifiziert werden.

Dennoch haben sich bisher keine wirklich weit gediehenen Open Source Aktivitäten in diesem Bereich entwickelt. Dies mag damit zusammenhängen, dass mit den DMS unternehmenskritische Informationen genutzt werden und Unternehmen besonders vorsichtig beim Einsatz entsprechender Software vorgehen. Ein besonders wichtiger Aspekt ist dabei die Gewährleistung einer langfristigen Weiterentwicklung der DMS.

Der Reifegrad von freier Software in diesem Bereich wurde wie folgt bewertet:

- Funktionalität: 2,9
- Benutzerfreundlichkeit: 2,5
- Zuverlässigkeit, Sicherheit: 3,3
- Leistungsfähigkeit: 3,3
- Support: 2,7

Insgesamt wird Open Source Software in diesem Bereich somit eine unterdurchschnittliche Ausgereiftheit bescheinigt.

#### 4.11 Application Service Providing (ASP)

Unter Application Service Providing wird zunächst eine Dienstleistung verstanden, im Rahmen derer Software von einem Dienstleister installiert, gewartet und dem Kunden über Fernzugriffe zur Verfügung gestellt wird. Dazu ist bei nicht web-basierten Systemen eine Software erforderlich, die es den Kunden ermöglicht, auf extern installierte Programme zuzugreifen.

Marktführer für ASP Software ist die Firma Citrix. Freie Software, die Citrix funktional ersetzen könnte, existiert (insbesondere im Hinblick auf ASP als Dienstleistung) derzeit nicht.

<sup>154</sup> Vgl. <http://www.easy.de/content/easy/easy000841/DMS%20Marktuntersuchung%20Ergebnis%20v2.pdf>

<sup>155</sup> Vgl. <http://www.ancoso.com/products/document%20management/richtiges>

Sofern der volle Funktionsumfang der Citrix Lösung nicht benötigt wird, wird häufig der Microsoft Terminal Server eingesetzt. Damit lassen sich Applikationen auf einem zentralen System installieren und von dezentralen Clients aus nutzen. Der Terminal Server ist in Windows Server Systeme integriert, sofern er jedoch von mehreren Benutzern gleichzeitig verwendet werden soll, müssen zusätzliche Lizenzen erworben werden.

Ähnliche Funktionalitäten bietet auch VNC<sup>156</sup>. Die einfache Version von VNC ist unter der GPL als Open Source Software freigegeben, erweiterte Versionen sind hingegen lizenzpflichtig. Vorteil von VNC ist vor allem die Plattformunabhängigkeit.

Unter Unix Systemen stehen von vornherein relativ mächtige Tools zur Verfügung, mit denen Zugriffe auf entfernte Systeme realisiert werden können (z.B. SSH oder der X Server).

Welche Software konkret eingesetzt werden kann, hängt von den jeweiligen Anforderungen ab. Eine generelle Empfehlung kann an dieser Stelle nicht ausgesprochen werden. Für einen rein internen Gebrauch in Organisationen oder Instituten (der in der Fraunhofer-Gesellschaft wohl die Regel darstellt) dürften VNC oder die UNIX Bordmittel in vielen Fällen ausreichen.

Der Reifegrad von freier Application Providing Software wurde gemessen an der kommerziellen Konkurrenz wie folgt bewertet:

- Funktionalität: 2,9
- Benutzerfreundlichkeit: 2,9
- Zuverlässigkeit, Sicherheit: 3,2
- Leistungsfähigkeit: 3,3
- Support: 2,3

Insgesamt wird Open Source Software in diesem Bereich somit eine unterdurchschnittliche Ausgereiftheit bescheinigt.

## 4.12 Content Management Systeme

Unter Content Management Systemen sollen hier in erster Linie Web Content Management Systeme verstanden werden, die das Erstellen und die Pflege von Webinhalten zum Ziel haben.

<sup>156</sup> <http://www.realvnc.com/>

Zope<sup>157</sup> ist freie Software für Web Content Management, die unter der Zope Public License (ZPL) veröffentlicht wird. Die ZPL ist eine BSD ähnliche Lizenz, die von der FSF jedoch als kompatibel mit der GPL betrachtet wird. Typo3<sup>158</sup> ist ebenfalls Open Source Software, die verwendete Lizenz ist die GPL. Ein weiteres Open Source Produkt ist OpenCMS<sup>159</sup>. Zu den bekanntesten kommerziellen Content Management Systemen zählen RedDot und OpenText.

Es kann festgestellt werden, dass im Bereich Content Management Systeme die vorhandene freie Software sämtliche benötigte Funktionalitäten bietet. Mit allen Produkten können Webauftritte erstellt, verwaltet und gepflegt werden. In der Benutzerfreundlichkeit, Leistungsfähigkeit und Zuverlässigkeit sind die freien CMS Systeme ihren kommerziellen Gegenspielern mindestens ebenbürtig. Für Zope ist auch kommerzieller Support zu bekommen – sollte dieser benötigt werden.

Für den Einsatz von kommerzieller Software gibt es im Bereich Content Management Systeme insgesamt wohl selten einen zwingenden Grund.

Der Reifegrad von freier Software in diesem Bereich wurde wie folgt bewertet:

- Funktionalität: 3,4
- Benutzerfreundlichkeit: 2,9
- Zuverlässigkeit, Sicherheit: 3,3
- Leistungsfähigkeit: 3,2
- Support: 2,2

Insgesamt wird Open Source Software in diesem Bereich somit eine mittlere Ausgereiftheit bescheinigt.

#### 4.13 Cluster / Grid Software

Die Cluster / Grid Software steht zum großen Teil noch im Entwicklungsstadium. So werden Themen wie z.B. Sicherheit, verteiltes Datenmanagement, Management von virtuellen Organisationen, Gridservices und Ressourcenmanagement in diesem ganzheitlichen Ansatz als Open Source Software weiterentwickelt. Herausragende Open Source Lösungen sind hierbei GLOBUS<sup>160</sup> und Unicore<sup>161</sup>.

<sup>157</sup> <http://www.zope.org/>

<sup>158</sup> <http://www.typo3.com/>

<sup>159</sup> <http://www.opencms.org/opencms/en/>

<sup>160</sup> <http://www.globus.org/ogsa/>

<sup>161</sup> <http://unicore.sourceforge.net/>

Kommerzielle Softwareprodukte fokussieren vor allem auf Teilbereiche des Gridcomputing wie z.B. das Ressourcenmanagement und damit die Auslastung von Rechnersystemen. Oracle 10g<sup>162</sup> ist aktuell die kommerzielle Cluster / Grid Lösung mit dem umfassendsten Ansatz. Viele weitere kommerzielle Lösungen in diesem Bereich beschränken sich rein auf das Ressourcenmanagement.

Der Reifegrad von Open Source Software in diesem Bereich wurde in der Befragung wie folgt gesehen:

- Funktionalität: 3,4
- Benutzerfreundlichkeit: 2,9
- Zuverlässigkeit, Sicherheit: 3,6
- Leistungsfähigkeit: 4,0
- Support: 2,8

Insgesamt wird Open Source Software in diesem Bereich somit eine leicht überdurchschnittliche Ausgereiftheit bescheinigt.

#### 4.14 Customer Relationship Management

Über 150 Anbieter von CRM Software tummeln sich auf dem deutschen Markt. Die umsatzstärksten sind SAP und Siebel. Platz drei belegt ein Branchenspezialist: Die Kaiserslauterer CAS GmbH ist Marktführer in der Konsumgüterindustrie. Auf Platz vier liegt mit IET Solutions erneut ein sektorenübergreifender Anbieter<sup>163</sup>.

Laut Handelsblatt dominieren aktuell die folgenden Technologietrends:

- Datenintegration: Sie führt Kundendaten verschiedener Vertriebskanäle zusammen.
- Geschäftsprozesse: Das Management von Geschäftsprozessen verbindet CRM mit angrenzenden Bereichen - etwa dem Beschwerdemanagement und dem Rechnungswesen.
- Proaktives CRM: Dieses hilft, Kunden zur richtigen Zeit das passende Angebot zu machen.
- Order Management: Es ermöglicht dem Vertrieb, verbindliche Lieferzusagen zu treffen und gewährleistet, dass ein beworbenes Produkt vorrätig ist.
- Open Source: Stellt sicher, dass andere Anbieter die Software weiterentwickeln können.

<sup>162</sup> <http://www.oracle.com/database/index.html>

<sup>163</sup> Vgl. Handelsblatt Nr. 048 vom 09.03.05 Seite b06

Open Source Lösungen sind im Vergleich zu kommerziellen Produkten noch nicht oft im Einsatz. Die Zahl der Open Source CRM-Lösungen wächst jedoch beständig<sup>164</sup>. Anbieter wie Wice<sup>165</sup>, Crixp<sup>166</sup> oder Sugar CRM<sup>167</sup> punkten immer öfter gegen die etablierten Anbieter.

Der Reifegrad von freier Software im diesem Bereich wurde wie folgt bewertet:

- Funktionalität: 2,5
- Benutzerfreundlichkeit: 2,7
- Zuverlässigkeit, Sicherheit: 2,6
- Leistungsfähigkeit: 2,5
- Support: 2,2

Insgesamt wird Open Source Software in diesem Bereich somit eine stark unterdurchschnittliche Ausgereiftheit bescheinigt. CRM Software gehört zu den am schlechtesten bewerteten Softwarearten.

#### 4.15 Videosoftware und Rendering

Videosoftware wird vor allem zur Erstellung und Bearbeitung von Videodaten genutzt. Bekannteste Produkte sind Adobe Premiere, Pinnacle Studio sowie Apple Final Cut. Alternative freie Software ist in diesem Bereich derzeit nicht erhältlich – zumindest keine, die kommerzielle Videoschnittsoftware auch nur annähernd ersetzen könnte. Derzeit konzentrieren sich die Bemühungen der Open Source Programmierer eher darauf, erst einmal die Abspielsoftware für die bekanntesten Videoformate zu erstellen. Hier sind beispielsweise mit XINE<sup>168</sup> auch schon Fortschritte erzielt worden.

Auch gibt es freie Software, mit denen Teilbereiche der Videobearbeitung durchgeführt werden können, wie beispielsweise die Umwandlung von Formaten oder die Bearbeitung von Tonspuren.

Im Bereich 3D Modellierung und Rendering sieht die Lage jedoch anders aus. Dem kommerziellen Platzhirsch 3D Studio machen hier mit Blender<sup>169</sup> und PovRay<sup>170</sup> gleich zwei leistungsstarke Open Source Produkte das Leben schwer. Mit Crystal Space<sup>171</sup> steht eine freie 3D Engine zur Gestaltung dreidimensionaler

<sup>164</sup> Vgl. [http://www.ig4.de/ticker\\_meldungen/1104668614.html](http://www.ig4.de/ticker_meldungen/1104668614.html), vom 2.1.2005

<sup>165</sup> Vgl. <http://www.wice.de/>

<sup>166</sup> Vgl. <http://www.crixp.com/>

<sup>167</sup> Vgl. <http://www.sugarcrm.com/home/>

<sup>168</sup> <http://xinehq.de/>

<sup>169</sup> <http://www.blender.org/>

<sup>170</sup> <http://www.povray.org/>

<sup>171</sup> <http://www.crystalspace3d.org/>

Welten zur Verfügung. Während die kommerziellen Produkte auch hier technisch noch an der einen oder anderer Stelle überlegen sind, so erscheint es dennoch ratsam, die freien Produkte in die Entscheidung mit einzubeziehen. Für einige Einsätze dürften diese durchaus geeignet sein.

Der Reifegrad von freier Videosoftware wurde von den Befragten folgendermaßen gesehen:

- Funktionalität: 2,5
- Benutzerfreundlichkeit: 2,4
- Zuverlässigkeit, Sicherheit: 2,6
- Leistungsfähigkeit: 3,1
- Support: 2,1

Insgesamt wird Open Source Software in diesem Bereich somit eine stark unterdurchschnittliche Ausgereiftheit bescheinigt. Videosoftware gehört zu den am schlechtesten bewerteten Softwarearten.

#### 4.16 Enterprise Resource Planning (ERP) Software

Marktführer im Bereich der ERP Software ist SAP. Es gibt jedoch noch weitere Anbieter, die sich mit ihren Produkten teilweise an kleinere Unternehmen richten, oder aber ihre Software speziell für bestimmte Branchen anbieten.

Die Fraunhofer-Gesellschaft selbst setzt das speziell entwickelte SIGMA als ERP System ein.

Open Source Software hat in diesem Bereich derzeit noch nicht die funktionale Ausgereiftheit, um mit SAP oder Navision mithalten zu können. An kleinere und mittlere Unternehmen richtet sich das Projekt Compiere<sup>172</sup>. Compiere steht unter der Mozilla Public License (MPL) und bietet durchaus schon heute einige Kernfunktionalitäten (z.B. Anlagenbuchhaltung und Faktura) an, die das Produkt für den Einsatz in dem einen oder anderen Szenario qualifizieren. Compiere wird von einem Unternehmen entwickelt und gepflegt, das auch Beratung und Support für die Software anbietet.

<sup>172</sup> <http://www.compiere.org>

Der Reifegrad von Open Source ERP Systemen wurde wie folgt gesehen:

- Funktionalität: 2,3
- Benutzerfreundlichkeit: 2,3
- Zuverlässigkeit, Sicherheit: 2,6
- Leistungsfähigkeit: 2,8
- Support: 2,3

Insgesamt wird Open Source Software in diesem Bereich somit eine stark unterdurchschnittliche Ausgereiftheit bescheinigt. Die freien ERP Systeme sind damit einer der am schlechtesten bewerteten Softwarearten.

#### 4.17 Sprachverarbeitung

Der Markt für Sprachverarbeitungssoftware ist recht eng. Dies liegt wohl auch darin bedingt, dass sich diese Technologie bis heute auf dem Massenmarkt nie richtig durchsetzen konnte. Das einzige bedeutende Projekt im Bereich freie Sprachverarbeitungssoftware ist Sphinx<sup>173</sup>. Das von der Carnegie Mellon University in Zusammenarbeit mit mehreren großen Unternehmen getragene Projekt befindet sich aber noch in der Entwicklungsphase.

Der Reifegrad von freier Sprachverarbeitungssoftware wurde wie folgt eingeschätzt:

- Funktionalität: 2,3
- Benutzerfreundlichkeit: 2,4
- Zuverlässigkeit, Sicherheit: 2,6
- Leistungsfähigkeit: 2,7
- Support: 2,1

Insgesamt wird Open Source Software in diesem Bereich somit eine stark unterdurchschnittliche Ausgereiftheit bescheinigt. Freie Sprachverarbeitungssoftware wurde damit als mit am schlechtesten bewertet.

#### 4.18 Zusammenfassung

Wie zu erwarten, ist das Ergebnis der Gegenüberstellung kommerzieller und freier Software uneinheitlich. In einigen Einsatzgebieten ist freie Software der kommerziellen Konkurrenz qualitativ schlicht überlegen. Dies trifft insbesondere auf den Bereich Web Content Management Systeme und auf Teile der Entwicklungswerkzeuge zu. Je nach subjektiver Gewichtung der jeweiligen

<sup>173</sup> <http://cmusphinx.sourceforge.net/html/cmusphinx.php>

Merkmale kann man auch die freien Betriebssysteme und Office Pakete als teilweise stärker als die jeweiligen kommerziellen Produkte betrachten. Sie sind jedenfalls mindestens gleichwertig.

Ein recht ausgeglichenes Bild lieferten die Gebiete Datenbanken und Bildbearbeitungssoftware. Hier ist mit wenigen Ausnahmen kaum noch ein qualitativer Unterschied zwischen kommerzieller und freier Software zu erkennen.

Freie Groupware Systeme und 3D Modellierung / Rendering Software sind für viele, aber nicht alle Einsatzgebiete als gleichwertig zu kommerzieller Software zu werten. Mit zunehmender Komplexität der Anforderungen steigt auch die Wahrscheinlichkeit, dass die verfügbare freie Software diese Anforderungen (noch) nicht abdecken kann. Für die 3D Software trifft ähnliches zu, allerdings mit etwas schwächerer Tendenz.

Derzeit noch nicht mithalten kann freie Software in den Bereichen

- Illustrationssoftware
- Videobearbeitung
- Application Service Providing Software
- Enterprise Resource Planning Systeme
- SRM / CRM Systeme
- Dokumentenmanagement Systeme

In diesen Gebieten muss damit gerechnet werden, dass freie Software nicht alle gestellten Anforderungen erfüllen kann. Teilweise sind viel versprechende Open Source Projekte in diesen Bereichen zugange. Es ist sehr gut möglich, dass freie Software in manchen hier aufgeführten Gebieten mittelfristig sehr wohl konkurrenzfähig wird.

Für die Bereiche in denen keine qualitativen Argumente gegen den Einsatz freier Software sprechen, sollte über deren Einsatz ernsthaft nachgedacht werden. Um eine Entscheidung auch in wirtschaftlicher Hinsicht zu untermauern, können Wirtschaftlichkeitsbetrachtungen durchgeführt werden. Im Folgenden wird eine Wirtschaftlichkeitsbetrachtung für ein konkretes Entscheidungsszenario beispielhaft durchgeführt.

## 5 Wirtschaftlichkeitsbetrachtung

Während in den vorherigen Kapiteln der Einsatz von Open Source Software in den verschiedenen Anwendungsfeldern vor allem aus funktionaler und technischer Sicht untersucht wurde, steht im folgenden Kapitel die wirtschaftliche Betrachtung im Mittelpunkt. Aufgrund der Vielfalt und der unterschiedlichen Bedeutung der Open Source Anwendungsfelder für die Fraunhofer-Gesellschaft, wurden daher nur die drei wichtigsten für eine nähere Wirtschaftlichkeitsbetrachtung ausgewählt. Diese ergaben sich aus einer Befragung von ca. 50 IT-Leitern der Fraunhofer-Gesellschaft nach den relevanten Einsatzgebieten von Open Source Software. Als Ergebnis wurden die Anwendungsfelder Betriebssysteme, Office- und Groupware Anwendungen als relevant für die Fraunhofer-Gesellschaft identifiziert (siehe dazu auch Abbildung 2: Wichtigkeit von Softwarearten für die Fraunhofer-Gesellschaft). Die Betrachtung ist ein exemplarisches Beispiel und kann dann analog auch für weitere Anwendungsfelder durchgeführt werden.

Die Wirtschaftlichkeitsbetrachtung erfolgt anhand eines konkreten Szenarios. Der Kern des Szenarios ist die Migration von persönlichen Mitarbeiterrechnern mit aktuell Microsoft-Software auf Open Source Software. Hierfür werden auf Basis einer Total-Cost-of-Ownership-Analyse (TCO) alle relevanten Kostentreiber ermittelt und monetär bewertet. Eine Prüfung, ob Open Source Software in den besagten Anwendungsfeldern für eine Migration geeignet ist, wurde in den vorherigen Kapiteln eingehend betrachtet und wird im weiteren Verlauf der Wirtschaftlichkeitsbetrachtung als gegeben vorausgesetzt.

Grundlage der Bewertung waren konkrete Informationen aus den Bereichen Kostenrechnung, Beschaffung, Personalwirtschaft und mehrerer IT-Abteilungen einzelner Fraunhofer Institute. Die Informationen stammen überwiegend aus telefonischen und persönlichen Befragungen der verantwortlichen Mitarbeiter.

Die Kosten für die Fortführung einer Microsoft-basierten und einer reinen Open Source-basierten IT-Infrastruktur wurden anhand der Kapitalwertmethode aufbereitet und zum Vergleich gegenübergestellt. Hierbei werden zum einen die Vollkosten und zum anderen nur die haushaltswirksamen Teilkosten verglichen. Bei einem Vollkostenvergleich ergibt sich ein Einsparpotenzial von ca. 2,4 Prozent. Betrachtet man jedoch nur die wirklich haushaltswirksamen Kosten (Kosten, die erst aufgrund des zu betrachtenden Vorhabens entstehen), so lassen sich ca. 6,9 Prozent der zusätzlich zu den schon vorhandenen Kosten aufzubringenden Mittel einsparen. Dies stellt eine erste Abschätzung des Einsparpotenzials dar, die im Falle einer Umsetzung konkretisiert werden sollte.

## 5.1 Beschreibung des Migrationsszenarios

Mit der Ankündigung des neuen Microsoft-Betriebssystems Windows Longhorn (neu: Vista) zum Sommer 2006 steht der Fraunhofer-Gesellschaft die Umstellung zahlreicher Rechner mit Microsoft Windows 2000/XP bevor. Wie die Erfahrung der vergangenen Jahren zeigt, folgen nach der Erscheinung eines neuen Microsoft-Betriebssystems die Office-Anwendungen - Word, Excel, Powerpoint, Access und Outlook - in neuen Versionen, die dann ebenso erneuert (upgrade) werden müssen.

Die Umstellung auf neue Microsoft-Software Versionen wird als Anlass herangezogen, eine Open Source basierte Lösung als Alternative zu prüfen. In der nachfolgenden Untersuchung wird daher ein wirtschaftlicher Vergleich zwischen:

- einer fortführenden Migration im Rahmen der eingesetzten Produkte auf neue Versionen derselben und
- einer ablösenden Migration, d.h. einer Migration in eine komplett neue, Open Source basierte Software-Umgebung innerhalb der Fraunhofer-Gesellschaft

durchgeführt<sup>174</sup>.

In der Untersuchung wird eine Gesamtmigration betrachtet, d.h. die Migration der persönlichen Mitarbeiterrechner, die aktuell mit Microsoft-Software laufen, und diejenigen Server, die für die IT-Infrastruktur notwendigen Dienste bereitstellen, wie Mailserver (E-Mail Dienste), Fileserver (Dokumentenablage) und Printserver (Druckdienste). Neben den eben genannten Rechnern existieren sog. Projektserver und –rechner, die nach projektspezifischen Anforderungen eingerichtet werden und im Vergleich zu Rechnern der grundlegenden IT-Infrastruktur als überwiegend heterogen eingestuft werden. Da eine Erhebung der Rechner aufgrund ihrer Heterogenität in Bezug auf verwendete Betriebssysteme und Software im Rahmen der Studie nicht möglich war, werden die Projektserver und –rechner explizit von der Betrachtung ausgenommen.

<sup>174</sup> Vgl. KBSt.; BMI: Migrationsleitfaden, EDS Business Solutions, Berlin, 2005.

In der folgenden Tabelle sind diejenigen Softwareprogramme aufgelistet, die für den Einsatz in den Migrationsszenarien festgelegt wurden:

<b>Microsoft-basierte Rechner</b>	<b>Open Source-basierte Rechner</b>
<p><b>Clients:</b></p> <ul style="list-style-type: none"> <li>- Microsoft Windows Vista Professional,</li> <li>- Microsoft Office mit Word, Excel, PowerPoint, Access,</li> <li>- Microsoft Outlook</li> </ul> <p><b>Mailserver:</b></p> <ul style="list-style-type: none"> <li>- Microsoft Windows Vista Server,</li> <li>- Microsoft Exchange Server</li> </ul> <p><b>Fileserver/Printserver:</b></p> <ul style="list-style-type: none"> <li>- Microsoft Windows Vista Server</li> </ul>	<p><b>Clients:</b></p> <ul style="list-style-type: none"> <li>- Linux,</li> <li>- OpenOffice.org,</li> <li>- OpenGroupware.org (Client)</li> </ul> <p><b>Mailserver:</b></p> <ul style="list-style-type: none"> <li>- Linux,</li> <li>- OpenGroupware.org (Server)</li> </ul> <p><b>Fileserver/Printserver:</b></p> <ul style="list-style-type: none"> <li>- Linux</li> </ul>

Tabelle 25: Softwareprogramme für die Migrationsszenarien

Die Migrationsszenarien lassen sich in vier grundlegende Phasen einteilen, Beschaffungsphase (1), Einführungsphase (2), Betriebsphase (3) und Upgrade/Neuanschaffungsphase (3b). Letztere ist keine eigenständige Phase, nimmt aber im Rahmen der Betriebsphase eine besondere Bedeutung für den Wirtschaftlichkeitsvergleich ein, da auf Seiten der proprietären Software in der Regel nach ca. drei Jahren ein kostenpflichtiges Upgrade angeboten wird. Für einen ganzheitlichen Vergleich sind nachfolgend diese Kosten separat berücksichtigt.

## 5.2 Ausgangssituation und Annahmen

Der Wirtschaftlichkeitsbetrachtung liegen die folgenden Rahmenparameter zugrunde:

### Anzahl Mitarbeiter

Von der Migration betroffen sind alle Mitarbeiter, die über einen Rechnerarbeitsplatz verfügen. Gemäß den unterschiedlichen Kenntnisständen bzgl. der Computernutzung werden die Mitarbeiter der Fraunhofer-Gesellschaft für die Untersuchung in die folgenden drei Kategorien eingeteilt:

- **Wissenschaftliche Mitarbeiter:** Bei dieser Gruppe werden tendenziell sehr weit reichende IT-Kenntnisse vorausgesetzt. In vielen Fällen müssen sich wissenschaftliche Mitarbeiter im Rahmen ihrer Tätigkeiten intensiv mit IT-Themenstellungen auseinandersetzen. Auch bestehen oft schon Erfahrungen mit Open Source Software. Der Anteil wissenschaftlicher Mitarbeiter, die an einer Schulung teilnehmen werden, wird im Vergleich zu Verwaltungsmitarbeitern und Administratoren geringer eingestuft.
- **Verwaltungsmitarbeiter:** Bei dieser Gruppe wird davon ausgegangen, dass die vorhandenen IT-Kenntnisse über die Anwendung der genannten Software gut sind, jedoch ein etwas größerer Einarbeitungsaufwand für neue Software notwendig werden wird. Der Anteil der Mitarbeiter dieser Gruppe, die an einer Schulung teilnehmen werden, wird im Vergleich zu den wissenschaftlichen Mitarbeitern als hoch eingestuft. Alles in allem wird die Gruppe als sehr homogen erachtet.
- **Hilfskräfte:** Diese Gruppe ist aufgrund der ständigen Zu- und Abgänge am schwierigsten einzuschätzen. Hierunter fallen vor allem die studentischen Hilfskräfte, die je nach Studienrichtung sehr unterschiedliche Voraussetzungen zur Bedienung von Software mit sich bringen. Je nach Betreuer und Aufgabengebiet fallen die Anforderungen bzgl. IT-Kenntnisse unterschiedlich aus. Im Rahmen der Wirtschaftlichkeitsbetrachtung werden Kenntnisse in der Nutzung von Windows und ausgewählter Office-Anwendungen vorausgesetzt. Eine Schulung der Hilfskräfte ist in diesem Konzept nicht vorgesehen.

In Abhängigkeit von der Gruppe fällt unterschiedlicher Aufwand in einzelnen Arbeitsschritten der Einführungs- sowie Betriebsphase an.

## Anzahl Administratoren

Die meisten Fraunhofer Institute haben ihre eigenen IT-Abteilungen, die sehr unterschiedlich aufgestellt sind. Allgemein kann zwischen zentralen und dezentralen IT-Abteilungen unterschieden werden:

- Zentrale IT-Abteilungen: Organisatorisch ist die IT zentral angeordnet und stellt den Mitarbeitern der Fraunhofer Institute IT-Dienstleistungen über zentrale Mailserver, Fileserver und Printserver zur Verfügung. Die Client-rechner der Mitarbeiter werden ebenso zentral betreut.
- Dezentrale IT-Abteilungen: Organisatorisch werden die IT-Dienstleistungen nicht von einer IT-Abteilung erbracht, sondern sind auf mehrere Mitarbeiter oder Gruppen innerhalb eines Instituts aufgeteilt.

Je nach zentraler oder dezentraler Ausrichtung der IT-Abteilungen entstehen im Rahmen der Migrationsszenarien Vor- und Nachteile hinsichtlich:

- Komplexität und Transparenz der Lizenzverwaltung,
- Schulung von Administratoren,
- Arbeitsbelastung je Administrator, etc.

In Abhängigkeit der Institutsgröße wurde die Anzahl der Administratoren abgeschätzt. Insgesamt betreuen ca. 240 Administratoren in der Fraunhofer-Gesellschaft ca. 80 Institute und Einrichtungen.

## Personalkostensätze

Die Personalkostensätze wurden für die drei Gruppen wissenschaftliche Mitarbeiter, Verwaltungsmitarbeiter und Hilfskräfte berechnet. Zur Berechnung wurden die auf Projekte gebuchte Stunden sowie die angefallenen Gemeinkosten gemäß Nachkalkulation des Jahres 2004 herangezogen<sup>175</sup>.

## Anzahl der Rechner

Die Anzahl der Rechner wurde über die Anzahl der Mitarbeiter ermittelt. Einige dieser Rechner werden aus den folgenden zwei Gründen nicht migriert:

- Auf einigen der Rechnern wird schon Open Source Software verwendet, so dass sie damit auch nicht in die Wirtschaftlichkeitsbetrachtung einfließen,
- Einige der Microsoft-basierten Rechner können nicht migriert werden, da die verwendeten Fachanwendungen nur unter Microsoft Windows zur Verfügung stehen und weder im Rahmen eines Windows-Emulators ablaufen noch über einen Terminalserver benutzt werden können.

<sup>175</sup> Informationen von der Zentralverwaltung in München erhalten

Von insgesamt ca. 10 000 Mitarbeiterrechnern werden ca. 1 300 Rechner aus den oben genannten Gründen nicht migriert. Das entspricht ungefähr 13 Prozent der Mitarbeiterrechner. Die Anzahl der Mailserver, Fileserver und Printserver zur Abwicklung der IT-Dienste wurde je Institut bestimmt. Ca. 800 Server wurden in der Fraunhofer-Gesellschaft ermittelt (projektbezogene Server ausgenommen).

### Annahmen im Überblick

Tabelle 26 zeigt einige der ermittelten und festgelegten Annahmen im Überblick.

Tabelle 26: Auflistung der Annahmen

<b>Mitarbeiter</b>	
wissenschaftliche Mitarbeiter (Kontingent)	4.974
Verwaltungsmitarbeiter (Kontingent)	3.354
Hiwi (Personen)	3.200
Administratoren (Personen)	240
<b>Anzahl Rechner</b>	
Clientrechner (insgesamt)	10.076
Clientrechner Linux	300
Clientrechner nicht migrierbar	1.001
Clientrechner migrierbar	8.775
Groupwareserver	240
Print-/File-Server	588
<b>Schulungen</b>	
Dauer einer Microsoft-Windows/Office Schulung in Std.	6-8
Dauer einer Linux/Open Office Schulung in Std.	8-16
Anteil Mitarbeiter an Microsoft-Schulungen je nach Mitarbeitergruppe	10-70%
Anteil Mitarbeiter an Open Source Schulungen je nach Mitarbeitergruppe	15-80%
<b>Installation</b>	
Installation je Client in Std. (Microsoft/Open Source)	1
Installation je Server in Std. (Microsoft/Open Source)	4
<b>Produktivitätsausfall</b>	
Durchschn. Produktivitätsausfall je Mitarbeiter in Std. in der Einführungsphase bei einer Microsoft-Installation	2
Durchschn. Produktivitätsausfall je Mitarbeiter in Std. in der Einführungsphase bei einer Open Source-Installation	2,5
<b>Administration Einführungsphase</b>	
Durchschn. Administrationsaufwand je Mitarbeiter in Min. in der Einführungsphase (Microsoft)	3
Durchschn. Administrationsaufwand je Mitarbeiter in Min. in der Einführungsphase (Open Source)	4
<b>Administration Betriebsphase</b>	
Durchschn. Administrationsaufwand je Mitarbeiter in Min. pro Jahr (Microsoft/Open Source)	ca. 6

### 5.3 Kostentreiber

Die Berücksichtigung des vollständigen Software-Lebenszyklus spielt bei den Migrationsszenarien eine besondere Rolle. So vergleichen einige Studien lediglich die Einführung einer Software ohne die Betriebs- und die Upgrade-Phase zu berücksichtigen. Insbesondere in den letzten beiden Phasen des Software-Lebenszykluses fallen oft noch relevante Kosten an, wie z.B. Upgrade-Kosten, Aufwand für Administration und externer Support.

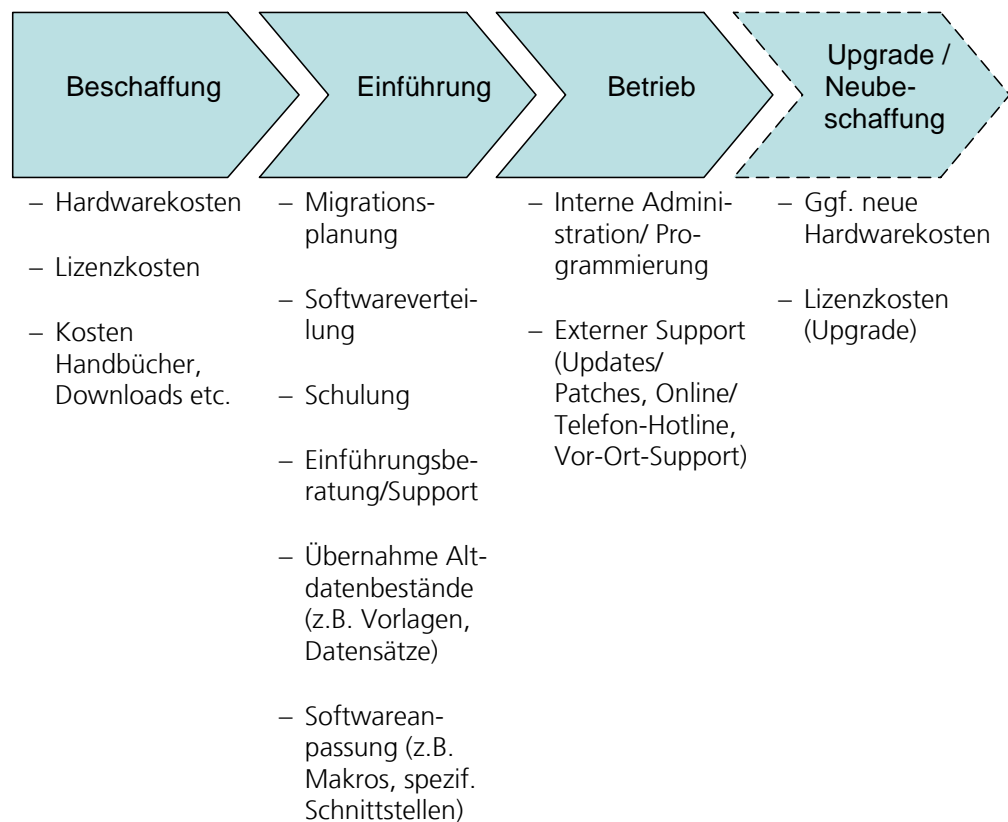


Abbildung 32: Software-Lebenszyklus und Kostentreiber<sup>176</sup>.

Um eine ganzheitliche Betrachtung der Wirtschaftlichkeit der Migrationsszenarien zu gewährleisten, finden die Kostentreiber mittels Kapitalwertmethode über einen Sechs-Jahreszeitraum Berücksichtigung. Nachfolgend die Beschreibung der einzelnen Kostentreiber und deren Bedeutung für die Wirtschaftlichkeitsbetrachtung.

<sup>176</sup> In Anlehnung an Binder, S.; Henke, M.: Kassensturz: Open-Source und proprietäre Software im Vergleich, SOREON Research, Frankfurt, 2003.

### 5.3.1 Hardwarekosten (haushaltswirksam)

In der Betrachtung wird vorausgesetzt, dass sich die Anforderungen der proprietären und der Open Source Software an die Hardware nicht signifikant unterscheiden. Daher werden für beide Migrationsszenarien Hardwarekosten in gleicher Höhe angenommen.

Die Hardware wird zu Beginn der Migration beschafft. Aufgrund des Alterungsprozesses wird eine Ersatzbeschaffung der Hardware nach fünf Jahren angesetzt.

### 5.3.2 Lizenzkosten (haushaltswirksam)

Die Lizenzkosten gehören zu den relevanten Kostentreibern beim Vergleich der Migrationsszenarien auf proprietäre oder Open Source Software. Bei einer Microsoft-Migration fallen Lizenzkosten durch die Beschaffung der OEM-Betriebssystemlizenzen (Home-Edition) im ersten und nach dem fünften Jahr an und zusätzlich entstehen der Fraunhofer-Gesellschaft jährliche Lizenzkosten aufgrund des Campusvertrags.

Lizenzkosten für weitere proprietäre Software, wie z.B. Virenschutzprogramme, Software zur Komprimierung, Personal Firewalls und ggf. Adobe Acrobat, sind bei einer Microsoft-basierten Lösung darüber hinaus zu beschaffen. Diese Lizenzkosten entfallen bei einer Open Source Lösung. In der Wirtschaftlichkeitsbetrachtung werden diese Kosten nicht berücksichtigt.

Bei einer Open Source Lösung entstehen keine Lizenzkosten für die Fraunhofer-Gesellschaft. Eine zentrale IT-Abteilung erstellt eine Fraunhofer-Gesellschaftweite Installationsversion des Linux-Betriebssystems, der Office- und Goupware-Anwendungen. Diese Version kann über das Intranet zum Download zur Verfügung gestellt werden. Da jedoch nicht alle Rechner auf Open Source Software umgestellt werden können, müssen Lizenzen für die ca. 1 000 nicht migrierbaren Rechner beschafft werden. Die Lizenzkosten fallen im ersten sowie bei einem Upgrade nach dem dritten Jahr an.

### 5.3.3 Migrationsplanung (nicht haushaltswirksam)

Bei der Einführung von Open Source Software ist eine Migrationsplanung notwendig, in der ggf. die Umstellung von Servern (z.B. Directory Server, Mail- und Kalender-Server, File- und Print-Server, DNS- und DHCP-Server) und der Desktop-Umgebung vorbereitet wird. Die Planung muss sowohl bei der Migration auf proprietärer als auch auf Open Source Software vorgenommen werden. Der Aufwand sollte sich in diesem Punkt überwiegend die Waage halten und wird damit als kostenneutral angenommen.

### 5.3.4 Installation (nicht haushaltswirksam)

Die einzelnen Rechner werden von den Administratoren installiert. Zur Vereinfachung können bei gleicher Hardware Images erstellt werden, die auf die einzelnen Rechner überspielt werden. Eine manuelle Installation z.B. spezifischer Software kann jedoch damit nicht ersetzt werden. Der Installationsaufwand fällt für beide Migrationsszenarien in gleicher Höhe an.

### 5.3.5 Übernahme Altdatenbestände (nicht haushaltswirksam)

Die folgenden Altdatenbestände treten in den vorliegenden Szenarien auf:

- Dokumentenvorlagen (betrifft Word, Excel, Powerpoint): Die Importschnittstelle der Open Office Anwendungen können die Microsoft-Dokumente relativ problemlos einlesen (siehe dazu die Ausführungen in Abschnitt 4).
- Kontakte, Aufgaben, Termine und E-Mails (betrifft Outlook/Exchange-Server): Die besagten Daten müssen aus dem Altsystem übernommen werden. Die Übernahme muss bei beiden Migrationsszenarien erfolgen.

Die Kosten der Übernahme von Altdatenbeständen werden im Vergleich zu den anderen Kostentreibern als vernachlässigbar bzw. kostenneutral angesehen.

### 5.3.6 Softwareanpassung (nicht haushaltswirksam)

Unter dem Begriff ist vor allem die Anpassung von Makros, plattform-spezifischem Programmcode oder spezifischen Schnittstellen zu verstehen. Softwareanpassungen kommen in den geschilderten Szenarien in Form von Makrokonvertierungen vor. Dies kann bei einer Microsoft-basierten sowie bei einer Open Source-basierten Migration erfolgen.

Makros kommen in der Fraunhofer-Gesellschaft punktuell vor. Eine Aussage über den Aufwand einer derartigen Konvertierung ist schwierig. Im Vergleich zu den anderen Kostentreibern werden die Kosten als vernachlässigbar angesehen.

### 5.3.7 Schulung (teilweise haushaltswirksam)

Schulungen zur Anwendung des neuen Betriebssystems sowie der Office-Anwendungen werden für die Mitarbeiter angeboten. Darüber hinaus werden die Administratoren in der Installation, Konfiguration und Nutzung der neuen Software für Clients im Netzwerk sowie für die Server geschult.

Hierbei wird davon ausgegangen, dass die Dauer der Schulungen des Betriebssystems und der Office-Anwendungen der Microsoft-Software ca. 6 Stunden

und der Open Source Software ca. 8 Stunden beträgt. Aufgrund der Ähnlichkeit der Bedienung und der Funktionalitäten beider Softwaretypen kann auch bei einer Einführung in die Open Source Software von grundlegenden Vorkenntnissen ausgegangen werden.

Bei einer Betrachtung der einzelnen Gruppen von Mitarbeitern werden sicherlich unterschiedliche Verhaltensmuster hinsichtlich der Schulungsbeteiligung festzustellen sein. So wird davon ausgegangen, dass die Schulungsteilnahme von wissenschaftlichen Mitarbeitern tendenziell gering, die der Verwaltungsmitarbeiter tendenziell hoch und die der Administratoren ebenso tendenziell hoch ausfallen wird.

Bei den Schulungskosten wird zwischen haushaltswirksamen Kosten, wie z.B. die Nutzung externer Trainer, sowie nicht haushaltswirksamen Kosten, wie z.B. der Produktivitätsausfall der Mitarbeiter, unterschieden. Die Schulungen bilden einen relevanten Kostenblock, der für den Vergleich eine besondere Rolle spielt.

#### **5.4 Reduzierte Produktivität (nicht haushaltswirksam)**

Bei der Nutzung der neuen Software wird es voraussichtlich bei den Mitarbeitern in den ersten Wochen zu einigen Problemen kommen, für deren Lösung Zeit in Anspruch genommen wird, die im Alltagsgeschäft erstmal fehlen wird. Als Einarbeitungszeit werden acht Wochen angesetzt. In der Wirtschaftlichkeitsbetrachtung wird davon ausgegangen, dass die Ausfallzeiten bei der Open Source Lösung in der Einarbeitungsphase etwas höher ausfällt als bei einer Microsoft-basierten Lösung.

Bei einem Vergleich der Benutzeroberflächen der betrachteten Software lässt sich eine starke Ähnlichkeit erkennen. So sind z.B. Microsoft Office und Open Office in der Bedienung und in ihrer Funktionalität nahezu identisch, was sich in einer schnellen Einarbeitungszeit der Mitarbeiter in die Softwarebedienung bemerkbar machen wird.

##### **5.4.1 Einführungsberatung/-unterstützung (nicht haushaltswirksam)**

In der Einführungsberatung/-unterstützung stehen die Administratoren zur Problemlösung zur Verfügung. Hier wird, wie bei dem vorherigen Punkt, davon ausgegangen, dass der Aufwand für Beratung und Unterstützung insbesondere in der Einführungsphase etwas höher auftreten wird und anschließend in die routinemäßige Administration übergehen wird. Die Einführungsberatung/-unterstützung wird für die ersten acht Wochen angesetzt. Bei der Microsoft-basierten Migration wird von einem etwas geringeren Aufwand diesbezüglich ausgegangen als bei einer Open Source Lösung.

#### **5.4.2 Interne Administration/ Programmierung (nicht haushaltswirksam)**

In der Betriebsphase geht die Einführungsberatung und -unterstützung in die interne Administration über. Die interne Administration bietet die Dienstleistungen eines Helpdesk an. Die Kosten hierfür sind für beide Migrationsszenarien vergleichbar und somit werden sie im Vergleich als kostenneutral berücksichtigt.

#### **5.4.3 Externer Support (nicht haushaltswirksam)**

Unter externem Support sind u. a. Updates/Patches, Online/Telefon-Hotline und Vor-Ort-Support zu verstehen. Externer Software-Support z.B. in Form einer Telefon-Hotline oder eines Vor-Ort-Service ist – wie bisher in der Fraunhofer-Gesellschaft üblich – nicht vorgesehen. Die Updates/Patches sind über das Internet für Microsoft sowie Linux automatisch abrufbar. Aus diesem Grund werden für den externen Support keine Kosten angesetzt.

#### **5.4.4 Upgrades/Neuanschaffung (haushaltswirksam)**

Die Erfahrungen der letzten Jahre zeigte, dass mit neuen Versionen von Microsoft Windows sowie Office alle ca. 3 Jahre gerechnet werden kann<sup>177</sup>. Um mit dieser Entwicklung Schritt zu halten, werden Investitionen für Upgrades der proprietären Software nach dem dritten Jahr notwendig.

### **5.5 Wirtschaftlichkeitsbetrachtung**

Nachfolgend wird für die zwei Migrationsszenarien die Wirtschaftlichkeit unter Berücksichtigung der oben genannten Kostentreiber berechnet. Der Betrachtungszeitraum ist sechs Jahre. Eine Abzinsung der Kapitalströme wurde dabei nicht vorgenommen. Die in den folgenden Total Cost of Ownership Analysen (TCO) dargestellten Prozentzahlen basieren auf den einfachen Summen der zugrunde liegenden negativen Kapitalströme der einzelnen Jahre, vergleichbar mit den operativen IT-Budgets von Unternehmen<sup>178</sup>.

<sup>177</sup> Vgl. Binder, S.; Henke, M.: Kassensturz: Open Source und proprietäre Software im Vergleich, SOREON Research, Frankfurt, 2003.

<sup>178</sup> Vgl. Binder, S.; Henke, M.: Kassensturz: Open Source und proprietäre Software im Vergleich, SOREON Research, Frankfurt, 2003, S.19-20.

**Total Cost Of Ownership Microsoft-Open Source Software im Vergleich zum der SOREON-Studie (in TEuro)**

Migrationphase	Kostentreiber	Relevanz	Vollkostenrechnung Diff. (%) <sup>1)</sup>	Teilkostenrechnung mit Kriterien der SOREON Studie Diff. (%) <sup>1)</sup>
Beschaffungsphase	Hardware (Server, Client-Rechner)	kostenneutral	0,0%	-
	Lizenzkosten	hoch	-7,0%	-22,5%
	Migrationplanung	kostenneutral	0,0%	-
Einführungsphase	Übernahme Altdatenbestände	gering	0,0%	-
	Softwareanpassung	gering	0,0%	-
	Installation	kostenneutral	0,0%	0,0%
	Schulungen (haushaltswirksam)	mittel	0,7%	2,3%
	Schulungen (nicht haushaltswirksam)	hoch	2,0%	6,4%
Betriebsphase	Reduzierte Mitarbeiterproduktivität	mittel	0,7%	-
	Einführungsberatung/-unterstützung	gering	0,2%	0,5%
	Interne Administration/ Externer Support	kostenneutral	0,0%	0,0%
	Hardware-investitionen Upgrades	hoch	1,0%	3,3%
<b>Gesamt-migrationskosten:</b>			<b>-2,4%</b>	<b>-10,0%</b>

<sup>1)</sup> relative Differenz zu den Gesamtkosten einer Microsoft-basierter Migration in Prozent

### 5.5.1 Vollkostenbetrachtung

Bei einem Vollkostenvergleich der beiden Migrationsszenarien wird ein Einsparpotenzial von ca. 2,4 Prozent ermittelt. In diesem Vergleich werden die relevanten Kostentreiber deutlich.

#### Lizenzkosten

Insbesondere die Lizenzkosten zu Beginn der Investition stellen mit einem Minus von ca. 7 Prozent das größte Einsparpotenzial dar. Aufgrund von ca. 1 000 Client-Rechner, die nicht auf Open Source Software migrierbar sein werden, entstehen bei einer Open Source basierten Migration nach drei Jahren Lizenzkosten für Microsoft-Upgrades von ca. 1,0 Prozent.

#### Schulungskosten

Bei einer Open Source basierten Migration entsteht ein Mehraufwand von ca. 2,7 Prozent. Dies liegt vor allem darin begründet, dass zum einen voraussichtlich eine größere Anzahl an Mitarbeitern die Schulungen besuchen wird und zum anderen ein zeitlicher Mehraufwand für die Schulungen zu den Themen Betriebssystem (ein Viertel Mehraufwand) sowie Administration (doppelten Mehraufwand) entstehen wird.

#### Mitarbeiterproduktivität

Der dritte Kostenblock bildet eine Reduzierung der Mitarbeiterproduktivität in der Einführungsphase, in der bei einer Open Source basierten Migration voraussichtlich 0,7 Prozent mehr an Kosten entstehen aufgrund von fehlender Arbeitszeit, die für die Lösung von Problemen benötigt wird.

#### Einführungsberatung/-unterstützung

Während der Einführungsberatung fällt ebenso ein geringfügiger Mehraufwand von 0,2 Prozent in einem Open Source-basierten Migrationsszenario an, da aufgrund der Neuartigkeit insbesondere des Betriebssystems mit Klärungsbedarf gerechnet werden kann.

### 5.5.2 Teilkostenbetrachtung von haushaltswirksamen Kostentreibern

Der Unterschied zwischen haushaltswirksamen und nicht haushaltswirksamen Kostentreibern ist wie folgt<sup>179</sup>:

- **Haushaltswirksame Kosten** sind Positionen, die erst *aufgrund des betrachteten Vorhabens* entstehen und die in der (nächsten) Veranschlagung des Haushaltes zu Mehr- oder Minderbeantragungen führen.
- **Nicht haushaltswirksame Kosten** sind solche Positionen, die *auch ohne das betrachtete Vorhaben* in gleicher Höhe entstehen.

Werden lediglich die haushaltswirksamen Kosten betrachtet, werden:

- die kostenneutralen Kostentreiber Installation und interne Administration zu den nicht haushaltswirksamen Kostentreibern eingeordnet,
- die Kostentreiber der reduzierten Mitarbeiterproduktivität und der Einführungsberatung/-unterstützung als nicht haushaltswirksam eingestuft,
- die Anteile der Schulungskosten, die aufgrund der Ausfallzeiten von Mitarbeitern entstehen, zu den nicht haushaltswirksamen Kosten gerechnet.<sup>180</sup>

Damit ergibt sich bei Betrachtung der haushaltswirksamen Kosten ein Einsparpotenzial von ca. 6,9 Prozent auf die haushaltswirksamen Gesamtkosten.

## 5.6 Vergleich der Ergebnisse mit Studien

Die Ergebnisse der Wirtschaftlichkeitsbetrachtung in der Fraunhofer-Gesellschaft wurden mit denjenigen der Soreon Studie verglichen. Bei einer Gegenüberstellung der oben genannten Ergebnisse und der Einsparpotenziale, müssen die Kostentreiber aufeinander abgestimmt werden. So werden in der Soreon Studie die Hardwarekosten, die Migrationsplanung und die reduzierte Mitarbeiterproduktivität nicht berücksichtigt.

Bei der Betrachtung der vergleichbaren Kostentreiber wird ein Einsparpotenzial von ca. 10 Prozent ermittelt. Die Soreon Studie<sup>181</sup> kommt zu Einsparungen von

<sup>179</sup> Vgl. KBSt.; BMI: Migrationsleitfaden, EDS Business Solutions, Berlin, 2005.

<sup>180</sup> Siehe Spalte »hw« in der Tabelle »Total Cost Of Ownership Microsoft-Open Source Software im Vergleich«

<sup>181</sup> Vgl. Binder, S.; Henke, M.: Kassensturz: Open Source und proprietäre Software im Vergleich, SOREON Research, Frankfurt, 2003

6 Prozent für mittelgroße Unternehmen mit ca. 100 Arbeitsplatzrechnern und von 20 Prozent für große Unternehmen mit ca. 2000 Arbeitsplatzrechnern und jeweils entsprechender IT-Infrastruktur.

Das Einsparpotenzial von 10 Prozent im Vergleich zu den Ergebnissen der Soreon Studie fällt eher gering aus, nach der mittelgroße Unternehmen ca. 6 Prozent und große Unternehmen ca. 20 Prozent ihrer Kosten einsparen. Nachfolgend einige relevante Aspekte, die die Einsparpotenziale bei den betrachteten Migrationsszenarien in der Fraunhofer-Gesellschaft beeinflussen:

- Anzahl der Mitarbeiter und mehr als 10 000 Lizenzen;
- einheitliche Lizenzkosten aufgrund von Rahmenverträgen;
- dezentrale/zentrale Strukturen aufgrund der Eigenständigkeit der Fraunhofer Institute und der Bedeutung der Zentralverwaltung;
- gute IT-Kenntnisse der Mitarbeiter insbesondere der wissenschaftlichen Mitarbeiter.

Die Einsparpotenziale der Fraunhofer-Gesellschaft liegen u. a. aufgrund der genannten Aspekte für eine Open Source Migration zwischen den Einsparpotenzialen der mittelgroßen und großen Unternehmen, die in der Soreon Studie ermittelt wurden.

## 6 Strategische Analyse

### 6.1 Bewertung der Vor- und Nachteile freier Software

Die Vorteile und Nachteile von Open Source Software wurden zu Beginn der Studie bereits identifiziert. In diesem Abschnitt werden diese noch einmal aufgeführt und bewertet.

#### 6.1.1 Vorteile freier Software

##### **Anpassbarkeit**

Open Source Software kann von jedem für individuelle Zwecke erweitert oder angepasst werden. Im Vergleich zu einer Neuentwicklung können damit umfangreiche Synergieeffekte realisiert werden. Die Vorteile von Standard-Software (bewährtes Grundprodukt) und Eigenentwicklung (Anpassung an individuelle Zwecke) können so vereinigt werden.

Die Möglichkeit der beliebigen Anpassbarkeit freier Software könnte sich insbesondere im Bereich Wissenschaftlich-Technische und sonstige Spezial Software positiv auswirken. So können beispielsweise spezielle (z. B. für Forschungszwecke benötigte) Funktionalitäten ohne Rücksprache oder langwierige Verhandlungen mit einem kommerziellen Anbieter einer Software hinzugefügt werden.

##### **Wiederverwendbarkeit von Code**

Durch die Verwendung von Komponenten eines Open Source Produkts in eigenen Produkten kann Entwicklungszeit eingespart werden.

Auch dieser Vorteil ist von hohem Wert. In vielen Bereichen wird Software entwickelt, hierfür könnten Komponenten aus Open Source Software die Entwicklungszeit erheblich beschleunigen. Im Falle von BSD lizenzierter Software ist sogar eine weitere kommerzielle Nutzung des resultierenden Produkts möglich. GPL lizenzierte Software kann allerdings nur verwendet werden wenn das Ergebnis wiederum freie Software ist, bzw. das Ergebnis nur rein internen Zwecken dient. Unter dem Gesichtspunkt der Kosteneffizienz sollte die Verwendung von freier Software oder Komponenten aus freier Software in Entwicklungsprojekten durchaus als strategisches Mittel betrachtet werden.

## Höhere Produktqualität

Open Source Software wird (zumindest von deren Befürwortern) eine tendenziell höhere Produktqualität zugeschrieben.

Die These der grundsätzlich höheren Produktqualität von freier Software wird insbesondere durch den grundlegend anderen Entwicklungsprozess untermauert, der für Open Source Software gilt. Diese unterliegt im Gegensatz von kommerzieller Software keinerlei Marktzwängen, insbesondere müssen keine festen Veröffentlichungstermine eingehalten werden. Dadurch entfällt der im kommerziellen Bereich vielfach vorzufindende Zeitdruck, der bei dieser Art von Software zu der berüchtigten »release now – patch later« Praxis geführt hat. Dabei wird, um feststehende Veröffentlichungstermine einzuhalten, teilweise unfertige und mangelhaft getestete Software auf den Markt gebracht und die unweigerlich beim Einsatz auftretenden Probleme anschließend durch nachgeschobene Patches behoben. Im Falle von Open Source Software können sich die Entwickler die Zeit nehmen, die sie benötigen (»it's done when it's done« Prinzip). Hier stehen technische Gesichtspunkte im Vordergrund, nicht kaufmännische.

Ob freie Software tatsächlich von höherer Qualität im Vergleich zu kommerzieller Software ist, hängt stark mit der jeweiligen Softwareart zusammen. Die Open Source Projekte weisen derzeit sehr unterschiedliche Entwicklungsstände auf. Es konnte gezeigt werden, dass dies in einigen Gebieten und für einige Einsatzszenarien auch tatsächlich schon der Fall ist (etwa für Office Pakete, Datenbanken, Entwicklungswerkzeuge und Web Content Management Systeme). In solchen Fällen ist die Verwendung freier Software grundsätzlich zu empfehlen.

## Anbieterunabhängigkeit

Open Source Software zwingt im Gegensatz zu manchem kommerziellen Pendant die Nutzer nicht in ein Abhängigkeitsverhältnis zu bestimmten Herstellern. Vielmehr genießen die Nutzer freier Software umfassende Freiheitsrechte in Bezug auf deren Einsatz.

In der Softwarebranche werden viele Märkte von sehr wenigen Großunternehmen beherrscht. Die Marktmacht dieser Unternehmen ist erheblich, was in der Praxis tatsächlich zu einem Abhängigkeitsverhältnis zu diesen Anbietern führt. Dies zieht sowohl finanzielle als auch strategische Konsequenzen nach sich. Es ist zu erwarten, dass die Quasi-Monopolstellung einiger Softwareunternehmen zu vergleichsweise hohen Preisen geführt hat bzw. führt. Auch größere Organisationen, die derzeit eher günstige Preise verhandeln können, sollten sich nicht darauf verlassen, dass die Lieferbedingungen sich zukünftig nicht zu ihren Ungunsten ändern könnten.

Umgekehrt ist vielmehr zu vermuten, dass der steigende Konkurrenzdruck – nicht zuletzt aus dem Bereich der Open Source Software - zu der derzeit zu beobachtenden Kompromissbereitschaft der Softwareanbieter geführt hat<sup>182</sup>. Sollte freie Software bei Unternehmensanwendern langfristig nicht gegen kommerzielle Anbieter bestehen können, würde dies mit großer Sicherheit zu einer Wiedererstarkung der Verhandlungsposition der Softwareanbieter führen.

Ein weiteres Kriterium betrifft die Abhängigkeiten von Nutzungsbedingungen der Anbieter. Diese können den Anwendern die Möglichkeiten des Einsatzes ihrer Produkte sehr weitgehend vorschreiben – und führen dies in der Praxis auch durch. Mit zunehmender Marktmacht der Anbieter werden diese auch in die Lage versetzt, die Bedingungen einseitig ändern bzw. verschärfen zu können. Im schlimmsten Fall führt dies dazu, dass die Produkte für bestimmte Einsatzzwecke nicht mehr verwendet werden dürfen. Diesem Risiko kann man sich durch den Einsatz freier Software weitestgehend entziehen.

### **Höhere Sicherheit**

Oft wird Open Source Software ein höheres Maß an Sicherheit zugesprochen als Closed Source Produkten. Da jeder Einsicht in den Quellcode nehmen kann, werden Sicherheitslücken und -probleme schneller entdeckt, als dies bei kommerzieller Software der Fall ist. Die Möglichkeit zum Einbau von Hintertüren besteht bei freier Software ebenfalls nicht, da der entsprechende Code im Quelltext schnell auffallen würde.

Sicherheit ist selbstverständlich für alle Anwender von Software von höchstem Interesse. Die Abhängigkeit von einer funktionierenden IT Infrastruktur ist heutzutage überall sehr hoch, teilweise werden auch vertrauliche Daten gehalten, bzw. verarbeitet.

Das Argument, freie Software ist aufgrund des offenen Quellcodes sicherer als kommerzielle Software, ist nicht unumstritten. Kommerzielle Anbieter weisen daher sehr gerne auch auf identifizierte Sicherheitsprobleme in freier Software hin. Auch wird aus dem Lager der kommerziellen Softwareanbieter gern behauptet, gerade die Geheimhaltung des Quellcodes verspricht ein höheres Maß an Sicherheit, da so die vorhandenen Sicherheitslücken gar nicht erst von Außenstehenden gefunden werden können. Die meisten Sicherheitsexperten sind aber der Meinung, dass diese »security by obscurity« These nicht haltbar ist<sup>183</sup>.

<sup>182</sup> Es sei in diesem Kontext an das Verhandlungsverhalten von Microsoft im Zuge der Überlegungen der Stadt München, nach Linux zu migrieren, erinnert (siehe dazu z.B. <http://www.zdnet.de/news/software/0,39023144,2133005,00.htm>).

<sup>183</sup> Vgl. z.B.: Schneier, Bruce: Internet Shield: Secrecy and security. <http://www.schneier.com/essay-033.html>

### **Offene Standards**

Die von Open Source Produkten verwendeten Dateiformate und Datenaustausch-Standards sind über die Verfügbarkeit der Quelltexte per Definition offen gelegt. Dies hat eine höhere Kompatibilität und Interoperabilität von Open Source Produkten mit anderer Software zur Folge, da entsprechende Schnittstellen mit geringerem Aufwand erstellt werden können.

Die Verwendung von auf offenen Standards basierender Software vermindert auch die Abhängigkeit von bestimmten Herstellern deutlich. Als Beispiel sollen hier die Microsoft Office Formate dienen, die von Microsoft sehr wohl zur Absicherung des Marktanteils seiner Office Suite missbraucht werden. Da eine solche Abhängigkeit aus Anwendersicht definitiv nicht wünschenswert ist, kann die Verwendung von offenen Standards als gravierender Vorteil für die Anwender von Software betrachtet werden.

### **Keine Lizenzkosten**

Für Open Source Softwareprodukte fallen im Normalfall keine Lizenzkosten an. Dies wirkt sich in der Regel positiv auf die Wirtschaftlichkeit von Software aus. Da der Kostendruck in fast allen Organisationen ein wichtiger Faktor ist, sollte auch der wirtschaftliche Aspekt freier Software entsprechend gewürdigt werden. Ob der Wegfall der Lizenzkosten tatsächlich zu einer Absenkung des TCO von Software führt, sollte durch die Durchführung von Wirtschaftlichkeitsanalysen im Einzelfall geklärt werden. In vielen Fällen ist dies tatsächlich so.

## **6.1.2 Nachteile freier Software**

### **Keine Gewährleistungsrechte**

Der Einsatz von Open Source Software ist in der Regel mit einem sehr weitgehenden Ausschluss von Haftungs- und Gewährleistungsrechten verbunden. Dieses Kriterium kann nahezu vollständig dadurch entkräftet werden, dass die sich Situation bei kommerzieller Software in der Regel davon nicht wesentlich unterscheidet. Diesem Nachteil sollte daher kein allzu großes Gewicht beigemessen werden.

### **Kein Support durch Hersteller bzw. Entwickler**

In vielen Fällen existiert für Open Source Software kein Support in einer Ausbaustufe wie dies bei vielen kommerziellen Produkten der Fall ist. Unterstützung und Beratung ist für viele Open Source Produkte tatsächlich schwer oder gar nicht zu bekommen. Vielmehr unterstützen sich die Anwender freier Software oft gegenseitig bzw. über Internet Foren.

Dies ist aber nicht für alle Open Source Produkte der Fall. Zunehmend wird freie Software von Firmen angeboten, deren Geschäftsmodell gerade im Angebot von Support für ihre Software besteht (z.B. MySQL AB, JBoss oder Zope).

Der teilweise fehlende Support ist letztendlich ebenfalls kein wirklich als hoch zu bewertender Nachteil. Die Befragung der IT Verantwortlichen ergab, dass der Support ohnehin als das unwichtigste Qualitätsmerkmal von Software betrachtet wird. In vielen Fällen ist ausreichend Know-how bei Mitarbeitern vorhanden, um auf externen Support verzichten zu können.

### **Höherer Schulungsaufwand**

In vielen Bereichen sind kommerzielle Softwareprodukte (noch) verbreiteter, als ihre Open Source Alternativen (z.B. Office-Produkte oder Betriebssysteme). Das führt dazu, dass sowohl unternehmensintern als auch auf dem Arbeitsmarkt eher Mitarbeiter mit Know-how für diese kommerziellen Produkte gefunden werden. So haben sehr viele Anwender Windows Know-how, während Linux Kenntnisse noch vergleichsweise selten sind. Sollen freie Alternativen zu diesen Produkten eingesetzt werden, müssen Mitarbeiter oftmals zunächst geschult bzw. neu eingestellt werden.

Dieser Nachteil hängt allerdings weniger mit qualitativen Eigenschaften von Open Source Software zusammen, als vielmehr mit der Tatsache, dass die noch vergleichsweise junge Open Source Software in fast allen Fällen später in den Markt gestartet ist, als die kommerzielle Konkurrenz.

Dieser Nachteil ist nicht völlig von der Hand zu weisen, aber auch hier ist zu vermuten, dass er insbesondere in größeren Unternehmen weniger stark ins Gewicht fällt, als in vielen kleineren Unternehmen. Der Bedarf zumindest an vergleichsweise teuren, externen Schulungsmaßnahmen dürfte für diese Anwendergruppe eher gering sein.

Des Weiteren ist freie Software in manchen Bereichen schon hinreichend verbreitet, um das Argument der höheren Schulungskosten weitestgehend zu entkräften. Dies ist insbesondere bei Webservern und Datenbanken der Fall. Hier ist bereits ausreichend Know-how bei Administratoren und Anwendern vorhanden.

### **Ungewisse Weiterentwicklung**

Für Open Source Produkte existieren selten langfristige Verpflichtungen zur Wartung, Pflege und Weiterentwicklung der Software. Vielmehr können Open Source Produkte von ihren Entwicklern jederzeit eingestellt werden.

Dieser, zunächst als gravierend erscheinender Nachteil ist aber kein wirkliches Argument gegen freie Software, da es einen Schutz vor Einstellung eines Produkts auch bei kommerzieller Software letztendlich nicht gibt. Auch Unternehmen können die Weiterentwicklung eines Produkts jederzeit einstellen und zumindest mittelfristig auch die Wartung und Pflege von bestehenden Produkten beenden. In der Praxis werden kommerzielle Softwareprodukte durch ihre Hersteller abgekündigt, sobald Nachfolgerprodukte am Markt sind. So endete der »Mainstream Support« für Windows 2000 bereits am 31. März 2005 – die Abkündigung erfolgte gerade einmal vier Jahre nach Markteinführung! Selbst der »Extended Support« für dieses Betriebssystem wird nur noch bis 31. März 2007 aufrechterhalten<sup>184</sup>.

In der Praxis sind die bedeutenden Open Source Projekte ohnehin keiner Gefahr durch Einstellung ausgesetzt. Linux, als das größte von der Community getragene Open Source Projekt hat in der Vergangenheit den Ausfall von Entwicklern problemlos kompensieren können. Es ist nicht zu vermuten, dass selbst der Ausfall von Leitfiguren wie Linus Torvalds oder Andrew Morton das Projekt gefährden könnte. Viele andere Open Source Projekte werden von Firmen getragen, die sehr wohl ein langfristiges Commitment für die Weiterentwicklung gegeben haben. Dies ist etwa bei OpenOffice, MySQL, Zope oder JBoss der Fall. Die Wahrscheinlichkeit, dass eines dieser Projekte aufgegeben werden könnte, ist auf keinen Fall höher als bei vergleichbarer kommerzieller Software.

Sogar bei Eintritt des nicht zu erwartenden Worst Case (Aufgabe eines für den Anwender wichtigen Open Source Projekts), kann im Notfall die Wartung des Produkts immer noch selbst durchgeführt werden – was bei kommerzieller Software nicht möglich wäre.

### **Benötigte Applikationen sind nicht verfügbar**

Selbstverständlich kann Open Source Software nur dann eingesetzt werden, wenn für einen bestimmten Einsatzzweck auch entsprechende freie Software zur Verfügung steht. Es konnte im Verlauf der Studie festgestellt werden, dass dies noch nicht überall der Fall ist. In diesen Fällen wird wohl nach wie vor kommerzielle Software eingesetzt werden (müssen), allerdings ist dies kein Nachteil von Open Source Software an sich.

Zum Nachteil wird dieses Kriterium insbesondere dann, wenn Seiteneffekte vorliegen. Diese treten beispielsweise auf, wenn bestimmte benötigte Applikationen nur für Windows erhältlich sind, als Betriebssystem aber eigentlich Linux eingesetzt werden soll.

<sup>184</sup> <http://www.microsoft.com/germany/windows2000/abkuendung.msp>

### **Mangelhafte Interoperabilität mit kommerzieller Software**

Kommerzielle Softwareanbieter haben meist kein Interesse an einem Zusammenspiel ihrer Software mit Open Source Produkten und legen daher ihre Formatspezifikationen und Schnittstellen nicht offen. Dies führt dann teilweise dazu, dass ein Datenaustausch zwischen kommerzieller und Open Source Software schwierig ist, oder gar ganz unmöglich wird. Bekanntestes Beispiel für dieses Verhalten sind die Microsoft Office Dateiformate.

Letztendlich ist dieser Nachteil aber kein Nachteil von Open Source Software, sondern ein Nachteil der kommerziellen Software, deren Schnittstellen nicht offen gelegt werden. In manchen Fällen könnte dies vielmehr ein Grund sein, für die die Zusammenarbeit verweigernde Software ebenfalls geeignete Open Source Alternativen zu suchen.

#### **6.1.3 Wertung**

Insgesamt sind die Vorteile von Open Source Software als deutlich bedeutsamer zu betrachten, als die Nachteile. Open Source Software sollte daher durchaus als ein strategischer Faktor in der IT Landschaft verstanden werden.

Aus dieser grundlegenden Feststellung lässt sich freilich noch keine konkrete Einsatzempfehlung für bestimmte Bereiche herleiten. Dies soll im Rahmen des folgenden Abschnittes vorgenommen werden.

### **6.2 Portfolio Analyse**

In diesem Abschnitt werden generelle Empfehlungen bezüglich des Einsatzes von Open Source Software ausgesprochen. Dazu wurde aus den Daten der Ist-Analyse ein Relevanz-Reifegrad Portfolio abgeleitet, in dem sowohl die in der Befragung erhobene Bedeutung der einzelnen Einsatzgebiete<sup>185</sup>, als auch die relative Reife der vorhandenen Open Source Software aufgezeichnet ist.

Der Reifegrad lässt sich ermitteln aus den fünf Teilkomponenten

- Reifegrad der Funktionalität
- Reifegrad der Benutzerfreundlichkeit
- Reifegrad der Zuverlässigkeit / Sicherheit
- Reifegrad der Leistungsfähigkeit
- Reifegrad des Supports

<sup>185</sup> Es wurden die Bedeutungswerte aus der in der Studie dargestellten Befragung zugrunde gelegt. Diese sind selbstverständlich nicht für alle Unternehmen identisch.

Die Gewichtung wurde aus den Top Two Werten der Antworten auf die Frage nach der Wichtigkeit der fünf Teilkriterien abgeleitet. Es ergab sich folgende Gewichtungstabelle:

<b>Kriterium</b>	<b>Gewicht</b>
Zuverlässigkeit	0,39
Funktionalität	0,32
Benutzerfreundlichkeit	0,18
Leistung	0,07
Support	0,05

Die Wichtigkeit der jeweiligen Softwarearten wurde in der Befragung direkt erhoben und konnte somit direkt in das Portfolio einfließen.

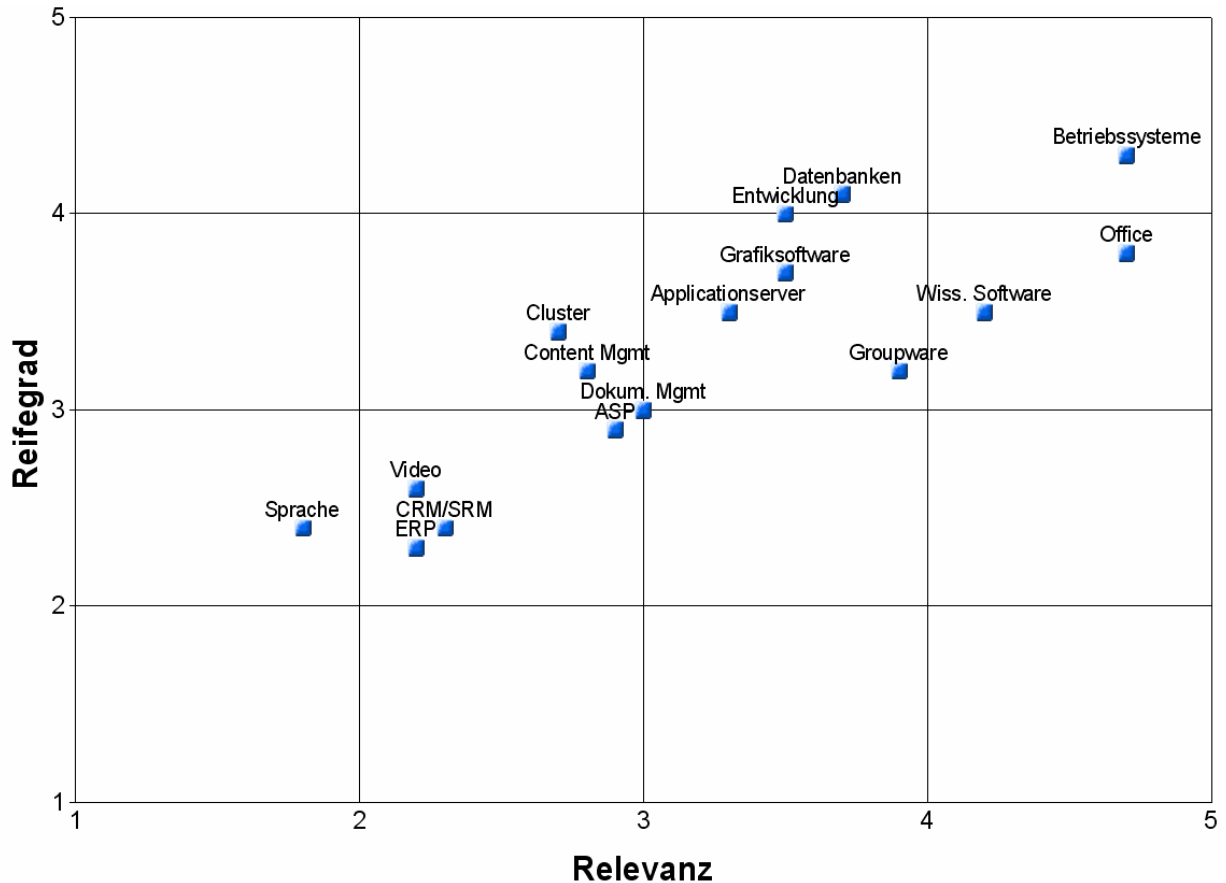


Abbildung 33: Das Relevanz-/Reifegrad-Portfolio der Open Source Software

Folgende Schlüsse können aus dem Portfolio gezogen werden:

- Im oberen, rechten Quadranten des Portfolios eingetragenen Einsatzgebiete weisen eine hohe Relevanz auf und sind gleichzeitig durch einen hohen Reifegrad von Open Source Software in diesem Bereich gekennzeichnet. Der Einsatz freier Software sollte daher in diesen Bereichen flächendeckend in Erwägung gezogen werden. Bedingt durch die hohe strategische Relevanz dieser Einsatzgebiete sind darüber hinaus gehende Maßnahmen zu überlegen. Beispielsweise könnten noch vorhandene Lücken in der Funktionalität freier Software durch gezielte eigene Entwicklungsbeiträge geschlossen werden. Die hier zu verfolgende Strategie könnte mit »Freie Software einsetzen und vorantreiben« umschrieben werden (Strategiegebiet I).

- Der obere, linke Quadrant kennzeichnet Einsatzgebiete mit hoher Verfügbarkeit von ausgereifter freier Software, aber gleichzeitig unterdurchschnittlicher Relevanz. In Fällen, wo Software in diesem Bereich benötigt wird, sollte Open Source Software in Erwägung gezogen werden. Aufgrund der geringen Relevanz sind diese Gebiete aber vermutlich weder durch eine hohe strategische Bedeutung noch durch hohe Einsparpotenziale gekennzeichnet. Weitere Maßnahmen neben dem reinen Einsatz von vorhandener Open Source Software sind daher wohl eher nicht zu befürworten (Strategiegebiet II: »Freie Software einsetzen«).
- Der untere, rechte Quadrant identifiziert Einsatzgebiete mit hoher Relevanz und gleichzeitigem Fehlen ausgereifter Open Source Software. Hier handelt es sich um Einsatzgebiete mit hoher strategischer Relevanz und hohen Einsparpotenzialen. Für diesen Bereich wäre eventuell eine Weiterentwicklung freier Software im Hinblick auf die fehlenden Funktionalitäten oder eine gezielte Förderung vorhandener Projekte denkbar. In der Zwischenzeit ist der weitere Einsatz kommerzieller Software die vermutlich beste Lösung (Strategiegebiet III: »Freie Software vorantreiben«).
- Der untere, linke Quadrant kennzeichnet Softwarearten mit geringerer Bedeutung und gleichzeitigem Fehlen von ausgereifter Open Source Software. Die strategische Bedeutung dieser Einsatzgebiete ist für die untersuchte Institution gering, die Einsparpotenziale vermutlich begrenzt. Hierbei ist zu beachten, dass sich die niedrige Relevanz von Enterprise Resource Planning Systemen nur auf den Einsatz in Kundenprojekten bezieht. Eine Eigenentwicklung oder die Förderung entsprechender Projekte ist daher nicht zu empfehlen. In diesen Gebieten sollte bei Bedarf auch weiterhin kommerzielle Software eingesetzt werden (Strategiegebiet IV: »Kein Handlungsbedarf«).

### 6.2.1 Strategiegebiet I: Freie Software einsetzen und vorantreiben

In dieses Handlungsfeld fallen folgende Softwarearten:

- Betriebssysteme
- Office Pakete
- Datenbanken
- Entwicklungswerkzeuge
- Wissenschaftlich-Technische Software
- Grafiksoftware
- Application Server
- Groupware

Diese Softwarearten haben aufgrund ihrer Relevanz eine strategische Bedeutung. Viele dieser Softwarearten sind zudem durch hohe Einsatzhäufigkeiten

gekennzeichnet. Das hohe Volumen könnte sich wiederum auf das Einsparpotenzial durch den Einsatz freier Software auswirken. Für diese Gebiete sollten daher zeitnah weitere Untersuchungen durchgeführt werden. Diese sollten vor allem folgende Aspekte beleuchten:

- Detaillierte Aufnahme von Anforderungen an Software im jeweiligen Bereich
- Abgleich der Anforderungen mit vorhandener freier Software
- Konzeption von konkreten Einführungs- oder Migrationsszenarien
- Durchführung von detaillierten Wirtschaftlichkeits- und Nutzwertanalysen für jeden dieser Bereiche
- Erarbeitung von Leitfäden und Einsatzszenarien für freie Software
- Unter Umständen auch Zusammenstellung von Know-how und Erfahrungsberichten – dies könnte etwa auch den dauerhaften Betrieb einer unternehmensweiten Informationsaustauschplattform beinhalten.

Des Weiteren sind möglicherweise flankierende Maßnahmen zu überlegen, wie z.B. die Förderung konkreter Open Source Projekte in diesen Gebieten. Dies könnte sowohl eine finanzielle oder ideelle Förderung als auch eine direkte Mitarbeit an den Entwicklungsarbeiten beinhalten. Dies würde es den Anwendern ermöglichen, eventuell noch benötigte Funktionen in die Software einzubringen.

Gegebenenfalls kann auch eine Mitarbeit in übergeordneten Gremien und Verbänden erwogen werden, die sich mit freier Software beschäftigen. Durch diese Mitarbeit können eventuell weitere Punkte identifiziert werden, in denen Handlungsbedarf besteht.

### 6.2.2 Strategiegebiet II: Freie Software einsetzen

In dieses Handlungsfeld fallen folgende Softwarearten:

- Web Content Management Systeme
- Cluster und Grid Software

In diesen Fällen ist der Einsatz freier Software zu empfehlen. Diese Softwarearten sind jedoch tendenziell weniger bedeutsam, so dass auf weitergehende Maßnahmen eher verzichtet werden sollte.

Zu den Möglichkeiten des Einsatzes sollten analog des Strategiegebiets I weitergehende Untersuchungen durchgeführt werden.

### 6.2.3 Strategiegebiet III: Freie Software vorantreiben

Die Portfolioanalyse zeigte keine Softwareart auf, die deutlich in diese Kategorie fällt. Am ehesten ist der Bereich Dokumentenmanagement Systeme hierunter zu sehen.

Aufgrund der recht hohen strategischen Bedeutung dieser Software wäre zu überlegen, geeignete Open Source Projekte zu identifizieren und deren Software durch gezielte Maßnahmen zur Einsatzreife zu verhelfen. Hierfür sollten geeignete Empfehlungen in einer weiterführenden Untersuchung erarbeitet werden.

#### **6.2.4 Strategiegebiet IV: Kein Handlungsbedarf**

In dieses Handlungsfeld fallen folgende Softwarearten:

- Application Service Providing Software
- Videosoftware
- Enterprise Ressource Planning Systeme
- CRM / SRM Systeme
- Sprachverarbeitung

Diese Softwarearten sind für die untersuchte Institution nicht strategisch bedeutsam genug, um ein konkretes Engagement für Open Source Projekte in diesen Bereichen zu rechtfertigen. Zu beachten ist, dass sich die niedrige Relevanz von Enterprise Ressource Planning Systemen nur auf den Einsatz in Kundenprojekten bezieht. Diese Bewertung kann nicht auf Unternehmen übertragen werden. Gleichzeitig ist vorhandene Open Source Software im Regelfall nicht ausgereift genug, um einen Einsatz oder eine Migration empfehlen zu können.

Für diese Einsatzgebiete ist es daher wohl zu empfehlen, vorläufig auch weiterhin kommerzielle Software einzusetzen. Gleichwohl sollten auch diese Bereiche beobachtet werden, da es in fast allen genannten Gebieten durchaus viel versprechende Open Source Projekte gibt, deren Software eines Tages auch die Anforderungen decken könnten.

### **6.3 Wirtschaftlichkeit**

Der Wegfall der Lizenzkosten führt in vielen Fällen zu einer Senkung des Total Cost of Ownership von Software. Die kommerziellen Anbieter bestreiten diesen Umstand teilweise heftig und argumentieren mit dem relativ geringen Anteil der Lizenzkosten am TCO und den hohen Schulungskosten, die ihrer Meinung nach bei einer Migration hin zu Open Source Software anfallen.

Trotzdem konnte von vielen Studien ein geringerer TCO von Open Source Software bereits bestätigt werden. So stellte Cybersource in einer Vergleichs-

studie zwischen dem TCO von Linux und Windows einen zwischen 19% und 36% niedrigeren TCO von Linux fest<sup>186</sup>.

Wahr ist, dass die Wirtschaftlichkeit von Open Source Software stark vom jeweiligen Einsatzgebiet abhängt. Der TCO von Open Source Software ist im Vergleich zu proprietärer Software insbesondere dann deutlich niedriger, wenn wenig zusätzlicher Schulungsaufwand anfällt und gleichzeitig große Lizenzkostenblöcke eingespart werden können. Soreon stellte daher insbesondere bei Großunternehmen mit hohen Lizenzkosten massive Einsparpotenziale bis zu 30% fest<sup>187</sup>. Finanziell besonders attraktiv sind dabei freie Office Pakete, Serverbetriebssysteme, Content Management Systeme und Datenbanken.

Der Vergleich zwischen einer Linux- und einer Windows-basierten Infrastruktur im Zuge der Migrationsüberlegungen der Stadt München ergab, dass für die rund 14 000 Arbeitsplätze die Kosten beider Lösungen annähernd identisch waren – und das obwohl Microsoft massive Zugeständnisse an die prestigeträchtige bayerische Landeshauptstadt machte<sup>188</sup>.

Dennoch ist festzuhalten, dass die Einsparpotenziale stark mit dem jeweiligen Einsatzszenario zusammenhängen. Während Open Source Software nur in den seltensten Fällen wirklich insgesamt teurer sein wird, als kommerzielle Software, so lässt sich doch nicht in jedem Fall eine drastische Senkung des TCO realisieren. Wichtig ist deshalb, eine konkrete Wirtschaftlichkeitsbetrachtung für die geplanten Einsatzszenarien vorzunehmen. Dass dies auch für kommerzielle Software seltener als nötig durchgeführt wird, darf an dieser Stelle kein Argument sein.

Das in Kapitel 5 dargestellte Vorgehen kann als Leitlinie für Wirtschaftlichkeitsbetrachtungen für Software dienen. Die dabei eingesetzte Methodik lässt sich durchaus auch auf andere Migrations- oder Einföhrungsszenarien übertragen.

<sup>186</sup> [http://www.cybersource.com.au/about/linux\\_vs\\_windows\\_tco\\_comparison.pdf](http://www.cybersource.com.au/about/linux_vs_windows_tco_comparison.pdf)

<sup>187</sup> Vgl. Binder, Steffen; Henke, Melanie: Kassensturz. Open Source und proprietäre Software in Vergleich. Frankfurt: Soreon Research, 2003

<sup>188</sup> [http://www.muenchen.de/vip8/prod2/mde/\\_de/rubriken/Rathaus/40\\_dir/limux/publikationen/clientstudie\\_kurz.pdf](http://www.muenchen.de/vip8/prod2/mde/_de/rubriken/Rathaus/40_dir/limux/publikationen/clientstudie_kurz.pdf)

## 7 Zusammenfassung

Open Source Software hat bereits einen festen Platz in der Softwarelandschaft gefunden. Open Source Software wird gegenwärtig insbesondere in den Gebieten Datenbanken, Entwicklungssysteme, Web-Browser und –Server, Content Management Systeme sowie Cluster und Grid Software eingesetzt.

Open Source Software ist in diesen und auch in anderen Einsatzbereichen eine Alternative zu herkömmlicher kommerzieller Software. Es konnte gezeigt werden, dass sich freie Software in vielen Bereichen qualitativ hinter der kommerziellen Konkurrenz nicht zu verstecken braucht. Freie Betriebssysteme, Office Pakete, Entwicklungssysteme, Datenbanken und Content Management Systeme – um nur einige Beispiele zu nennen – erfüllen die qualitativen Anforderungen ebenso gut wie die oft viel teureren proprietären Produkte.

Kostenaspekte sind ein Kriterium, das tendenziell eher für den Einsatz freier Software spricht, aber nicht das einzige. Darüber hinaus bringt der Einsatz freier Software auch strategische Vorteile mit sich. Diese sollten gegenüber dem – zweifellos wichtigen – Kriterium der Wirtschaftlichkeit nicht vernachlässigt werden. Die sonstigen Vorteile, die aus dem Einsatz freier Software realisiert werden können, rechtfertigen den Einsatz von Open Source Software möglicherweise auch dann, wenn die finanziellen Einsparpotenziale lediglich aus einer »schwarzen Null« bestehen. In vielen Fällen sind Einsparungen möglich.

Wichtig ist eine genaue Untersuchung des jeweiligen Einsatzgebietes, da Open Source Software stark unterschiedliche Reifegrade aufweist. Derzeit gibt es einige Softwarebereiche, in denen noch keine ausreichend ausgereiften Open Source Produkte existieren. Die Entwicklung lässt aber den Schluss zu, dass die Anzahl dieser Fälle zukünftig abnehmen wird. In einigen Bereichen, in denen Open Source Software derzeit noch hinter den kommerziellen Produkten zurückfällt, sind viel versprechende Open Source Projekte aktiv.

Die Frage, ob sich Open Source Software gegenüber kommerzieller Software durchsetzen können wird, stellt sich in dieser Form nicht mehr. Sie hat es in vielen Bereichen bereits erreicht. Für die Zukunft ist mit einer weiteren Verbreitung von Open Source Software zu rechnen, auch und gerade in den Bereichen, in denen sie bisher unterrepräsentiert ist. Das Gewicht der Nachteile von Open Source Software – von denen viele auf den derzeitigen, noch häufig überschaubaren Marktanteil von Open Source Software zurückzuführen sind – wird dadurch gegenüber den erheblichen Vorteilen immer geringer.

## Weiterführende Literatur

**Bärwolff, Matthias; Gehring, Robert A.; Lutterbeck, Bernd (Hrsg.):** Open Source Jahrbuch 2005. Zwischen Softwareentwicklung und Gesellschaftsmodell. Berlin: Lehmanns Media, 2005.  
([http://www.opensourcejahrbuch.de/2005/pdfs/OpenSourceJahrbuch2005\\_online.pdf](http://www.opensourcejahrbuch.de/2005/pdfs/OpenSourceJahrbuch2005_online.pdf))

**Berlecon Research:** Basisreport Linux und Open Source Strategien für CIOs. Berlin: Berlecon, 2004.

**Binder, Steffen; Henke, Melanie:** Kassensturz. Open Source und proprietäre Software in Vergleich. Frankfurt: Soreon Research, 2003.

**Brügge, Bernd et al:** Open-Source-Software. Eine ökonomische und technische Analyse. Berlin: Springer Verlag, 2004.

**Cybersource:** Linux vs. Windows. Total Cost of Ownership Comparison, 2004  
(<http://www.cybersource.com.au>)

**Fink, Martin:** The business and economics of Linux and open source. Upper Saddle River: Prentice Hall, 2003.

**Gehring, Robert A.; Lutterbeck, Bernd (Hrsg.):** Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell. Berlin: Lehmanns Media, 2004. (<http://ig.cs.tu-berlin.de/osjb/OpenSourceJahrbuch2004.pdf>)

**Gläßler, Lothar:** Open Source Software. Projekte, Geschäftsmodelle, Rechtsfragen, Anwendungsszenarien. Erlangen: Publicis Corporate Publishing, 2004.

**Jäger, Till; Metzger, Axel:** Open Source Software. Rechtliche Rahmenbedingungen der freien Software. München: Beck, 2002.

**Krumbein, Thomas:** Open Source Software einsetzen und integrieren. Bonn, Galileo Press, 2004.

**Leipelt, Detlef:** Grundgedanken zu Wirtschaftlichkeitsbetrachtungen für den Einsatz von Open Source Software.  
(<http://www.kbst.bund.de/Anlage305827/Grundgedanken-zu-Wirtschaftlichkeitsbetrachtungen-fuer-den-Einsatz-von-OSS-pdf-13-8-kB.pdf>)

**Marx, Ben (Hrsg):** Linux Manager Guide. Nürnberg: SuSE Linux AG, 2003.

**Netproject:** The IDA Open Source Migration Guidelines. Oo: Netproject, 2003.

**Stadt Wien:** Studie OSS. Open Source Software am Arbeitsplatz im Magistrat Wien.

**Statskontoret:** Free and Open Source Software – a feasibility study. Stockholm: Swedish Agency for Public Management, 2003.

**Unilog Integrata:** Client Studie der Landeshauptstadt München. Kurzfassung. München, 2004.  
([http://www.muenchen.de/vip8/prod2/mde/\\_de/rubriken/Rathaus/40\\_dir/limux/publikationen/clientstudie\\_kurz.pdf](http://www.muenchen.de/vip8/prod2/mde/_de/rubriken/Rathaus/40_dir/limux/publikationen/clientstudie_kurz.pdf))

**Weber, Steven:** The success of open source. Cambridge: Harvard Press, 2004.

Open Source Software (OSS) ist mittlerweile in vielen Bereichen eine realistische Alternative zu kommerziellen Produkten geworden. Sie wird in Unternehmen jeder Größenordnung und zunehmend auch in öffentlichen Institutionen eingesetzt, nicht nur wegen kostenfreier Lizenzierungsmodelle. Qualität und Quantität von Open Source Lösungen haben in den letzten Jahren deutlich zugelegt. Neben dem Betriebssystem »Linux« etablieren sich immer mehr Open Source Produkte.

Die Studie stellt grundlegende Eigenschaften der Open Source Software vor und beleuchtet deren Vor- und Nachteile aus der Sicht von Anwendern und Entwicklern. Anhand einer Erhebung, die in der Fraunhofer-Gesellschaft durchgeführt wurde, werden Verbreitung und Einsatz von OSS in einer Großorganisation exemplarisch untersucht. Für wichtige Anwendungsfelder werden relevante kommerzielle und freie Softwareprodukte gegenübergestellt. Ein methodisches Vorgehen zur Erstellung von Wirtschaftlichkeitsbetrachtungen für den Einsatz von Open Source Software wird dargestellt. Die Studie bietet eine wertvolle Hilfestellung für Entscheider und Anwender, die die Potenziale von Open Source Software für sich nutzbar machen wollen.