

# Diplomarbeit

## LiMux

Ist Open-Source-Software eine Alternative für  
die öffentliche Verwaltung?

Überprüfung des Migrationsprozesses der Stadt München  
von Windows auf Linux anhand eines Kriterienkatalogs

Erstellt am:  
Arbeitsbereich Softwaretechnik  
Departement Informatik  
MIN Fakultät  
**Universität Hamburg**

**Erstbetreuer: Dr. Wolf-Gideon Bleek**  
Zweitbetreuer: Prof. Dr.-Ing. Bernd Page

Abgabedatum: 02.05.2007

Erstellt von:

**Benjamin Stukenberg**  
Matrikelnummer: 5498671  
Studiengang: Wirtschaftsinformatik  
Adresse: Rantzastr. 85, 22926 Ahrensburg



## Danksagung

Ich danke meinem Erstbetreuer Dr. Bleek für die Betreuung und für Anregungen für meine Diplomarbeit und meinem Zweitbetreuer Prof. Dr.-Ing. Page für die Zweitbetreuung.

Ich danke der Stadt München dafür, dass sie auf Open-Source-Software umsteigt und mir somit das Thema meiner Diplomarbeit ermöglicht hat. Insbesondere danke ich dem stellvertretenden Projektleiter des LiMux-Projekts Herrn Lubig-Konzett für das Interview.

Des weiteren danke ich allen Entwicklern von Open-Source-Software, ohne sie wäre diese Diplomarbeit nicht möglich gewesen, d. h. ohne sie hätte ich nicht über dieses Thema schreiben können. Insbesondere danke ich den Entwicklern von OpenOffice.org und Linux, denn ich habe diese Diplomarbeit mit OpenOffice.org unter Linux ohne einen einzigen Absturz verfassen können.

## Inhaltsverzeichnis

Tabellenverzeichnis.....	6
Abbildungsverzeichnis.....	6
Abkürzungsverzeichnis.....	6
1. Einleitung.....	8
2. Was ist Open-Source?.....	10
2.1. Geschichte von Open-Source-Software.....	11
2.2. Prominente Open-Source-Software.....	11
2.2.1. Linux.....	12
2.2.2. OpenOffice.org.....	12
2.2.3. Mozilla.....	13
2.2.4. GIMP.....	13
2.2.5. Apache.....	13
2.2.6. Samba.....	14
2.3. Open-Source-Lizenzen.....	14
2.3.1. GPL.....	14
2.3.2. LGPL.....	14
2.3.3. BSD-Lizenz.....	15
2.3.4. Weitere Lizenzen.....	15
2.4. Open-Source vs. Closed-Source.....	15
2.4.1. Stärken von Open-Source-Software.....	15
2.4.2. Schwächen von Open-Source-Software.....	16
2.4.3. Vergleich von Open-Source und Closed-Source.....	17
3. Konzepte für Upgrade- und Migrationsprozesse.....	18
3.1. Migrationsphasen.....	18
3.2. Migrationswege.....	19
3.2.1. Schnelle Migration.....	19
3.2.2. Sanfte Migration.....	19
3.3. Kostenfaktoren.....	20
3.4. Ausschreibung.....	20
3.5. Fazit Migrationsprozesse.....	21
4. Kriterien für eine Software-Migration.....	22
4.1. Kriterienkatalog.....	22
4.1.1. Strategie.....	23
4.1.2. Kosten.....	23
4.1.3. Software.....	24
4.1.4. Personal.....	26
4.1.5. IT-Abteilung.....	26
4.1.6. Politik.....	27
4.2. Fazit Kriterien für eine Software-Migration.....	27
5. Open-Source in der öffentlichen Verwaltung.....	28
5.1. Schwäbisch Hall.....	28
5.2. Treuchtlingen.....	29
5.3. Wienux.....	29
5.4. NIVADIS.....	30
5.5. Fazit Open-Source in der öffentlichen Verwaltung.....	31
6. Das LiMux-Projekt.....	32
6.1. Ausgangszustand.....	32
6.2. Auswahlprozess.....	33
6.2.1. Qualitative Betrachtungen.....	34

---

6.2.2. Wirtschaftlichkeitsbetrachtung.....	35
6.2.3. Strategische Auswirkungen.....	38
6.2.4. Empfehlung.....	40
6.2.5. Entscheidung des Stadtrats.....	42
6.3. Implementierungsprozess.....	42
6.3.1. Akzeptanzförderung.....	43
6.3.2. LiMux-Basis-Client.....	43
6.3.3. WollMux.....	44
6.3.4. Fachanwendungen.....	45
6.3.4.1. Übergangsszenario.....	46
6.3.4.2. Zwischenlösung.....	47
6.3.4.3. Migrationslösung.....	47
6.4. Aktueller Zustand des Projekts.....	47
6.5. Anwendungen des Kriterienkatalogs auf das LiMux-Projekt.....	48
6.5.1. Strategie.....	48
6.5.2. Kosten.....	50
6.5.3. Software.....	51
6.5.4. Personal.....	53
6.5.5. IT-Abteilung.....	53
6.5.6. Politik.....	54
6.6. Chancen und Risiken.....	55
6.6.1. Chancen.....	55
6.6.2. Risiken.....	56
7. Fazit.....	57
Anhang.....	59
Interview vom 15.11.2006 mit Manfred Lubig-Konzett (stv. Projektleiter LiMux).....	59
Literaturverzeichnis.....	63

## Tabellenverzeichnis

Tabelle 1: Vergleich von Open-Source-Lizenzen.....	15
Tabelle 2: Vergleich von Open-Source und Closed-Source.....	17
Tabelle 3: Klassifizierung der in Wien eingesetzten Software.....	30
Tabelle 4: Migrationsalternativen der Stadt München.....	34
Tabelle 5: Kostenaufschlüsselung Windows XP-Client-Alternativen.....	36
Tabelle 6: Kostenaufschlüsselung Linux-Client-Alternativen.....	37
Tabelle 7: Strategische Kriterien.....	39
Tabelle 8: Ergebnis der qualitativ-strategischen Betrachtung.....	40
Tabelle 9: Kapitalwert pro Nutzwert.....	41
Tabelle 10: LiMux - Kriterienkatalog – Strategie.....	49
Tabelle 11: LiMux - Kriterienkatalog – Kosten.....	50
Tabelle 12: LiMux - Kriterienkatalog – Software.....	52
Tabelle 13: LiMux - Kriterienkatalog – Personal.....	53
Tabelle 14: LiMux - Kriterienkatalog – IT-Abteilung.....	54
Tabelle 15: LiMux - Kriterienkatalog – Politik.....	55

## Abbildungsverzeichnis

Abbildung 1: Migrationsszenarien der Fachverfahren.....	46
---------------------------------------------------------	----

## Abkürzungsverzeichnis

Abkürzung	Bedeutung	Erläuterung
API	Application Programming Interface	Programmierschnittstelle, die ein Programm anbietet, damit andere Programme mit ihm kommunizieren können (vgl. [BroCIT03], S. 59).
BSD <sup>1</sup>	Berkeley Software Distribution	Eine Quelltext-offene Unix-Version, die ursprünglich an der University of California in Berkeley entwickelt wurde (vgl. ebd. S. 137).
FAI <sup>2</sup>	Fully Automatic Installation	Ein Programm, zum automatischen Installieren und Konfigurieren von Debian-Linux.
GIMP <sup>3</sup>	GNU Image Manipulation Program	Ein freies Bildbearbeitungsprogramm.
GNOME <sup>4</sup>	GNU Network Object Model Environment	Eine graphische Oberfläche für Unix- und Linux-Systeme
GNU	GNU is Not Unix	Rekursives Akronym, zur Abgrenzung zum unfreien Unix (vgl. ebd., S. 393).
GPL	Gnu General Public License	Eine freie Software-Lizenz (vgl. [DuInf], 472).

1 <http://www.bsd.org/> (zuletzt besucht am: 24.04.2007)

2 <http://www.informatik.uni-koeln.de/fai/> (zuletzt besucht am: 24.04.2007)

3 <http://www.gimp.org/> (zuletzt besucht am: 24.04.2007)

4 <http://www.gnome.org/> (zuletzt besucht am: 24.04.2007)

<b>Abkürzung</b>	<b>Bedeutung</b>	<b>Erläuterung</b>
(n.) hw.	<b>(nicht) haushaltswirksam</b>	Aufwand, der (nicht) mit tatsächlichen Ausgaben verbunden ist.
ISO <sup>5</sup>	International Organization for Standardization	Eine internationale Standardisierungsorganisation.
IT	<b>Informationstechnik</b>	Ein Sammelbegriff, welcher sowohl Computer, Soft- und Hardware als auch z. B. Datenbanken umfasst (vgl. [BroCIT03], S. 482).
KDE <sup>6</sup>	<b>K Desktop Environment</b>	Eine graphische Oberfläche für Unix- und Linux-Systeme.
LDAP	<b>Lightweight Directory Access Protocol</b>	Ein Verzeichnisdienst, z. B. für Benutzerrechte (vgl. [DuInf], S. 162).
LGPL	<b>GNU Lesser General Public License</b>	Eine freie Software-Lizenz, welche das Verbinden mit unfreier Software gestattet (vgl. ebd., S. 472).
OASIS <sup>7</sup>	<b>Organization for the Advancement of Structured Information Standards</b>	Eine Standardisierungsorganisation für Datenstandards.
ODF <sup>8</sup>	<b>OASIS Open Document Format for Office Applications</b>	Ein offener Dokumentenstandard der OASIS, welcher u. a. von OpenOffice.org verwendet wird.
OSS	<b>Open-Source-Software</b>	Freie, Quelltext-offene Software (vgl. [BroCIT03], S. 657).
SMB	<b>Server Message Block</b>	Ein proprietäres Protokoll für Datei- und Druckdienste (vgl. ebd., S. 816).
TS	<b>Terminal Server</b>	Ein Server, der Anwendungen für Clients bereitstellt. Die Clients übernehmen nur die graphische Darstellung (vgl. ebd., S. 889f.).
VM	<b>Virtual Machine</b>	Eine Software, die einen Computer emuliert, in der ein Betriebssystem laufen kann (vgl. ebd., S. 953).
WINE <sup>9</sup>	<b>Wine Is Not an Emualtor</b>	Ein Programm, das versucht die Windows API nachzubilden, sodass Windows-Programme z. T. auch unter Linux betrieben werden können.

5 <http://www.iso.org/> (zuletzt besucht am: 24.04.2007)

6 <http://www.kde.org/> (zuletzt besucht am: 24.04.2007)

7 <http://www.oasis-open.org/> (zuletzt besucht am: 24.04.2007)

8 <http://www.oasis-open.org/committees/office> (zuletzt besucht am: 24.04.2007)

9 <http://www.winehq.com/> (zuletzt besucht am: 24.04.2007)

## 1. Einleitung

In den letzten Jahren ist immer häufiger zu lesen, dass Unternehmen und öffentliche Einrichtungen auf Open-Source-Software setzen. Es gibt unterschiedliche Gründe, warum Open-Source-Software gewählt wird. Für die einen stehen die nicht vorhandenen Lizenzgebühren im Vordergrund, für andere ist die Unabhängigkeit von einem monopolistischen Anbieter und die Freiheit von Open-Source-Software, diese im Notfall selbst verändern zu können, entscheidend. Andere wiederum ziehen Open-Source-Software in Erwägung, weil der Support für ihre bestehenden proprietären Systeme eingestellt wird. Letzteres ist zum Beispiel bei der Stadtverwaltung der Bayrischen Landeshauptstadt München der Fall.

Bei der Münchner Stadtverwaltung besteht ein Migrationsdruck, weil die Firma Microsoft den Support für die in München eingesetzte Software Windows NT und Office 97 eingestellt hat und die Stadt München somit gezwungen ist, eine Migration auf modernere Software durchzuführen. Eine solche Migration ist mit erheblichen Kosten verbunden, denn modernere Software benötigt fast immer modernere Hardware, sodass nicht nur die Kosten für die Software anfallen (Lizenzkosten bei proprietärer Software), sondern auch für neue Hardware. Aber weitaus schwerer wiegen bei einer Softwaremigration die Kosten, die durch die Softwareumstellung selbst entstehen. Es muss die Migration vorbereitet werden, die neue Software muss an die eigenen Bedürfnisse angepasst werden und der Wechsel auf die neue Soft- und Hardware muss vollzogen werden, was erhebliche Kosten verursacht. Außerdem muss das Personal für die neue Software geschult werden. Darüber hinaus müssen möglicherweise die bestehenden Arbeitsprozesse modifiziert werden, damit die neuen Systeme sinnvoll genutzt werden können, was ebenfalls ein Aufwand ist, der weitere Kosten verursacht.

Um in Zukunft die Kosten einer Migration von einer Version auf eine andere zu minimieren, soll in München Open-Source-Software zum Einsatz kommen, sodass z. B. zukünftig keine Lizenzkosten mehr anfallen. Darüber hinaus können zukünftige Migrationen stärker selbst bestimmt durchgeführt werden und müssen sich nicht mehr nach den Vorstellungen eines einzelnen Anbieters richten.

Im Rahmen dieser Diplomarbeit soll erörtert werden, ob Open-Source-Software eine Alternative für die öffentliche Verwaltung sein kann. Dabei sollen die Vor- und Nachteile einer Migration von proprietärer auf Open-Source-Software abgewogen werden. Als Kriterien sollen sowohl die Migrationskosten und die laufenden Kosten dem Nutzen gegenüber gestellt werden als auch die strategischen Auswirkungen (z. B. Unabhängigkeit von einem Hersteller) ausgearbeitet und bewertet werden. Dazu soll ein Kriterienkatalog erarbeitet werden, welcher die entscheidenden Gesichtspunkte eines solchen Migrationsprozesses umfassen soll. Anhand dieses Kriterienkatalogs wird die Migration der Stadt München, das LiMux-Projekt, kritisch durchleuchtet.

Diese Diplomarbeit erläutert am Anfang, was Open-Source-Software ist. Es werden einige ihrer Vertreter (z. B. Linux) kurz vorgestellt, die für den Erfolg des LiMux-Projektes essenziell sind, und einige Open-Source-Lizenzen erläutert sowie die wesentlichen Unterschiede von Open-Source- und proprietärer Software dargelegt. Anschließend werden Konzepte für Update- und Migrationsprozesse behandelt, der Kriterienkatalog erarbeitet und einige Projekte vorgestellt, bei denen Organe der öffentlichen Verwaltung bereits auf Open-Source-Software migriert sind, beziehungsweise in der nahen Zukunft migrieren werden. Die Auswahl der Projekte soll verdeutlichen, welche Herangehensweisen es an eine Migration auf Open-Source-Software gibt und in wie fern diese mit dem LiMux-Projekt vergleichbar sind.



Der Fokus dieser Diplomarbeit liegt auf dem Kriterienkatalog und der Vorstellung des LiMux-Projekts, das u. a. anhand des Kriterienkatalogs analysiert wird. Es wird auf den Ausgangszustand vor der Migration, den Softwareauswahl- und den Migrationsprozess eingegangen sowie die Chancen und Risiken erörtert.

Zum Schluss wird ein Fazit gezogen, ob und in wie weit Open-Source-Software für die öffentliche Verwaltung geeignet ist und welche Kriterien bei einer solchen Migration besonders wichtig sind, damit diese erfolgreich abgeschlossen werden kann.

## 2. Was ist Open-Source?

Open-Source bedeutet, wörtlich aus dem Englischen übersetzt, „offene Quelle“, es bedeutet, dass der Quelltext von einer Software offen zugänglich ist. Jeder kann sich den Quelltext von Open-Source-Software angucken und diesen bei Bedarf verändern (vgl. [Brockh06], Bd. 20, S. 362).

Ein anderer weitläufig als Synonym benutzter Begriff für Open-Source-Software ist freie Software, wobei das „frei“ nicht zwangsläufig mit kostenlos gleichgesetzt werden muss. Es ist durchaus erlaubt für das Bereitstellen (z. B. brennen auf CDs) von Open-Source-Software Geld zu verlangen (vgl. ebd.).

Von der Open Source Initiative<sup>10</sup> wird Open-Source-Software durch folgende Eigenschaften (vgl. [OsiOsd06]) charakterisiert:

- Die Weitergabe der Software muss frei sein, d. h. jeder darf die Software an jeden beliebigen Anderen weitergeben.
- Der Quelltext muss frei zugänglich sein.
- Jede Änderung der Software muss wieder frei, also Open-Source-Software, sein.
- Keine Diskriminierung von Personen, Gruppen oder Einsatzfeldern für die Software, d. h. jedem und jeder muss es gestattet sein die Software so einzusetzen, wie er oder sie es will.
- Die Open-Source-Lizenz, unter der die Open-Source-Software veröffentlicht wird, muss immer der Software beigelegt sein, damit sich der Anwender über seine Rechte informieren kann.
- Die Open-Source-Lizenz darf nicht produktspezifisch sein oder die Verwendung anderer Software neben der Open-Source-Software einschränken.

Es gibt verschiedene Gründe, warum Software-Entwickler sich entscheiden ihre Software als Open-Source freizugeben.

Altenhöner sieht folgende Gründe, warum Open-Source-Software entwickelt wird: Offenheit, Flexibilität, Geschwindigkeit, Motivation und Standards (vgl. [Alt05] S. 3).

*Offenheit* ist für Altenhöner die traditionelle Weise, wie akademisch gearbeitet wurde und wird. Es ist eine weit verbreitete „Regel“ sich auszutauschen und zu kooperieren.

*Flexibilität* – Viele Open-Source-Software-Projekte werden von Menschen mit verschiedenen Interessenschwerpunkten entwickelt. Open-Source-Software ermöglicht ihnen schnell individuelle Anpassungen vorzunehmen.

*Geschwindigkeit* – Open-Source-Software ist bekannt dafür, dass auftretende Fehler schnell behoben werden. Die Schließung von Sicherheitslücken gelingt in der Regel bedeutend schneller als bei vergleichbarer proprietärer Software.

*Motivation* – Dadurch, dass Open-Source-Software meist in Communities entwickelt wird, entsteht ein Gemeinschaftsgefühl, das maßgeblich zur Motivation beiträgt. Darüber hinaus wird Open-Source-Software nicht nur von Freiwilligen, sondern auch von professionellen Entwicklern programmiert und durch verschiedene Regierungsprogramme gefördert.

---

<sup>10</sup> <http://www.opensource.org/> (zuletzt besucht am: 24.04.2007)

Daten-*Standards* sorgen für eine Unabhängigkeit von einem einzigen Hersteller, d. h., wenn ein Daten-Format standardisiert ist, kann jeder Software entwickeln, die mit diesen Daten umgehen kann. Außerdem sorgen Standards dafür, dass die Daten auch in Zukunft, wenn es das Programm, mit dem sie erstellt wurden, möglicherweise nicht mehr gibt, gelesen und bearbeitet werden können (vgl. [Alt05], S. 3).

## **2.1. Geschichte von Open-Source-Software**

Gemäß Grassmuck ([Gra04], S. 202-205) ist Open-Source keine Erfindung der letzten Jahre. Die Geschichte von Open-Source-Software reicht bis in die Anfänge des Computerzeitalters zurück. In den frühen Jahren (von Ende der 40ern bis in die 70er) war es üblich, neben den kompilierten Programmen, auch den Quelltext von Programmen zu veröffentlichen, was vor allem daran lag, dass es noch keinen eigenständigen Markt für Software gab. Die Einsatzfelder für die frühen Computer beschränkten sich aufgrund ihres Preises, ihrer Größe und der limitierten Leistungsfähigkeit zunächst auf Universitäten und militärische Einrichtungen. Erst später, als der Computer in der Privatwirtschaft größere Verbreitung fand, wurden vermehrt kommerzielle, proprietäre Programme eingesetzt und dominieren seit der Verbreitung des PCs (Personal Computer) den Softwaremarkt.

Im Jahre 1980 entstand die erste Open-Source-Lizenz. AT&T, der Erfinder des Unix-Betriebssystems, entwickelte zusammen mit der Universität von Berkeley die BSD-Lizenz. unter der BSD-Unix<sup>11</sup> veröffentlicht wurde (vgl. ebd., S. 214). Die BSD-Lizenz wird im Kapitel: 2.3.3 näher erläutert.

1984 initiierte Richard Stallman das GNU-Projekt. GNU ist ein rekursives Akronym für GNU is Not Unix (zu deutsch: GNU ist nicht Unix). Ziel Stallmans war es ein zu Unix funktional äquivalentes Betriebssystem zu schreiben, das aber nicht das Schicksal von Unix erleiden würde. Er wollte, dass der Code dieses neuen Betriebssystems immer frei bliebe. Im Gegensatz zu Unix, die u. a. unter BSD-Lizenz entwickelte wurde, sollte es nicht möglich sein den Open-Source-Code in einem proprietären Programm zu verwenden (vgl. ebd., S. 222). So entstand 1989 unter Stallmans Federführung die GPL, die es verbietet Open-Source-Code für proprietäre Software zu verwenden. 1992 stellte Linus Torvalds Linux unter die GPL, was dieser zum Durchbruch verhalf (vgl. ebd., S. 226f.). In Kapitel 2.3.1 wird die GPL und in Kapitel 2.2.1 wird Linux ausführlicher behandelt.

Heutzutage gibt für fast jedes Einsatzfeld eine Open-Source-Lösung: Linux als Betriebssystem, OpenOffice.org als Office-Paket, Firefox als Browser, Thunderbird als E-Mail-Client usw.

## **2.2. Prominente Open-Source-Software**

Um einen Überblick über die wichtigsten Open-Source-Alternativen zu geben, werden in diesem Abschnitt einige Open-Source-Programme kurz vorgestellt. Die für das LiMux-Projekt essenziellen Open-Source-Produkte sind das Betriebssystem Linux, das Office-Paket OpenOffice.org, die Mozilla-Produkte Firefox (Webbrowser) und Thunderbird (E-Mail-Client), sowie das Bildbearbeitungsprogramm GIMP. Darüber hinaus werden die Open-Source-Serverprogramme der Webserver Apache und der Dateiserver Samba bei der Stadt München eingesetzt.

---

11 <http://www.bsd.org/> (zuletzt besucht am: 24.04.2007)

### 2.2.1. Linux

Dieses Unterkapitel erläutert basiert auf dem Brockhaus ([Brockh06], Bd. 17, S. 9f.), was Linux<sup>12</sup> ist und wie es entstanden ist.

Linus Torvalds begann 1991 das Betriebssystem Linux zu entwickeln. Er studierte zu dieser Zeit an der Universität von Helsinki und wollte für seinen privaten Bedarf ein Unix-ähnliches Betriebssystem für den Desktop entwickeln. Mittlerweile hat sich Linux zu einem der erfolgreichsten Open-Source-Projekten entwickelt. Linux läuft auf diversen Hardware-Plattformen (z. B. x86, x86\_64, PowerPC) und ist vielseitig einsetzbar. Linux kann von embedded Devices bis hin zu Mainframes auf fast jedem Computer eingesetzt werden. Heutzutage kann Linux in Verbindung mit den graphischen Oberflächen z. B. KDE<sup>13</sup> (K Desktop Environment) oder GNOME<sup>14</sup> (GNU<sup>15</sup> Network Object Model Environment) komfortabel auf dem Desktop eingesetzt werden. Sowohl KDE als auch GNOME bringen alle notwendigen Werkzeuge mit, die man von einer graphischen Oberfläche erwartet. Es gibt unter anderem einen Dateimanager, eine Task-Leiste und ein Menü, von dem aus Programme gestartet werden können.

Linux kann entweder direkt aus dem Internet heruntergeladen und selbst kompiliert werden oder man greift auf eine so genannte Linux-Distribution zurück. Eine Linux-Distribution besteht in der Regel aus mehreren tausend vorkonfigurierten Programmen, einer Installationsroutine und einem Konfigurationsprogramm. Durch den Einsatz von einfach bedienbaren Installationsprogrammen sind Linux Distributionen mittlerweile von Computer-Laien installierbar. Linux-Distributionen können entweder im Handel in einer Box mit CDs/DVDs und zum Teil mit Handbüchern und Support erworben oder kostenlos aus dem Internet heruntergeladen werden. Linux-Distributoren finanzieren sich hauptsächlich durch den Support für ihre Linux-Distributionen, da die allermeisten Programme einer Linux-Distribution Open-Source-Programme sind.

Einige der am weitesten verbreiteten Linux-Distributionen sind OpenSUSE<sup>16</sup> von Novell<sup>17</sup>, Ubuntu<sup>18</sup>, Fedora<sup>19</sup> von Red Hat<sup>20</sup> und Debian<sup>21</sup> (vgl. [Dist07]). Von Novell und Red Hat gibt es darüber hinaus spezielle Unternehmens-Distributionen (SUSE Linux Enterprise Desktop und Server respektive Red Hat Enterprise Linux), die auf die besonderen Bedürfnisse von Unternehmen zugeschnitten sind. Sie zeichnen sich durch eine konservative Paketauswahl (es werden nicht die allerneuesten Programmversionen verwendet, sondern ausschließlich Programme, die sich als stabil erwiesen haben) und längere Supportzeiträume aus.

### 2.2.2. OpenOffice.org

Im folgenden wird nach Kern ([Ker03] S. 15) erläutert, was OpenOffice.org<sup>22</sup> ist

OpenOffice.org ist ein freies Bürosoftwarepaket. Es besteht aus einer Textverarbeitung (Writer), einer Tabellenkalkulation (Calc), einem Präsentationsprogramm (Impress), einem Vektor-Graphikprogramm (Draw), einem Datenbankprogramm (Base) und einem

---

12 <http://www.kernel.org/> (zuletzt besucht am: 24.04.2007)

13 <http://www.kde.org/> (zuletzt besucht am: 24.04.2007)

14 <http://www.gnome.org/> (zuletzt besucht am: 24.04.2007)

15 GNU ist ein Akronym für „GNU is Not Unix“.

16 <http://www.opensuse.org/> (zuletzt besucht am: 24.04.2007)

17 <http://www.novell.com/> (zuletzt besucht am: 24.04.2007)

18 <http://www.ubuntu.com/> (zuletzt besucht am: 24.04.2007)

19 <http://www.fedoraproject.org/> (zuletzt besucht am: 24.04.2007)

20 <http://www.redhat.com/> (zuletzt besucht am: 24.04.2007)

21 <http://www.de.debian.org/> (zuletzt besucht am: 24.04.2007)

22 <http://de.openoffice.org/> (zuletzt besucht am: 24.04.2007)

Programm für mathematische Formeln (Maths). OpenOffice.org ist sowohl für Linux als auch für Solaris und Windows erhältlich.

OpenOffice.org ist der vollständig quelloffene Ableger von StarOffice von Sun Microsystems<sup>23</sup>. Gegenüber OpenOffice.org enthält StarOffice zusätzlich einen Konverter, um Microsoft Office Makros zu StarOffice/OpenOffice.org Makros zu konvertieren, eine kommerzielle Rechtschreibprüfung und mehr Vorlagen als OpenOffice.org.

StarOffice wurde ursprünglich von dem Hamburger Unternehmen Star Division entwickelt, das 1999 von Sun Microsystems übernommen wurde. Im Jahr 2000 wurde OpenOffice.org von Sun Microsystems als Open-Source-Projekt freigegeben.

### 2.2.3. Mozilla

Im Jahr 1998 wurde von Netscape der Quelltext des gleichnamigen Browser unter dem Namen Mozilla auf der Webseite mozilla.org freigegeben (vgl. [Netsc98]). Im Jahr 2003 wurde die Koordination des Mozilla-Projekts an die eigens zu diesem Zweck gegründete Mozilla Foundation<sup>24</sup> übertragen (vgl. [Moz03]).

Von der Mozilla Foundation gibt es unter anderem den Browser Firefox und das E-Mail-Programm Thunderbird. Firefox<sup>25</sup> zeichnet sich besonders durch seine Benutzerfreundlichkeit aus, z. B. durch Tabbed-Browsing oder einen Pop-Up-Blocker. Thunderbird<sup>26</sup> bietet alles, was man von einem modernen E-Mail-Client erwarten kann, z. B. unterstützt er diverse E-Mail-Protokolle und bringt einen sehr guten Spam-Filter mit.

Darüber hinaus gibt es noch die Mozilla Suite, die mittlerweile unter dem Namen Seamonkey<sup>27</sup> weiterentwickelt wird. Sie besteht aus einem Browser, einem E-Mail-Client, einem HTML<sup>28</sup>-Editor und einem IRC<sup>29</sup>-Client.

### 2.2.4. GIMP

Das Bildbearbeitungsprogramm GIMP<sup>30</sup> (GNU Image Manipulation Program) wurde ursprünglich an der University of California entwickelt und im Jahre 1995 erstmals der Öffentlichkeit vorgestellt (vgl. [KiMa95]).

GIMP unterstützt diverse Bildformate und ermöglicht beispielsweise eine digitale Nachbearbeitung von Fotos (Farben anpassen, skalieren etc.) und das automatische Bearbeiten von mehreren Bildern durch Skripte.

### 2.2.5. Apache

Der Webserver Apache<sup>31</sup> wurde Mitte der 90er Jahre des letzten Jahrhunderts am National Center for Supercomputing Applications der University of Illinois entwickelt. Ende 1995 wurde die Version 1.0 des Apache-Webserver freigegeben (vgl. [Apache05]).

Apache ist mittlerweile der am weitesten verbreitete Webserver mit einem Marktanteil von etwa 60% (vgl. [Netc07]). Der Apache Webserver zeichnet sich durch seine Modu-

23 <http://de.sun.com/> (zuletzt besucht am: 24.04.2007)

24 <http://www.mozilla.org/> (zuletzt besucht am: 24.04.2007)

25 <http://www.firefox-browser.de/> (zuletzt besucht am: 24.04.2007)

26 <http://www.thunderbird-mail.de/> (zuletzt besucht am: 24.04.2007)

27 <http://mozilla.kairo.at/> (zuletzt besucht am: 24.04.2007)

28 HyperText Markup Language

29 Internet Relay Chat

30 <http://www.gimp.org/> (zuletzt besucht am: 24.04.2007)

31 <http://www.apache.org/> (zuletzt besucht am: 24.04.2007)

larität aus. So lassen sich z. B. Module zur Unterstützung von Programmiersprachen einbinden. Ein solches Modul ist beispielsweise Tomcat<sup>32</sup>.

### 2.2.6. Samba

Die Entwicklung des Samba<sup>33</sup>-Servers begann im Jahr 1991 an der Australian National University mit dem Ziel von einem Unix-Computer mit DOS/Windows-Computern kommunizieren zu können (vgl. [Trid97]).

Samba ist ein Dateiserver, welcher das proprietäre Dateitransferprotokoll SMB von Microsoft unterstützt, sodass auch Windows-Clients auf einen Linux- oder Unix-basierten Dateiserver zugreifen können.

## 2.3. Open-Source-Lizenzen

Eine Softwarelizenz ist ein Vertrag zwischen einem Lizenzgeber und einem Lizenznehmer, in dem der Lizenzgeber dem Lizenznehmer an seiner urheberrechtlich geschützten Software bestimmte Nutzungsrechte überlässt und/oder sie beschränkt (vgl. [HaHo03], S. 20).

Es gibt verschiedene Open-Source-Lizenzen, die allerdings alle den Grundsatz haben, dass der Quelltext (zumindest von dem ursprünglichen Programm) offen gelegt werden muss und dass die Software als solche frei ist. Sowohl frei wie in kostenlos als auch frei wie in Freiheit (vgl. ebd., S. 10).

### 2.3.1. GPL

Die am weitesten verbreitete Open-Source-Lizenz ist die GPL<sup>34</sup> (GNU General Public Licence). Die GPL ist eine so genannte Copyleft-Lizenz. Wird ein Programm modifiziert, das unter einer Copyleft-Lizenz steht, so muss das neu erschaffene Derivat wieder unter dieser Copyleft-Lizenz stehen, sodass alle Abkömmlinge einer solchen Software immer frei sein müssen. Das bedeutet, dass z. B. der Quelltext jedweder Modifikation einer GPL-Software zwingend offen gelegt werden muss (vgl. [HaHo03], S. 25).

### 2.3.2. LGPL

Da die GPL das Einschränken der durch sie gewährten Rechte strikt untersagt, gibt es die LGPL<sup>35</sup> (GNU Lesser General Public Licence), die es gestattet Open-Source-Software mit proprietärer Software zu verbinden (vgl. [HaHo03], S. 25f. und [OS99], S.14). Beispielsweise ermöglicht die LGPL, dass eine Software-Bibliothek, die unter der LGPL steht, in einem proprietären Programm eingebunden werden kann, ohne dass das proprietäre Programm automatisch unter der LGPL stehen muss. Stünde die Bibliothek unter der GPL, müsste das Programm, das diese Bibliothek mit einbindet, unter der GPL veröffentlicht werden. Wird die unter der LGPL stehende Bibliothek verändert, so muss allerdings diese Änderung ebenfalls unter der LGPL stehen und der entsprechende Quelltext veröffentlicht werden.

---

32 <http://projects.apache.org/projects/tomcat.html> (zuletzt besucht am: 24.04.2007)

33 <http://de.samba.org/> (zuletzt besucht am: 24.04.2007)

34 <http://www.gnu.org/copyleft/gpl.html> (zuletzt besucht am: 24.04.2007)

35 <http://www.gnu.org/copyleft/lesser.html> (zuletzt besucht am: 24.04.2007)

### 2.3.3. BSD-Lizenz

Eine andere prominente Open-Source-Lizenz ist die BSD<sup>36</sup>-Lizenz (Berkeley Software Distribution). Die BSD-Lizenz gestattet es Open-Source-Software, die unter dieser Lizenz steht, zu modifizieren und/oder mit proprietärer Software zu verbinden, ohne dass das so entstehende Derivat wieder Open-Source sein muss. Die BSD-Lizenz verlangt nur, dass der ursprüngliche Autor erwähnt wird (vgl. [HaHo03], S. 26f. und [OS99], S. 11).

### 2.3.4. Weitere Lizenzen

Neben den GPL-artigen, den LGPL-artigen und den BSD-artigen Lizenzen gibt es noch weitere Open-Source-Lizenzarten, z. B. die NPL<sup>37</sup> (Netscape Public Licence), die dem ursprünglichen Autor gesonderte Rechte einräumt. Netscape behält sich bei der NPL z. B. vor den proprietären Code, den sie mit dem unter NPL stehenden Code verbinden, nicht offen zu legen (vgl. [HaHo03], S. 27f. und [OS99], S. 15).

Darüber hinaus besteht noch die Möglichkeit, dass ein Autor vollständig auf sein Copyright verzichtet, sodass jeder mit seiner Software machen kann, was er will. Diese nennt man dann Public Domain (vgl. [OS99], S. 16).

Lizenz	Kann mit kommerzieller Software verbunden werden	Eigene Veränderungen müssen wieder frei sein	Kann unter anderen Bedingungen veröffentlicht werden	Enthält besondere Rechte für den Lizenzinhaber
GPL	Nein	Ja	Nein	Nein
LGPL	Ja	Ja	Nein	Nein
BSD	Ja	Nein	Nein	Nein
NPL	Ja	Nein	Nein	Ja
Public Domain	Ja	Nein	Ja	Nein

Tabelle 1: Vergleich von Open-Source-Lizenzen. Quelle: [OS99], S. 16

## 2.4. Open-Source vs. Closed-Source

Im folgenden werden die Stärken und Schwächen von Open-Source-Software evaluiert und denen von Closed-Source-Software gegenüber gestellt.

### 2.4.1. Stärken von Open-Source-Software

Der größte Vorteil von Open-Source-Software ist, dass ihr Quelltext offen ist, denn so wird es ermöglicht, dass andere Programmierer den Code überprüfen und ggf. Fehler finden, die dann behoben werden können (vgl. [Wie04], S. 109f.). Durch diese so genannte „peer review“ kann die Qualität der Software gesteigert werden. Dieses Verfahren kommt zudem der Sicherheit der Software zu gute. Beispielsweise sind sichere Verschlüsselungsverfahren nur sicher, wenn sie korrekt implementiert worden sind, was sich nur anhand des Quelltextes überprüfen lässt. Dadurch, dass ein Programmierer von

<sup>36</sup> <http://www.opensource.org/licenses/bsd-license.php> (zuletzt besucht am: 24.04.2007)

<sup>37</sup> <http://www.mozilla.org/MPL/NPL-1.1.html> (zuletzt besucht am: 24.04.2007)

Open-Source-Software weiß, dass andere „seinen“ Quelltext lesen können, ist seine Motivation höher von Anfang an einen qualitativ besseren Code zu schreiben. Außerdem liegt Open-Source-Projekten häufig ein strenges Sicherheitsmodell, z. B. Trennung von Administrator- und User-Rechten, zu Grunde.

Ferner ermöglicht der offene Quelltext, dass andere davon lernen und diesen ggf. für ihr eigenes (Open-Source-)Projekt verwenden können. Bei Open-Source-Software kann eine bewährte Lösung immer wieder verwendet werden, sodass auf dem Vorhandenen aufbaut werden kann, um ein komplexeres Problem zu lösen.

*„Dieses allgemeine Innovationsprinzip, das die technische Entwicklung der Menschheit charakterisierte, ist mit geschlossener Software kaum möglich.“  
Quelle: [Wie04], S. 110*

Ein weiterer Vorteil von Open-Source-Software ist, dass sie generell inkrementell entwickelt wird, d. h. es werden häufig neue Versionen herausgegeben, die nur wenige Änderungen gegenüber der Vorversion aufweisen (vgl. [FeFi01], S. 24). Bei Closed-Source-Software hingegen werden in der Regel bei jeder neuen Version, die auf den Markt gebracht wird, viele neue Funktionen implementiert, was das Risiko von Fehlfunktionen erhöht.

## 2.4.2. Schwächen von Open-Source-Software

Da Open-Source Programmierer eher Technik-fixiert sind, wird häufig von ihnen die Benutzbarkeit und Dokumentation der Software vernachlässigt. Darunter leiden vornehmlich kleine Projekte. Hinter den meisten größeren Projekten stehen Unternehmen, die z. B. durch den Support dieser Software Geld verdienen wollen, was aber nur möglich ist, wenn die Software hinreichend benutzbar und dokumentiert ist. Allerdings werden von diesen Unternehmen nur eine begrenzte Anzahl von Programmen unterstützt, d. h. für viele Open-Source-Programme gibt es keinen kommerziellen Support, sodass man hier bei Bedarf auf den Support der entsprechenden Community im Internet angewiesen ist (vgl. [Wie04], S. 110).

Bei jungen, kleinen Open-Source-Projekte besteht das Problem, dass diese noch nicht den Reifegrad erlangt haben, der den alltäglichen Einsatz erlauben würde (vgl. ebs., S. 111). Allerdings wird bei der Open-Source-Entwicklung zwischen „unstable“ (unausgereiften) und „stable“ (ausgereiften) Softwareversionen zu unterscheiden, sodass erkennbar ist, welche Version nur für Beta-Tester und welche für die Allgemeinheit bestimmt ist. Der hohe Anspruch der Open-Source-Entwickler führt bei einigen Softwareprojekten dazu, dass selbst ausgereifte Programme noch als „unstable“ eingestuft werden, die bei einer kommerziellen Entwicklung schon längst vermarktet würden.



### 2.4.3. Vergleich von Open-Source und Closed-Source

Nach der Vorstellung von Open-Source-Software und der Erörterung seiner Stärken und Schwächen lässt sich folgende Gegenüberstellung von Open- und Closed-Source-Software folgern:

	<b>Open-Source</b>	<b>Closed-Source</b>
Quelltext verfügbar	ja	nein
Frei anpassbar	ja	nein
Kosten für den Erwerb	nicht für die Software selbst	ja
Lizenzkosten	nein	ja (außer Freeware)
Beschränkung auf eine bestimmte Benutzerzahl	nein	ja
Support	ja (z. T. nur Community)	ja (zeitlich begrenzt)
Supportkosten	nein (Community) ja (kommerziell)	ja
Dokumentation (Hilfe)	teilweise vorhanden (häufig nur im Internet)	meistens inklusive (z. B. Handbuch)
Ausgereift	ja (stable) / nein (unstable)	unterschiedlich
Sicherheit (Schutz gegen Schadsoftware)	gut	teilweise sehr schlecht (z. B. virenanfällig)
Peer review	ja	nein
Inkrementelle Entwicklung	ja	nein

*Tabelle 2: Vergleich von Open-Source und Closed-Source*

Dadurch, dass der Quelltext von Open-Source-Software frei verfügbar ist, lässt sich Open-Source-Software leichter anpassen als Closed-Source-Software, was der Qualität und der Sicherheit zugute kommt. Außerdem fallen für Open-Source-Software keine Lizenzkosten an, die Software ist nicht auf eine bestimmte Benutzerzahl beschränkt und es gibt einen kostenlosen Support in Form der Open-Source-Community.

Beim Erwerb von Closed-Source-Software bezahlt man in der Regel auch für den inklusive vorhandenen Support und eine Dokumentation (z. B. ein Handbuch).

Darüber hinaus wird auf Closed-Source-Software zurückgegriffen, wenn 1. keine Open-Source-Alternative vorhanden ist, z. B. wird in München in der stadt-eigenen Druckerstraße weiterhin Windows eingesetzt werden, weil die Spezialsoftware für die Druckerstraße nur für Windows verfügbar ist, wenn 2. die Open-Source-Alternativen nicht den nötigen Funktionsumfang bieten, z. B. wird beim LiMux-Projekt der proprietäre Adobe Reader verwendet, weil nur dieser PDF-Formulare beherrscht oder wenn 3. die Umstiegskosten zu hoch sind. Dazu mehr in den folgenden Kapiteln.

### 3. Konzepte für Upgrade- und Migrationsprozesse

Bevor auf die Konzepte für Upgrade- und Migrationsprozesse eingegangen werden kann, muss erst einmal definiert werden, was man unter diesen Schlagworten versteht. Sie werden vom Migrationsleitfaden der Bundesregierung (vgl. [MiLeiFa06], S. 15) wie folgt definiert:

Bei einem Upgradeprozess (vom Migrationsleitfaden als fortführende Migration bezeichnet) von Computersoftware werden die bestehenden Programme durch neuere Versionen der gleichen Programme ersetzt. Beispielsweise handelt es sich bei einem Umstieg von Microsoft Windows 2000 auf Microsoft Windows XP um einen Upgradeprozess.

Bei einem Migrationsprozess (vom Migrationsleitfaden als ablösende Migration bezeichnet) von Computersoftware werden die bestehenden Programme durch funktionsgleiche oder funktional ähnliche Programme ersetzt. Der Migrationsleitfaden unterscheidet die ablösende Migration in punktuell, partiell und vollständig. Beispielsweise handelt es sich bei einem Umstieg von Microsoft Office auf OpenOffice.org um eine punktuelle, beim Umstellen aller Server von Windows NT auf Linux um eine partielle und bei der Umstellung der gesamten IT-Landschaft von einer Windows-basierten auf eine Linux-basierte um eine vollständige, ablösende Migration.

Beim LiMux-Projekt handelt es sich um eine vollständige, ablösende Migration.

Für eine Migration gibt es laut Migrationsleitfaden (vgl. [MiLeiFa06], S. 21) folgende Gründe:

- Auslaufender Herstellersupport für wesentliche Produkte
- Erweiterte technische Anforderungen
- Konsolidierung der bestehenden Systemlandschaft
- Strategische Ziele, wie beispielsweise verstärkte Herstellerunabhängigkeit oder erhöhte Interoperabilität

#### 3.1. Migrationsphasen

Wenn einer der oben genannten Gründe eintritt, ist es notwendig eine Migrationsstrategie zu entwickeln. Laut Ganten (vgl. [Gan04], S. 249f.) ist es essenziell, bevor mit der Migration begonnen wird eine Ist-Analyse bezüglich der technischen, wirtschaftlichen und personellen Voraussetzungen durchzuführen, sowie eine Aufstellung von Zielen für die Weiterentwicklung der IT-Infrastruktur zu erarbeiten. Eine Migration ohne gewissenhafte Vorbereitung würde im Chaos enden.

Nachdem in der Planungsphase festgestellt worden ist, dass eine Migration prinzipiell möglich ist, folgen die Realisierungs- und Einsatzphase. Die Migrationsphasen sollten laut Migrationsleitfaden (vgl. [MiLeiFa06]) wie folgt gegliedert werden, damit eine die Migration erfolgreich durchgeführt werden kann:

- Planungsphase (Grob- und Feinkonzept)
  - Workshops (z. B. Vorgehen festlegen und Aufwand abschätzen)
  - Ist-Aufnahme (z. B. Anwendungen, Netzwerkinfrastruktur und Betriebsverfahren)
  - Lösungsansätze (z. B. Pflichtenheft erstellen, technische Machbarkeit und Aufbau einer Testumgebung)

- Realisierungsphase (Verfahrenserstellung, -entwicklung und Erprobung und Abnahme)
  - Konzepte, Verfahren, Installation (z. B. Detaillierte Festlegung des Funktionsumfangs, Entwicklung von Installationsverfahren und Softwareverteilung und Rollout-Planung)
  - Erprobung von Verfahren und Überprüfung der Funktion von Installationen (z. B. Pilotanwender umstellen und Lasttests)
- Einsatzphase (Einführung und Betrieb, Roll-Out)
  - Bereitstellung von Funktionen im Netzwerk und Installation in der Fläche (z. B. Inbetriebnahme der Installationsverfahren, Schulungen und Übergabe in den Regelbetrieb)

## **3.2. Migrationswege**

Laut Migrationsleitfaden gibt es grundsätzlich zwei Wege, wie eine Migration durchgeführt werden kann, entweder als schnelle oder als sanfte Migration (vgl. [MiLeiFa06], S. 490-494). Die schnelle Migration wird auch als harte und die sanfte Migration als weiche Migration bezeichnet.

### **3.2.1. Schnelle Migration**

Eine schnelle (harte) Migration (vgl. [MiLeiFa06], S. 490ff.) ist nicht, wie der Name vermuten ließe, durch ihre schnelle Umsetzung definiert, sondern dadurch, dass die gesamte Migration in einem fest definierten Zeitraum durchgeführt wird. Eine schnelle Migration stellt hohe Anforderungen an alle Beteiligten, Finanzen und an die Technik, weil aufgrund des engen Zeithorizonts das gesamte Personal zeitgleich geschult werden muss, die Gelder nicht über mehrere Zeitperioden verteilt ausgegeben werden können, und die gesamte neue Software annähernd gleichzeitig eingeführt werden muss.

Die Vorteile einer solchen Migration sind folgende: Nicht mehrere Systemlandschaften müssen gleichzeitig betreut werden und das Personal muss nur einmal für die gesamte neue Software und nicht bei jeder neuen Softwarekomponente auf die sich neu ergebenden Softwarekombinationen geschult werden.

Aufgrund des hohen Aufwands für eine kurze Zeitperiode ist eine schnelle Migration nur empfehlenswert, wenn die Komplexität der zu migrierenden Systemlandschaft nicht besonders hoch ist, d. h. wenn nur sehr wenige unterschiedliche Fachanwendungen verwendet werden.

### **3.2.2. Sanfte Migration**

Bei einer sanften (weichen) Migration (vgl. [MiLeiFa06], S. 492ff.) wird das Ziel der Migration definiert und der Zeitrahmen nur grob festgelegt. Bei einer solchen Migration werden die verschiedenen Komponenten der IT-Landschaft sukzessive und nur z. T. zeitgleich migriert, was den Aufwand der Migration auf mehrere Zeitperioden verteilt. Es werden in der Regel als erstes die einfachen, nicht organisationskritischen und erst zum Schluss die komplexesten Anwendungen umgestellt.

Vorteile einer sanften Migration sind folgende: Die Migrationsgeschwindigkeit kann an die Haushaltslage angepasst, fehlendes Know-How kann nach und nach erworben, bestehende Widerstände können langsam abgebaut und Vorbehalte aufgelöst und komplexe IT-Strukturen können schrittweise bewältigt werden.

### **3.3. Kostenfaktoren**

Neben der technischen und organisatorischen Umsetzbarkeit sind bei einer Migration die Kosten zu betrachten, die in den unterschiedlichen Phasen (Beschaffung, Einführung, Betrieb und Upgrade/Neubeschaffung) einer Migration anfallen.

Laut der Fraunhofer-Studie [Ren05] zu den Einsatzpotenzialen und der Wirtschaftlichkeit von Open-Source-Software gibt es folgende Kostenfaktoren (vgl. [Ren05], S. 160-164), die bei einer Softwaremigration von Bedeutung sind:

Kosten während der Beschaffungsphase für:

- Hardware
- ggf. Lizenzen
- ggf. Handbüchern und Downloads

Kosten während der Einführungsphase:

- Migrationsplanung
- Softwareverteilung
- Schulung
- Support
- Übernahme von Altbeständen (z. B. Vorlagen und Datensätze)
- Softwareanpassung (z. B. Makros)
- Reduzierte Produktivität während der Umstellung

Kosten des Betriebs:

- Administration
- externer Support

Kosten bei Upgrade und Neubeschaffung für:

- ggf. neue Hardware
- ggf. neue Lizenzen

### **3.4. Ausschreibung**

Außer den Kosten und der Art der Durchführung einer Migration gibt es bei einer Migration in einer öffentlichen Verwaltung noch eine Besonderheit, die Software muss ausgeschrieben werden (vgl. [MiLeiFa06], S. 297f.). Gemäß den Vergabevorschriften für Behörden sind Ausschreibungen von öffentlichen Einrichtungen grundsätzlich neutral zu formulieren, d. h. es darf keine Lösung, z. B. ein bestimmtes Betriebssystem, von vornherein bevorzugt werden. Der Migrationsleitfaden der Bundesregierung empfiehlt anstelle der Bezeichnung Linux-Server oder Open-Source-Software deren Merkmale neutral zu beschreiben, um den Vergabevorschriften nachzukommen. Beispielsweise könnte die Einsichtbarkeit des Quelltextes mit der Begründung verlangt werden, dass ein erhöhtes Maß an Sicherheit verlangt wird, um Backdoors und Virenattacken zu vermeiden.

### **3.5. Fazit Migrationsprozesse**

Damit eine Migration gelingen kann, müssen alle Phasen der Migration gut vorbereitet werden und es sollten alle Kostenfaktoren, insbesondere auch die Betriebskosten und die Kosten einer zukünftigen, erneuten Migration, berücksichtigt werden.

Je nach Komplexität der zu migrierenden IT-Landschaft ist die eine oder andere Migrationsart von Vorteil. Je komplexer die IT-Landschaft, desto günstiger die weiche Migration und je einfacher die IT-Landschaft desto günstiger die harte Migration. Eine pauschale Anleitung, wie eine Migration optimal durchzuführen ist, gibt es nicht.

Bei einer Migration in der öffentlichen Verwaltung ist außerdem noch die Pflicht zur neutralen Ausschreibung zu beachten.

## 4. Kriterien für eine Software-Migration

Eine Migration auf neue Software oder auf eine neue Version einer bestehenden Software sollte auf einer Entscheidung beruhen, die anhand von fundierten Kriterien getroffen wurde. Je besser eine Migration vorbereitet wird, desto höher sind die Chancen, dass diese erfolgreich durchgeführt werden kann.

Ein solcher Kriterienkatalog sollte möglichst umfassend alle Faktoren einer Migration berücksichtigen. Im folgenden Unterkapitel wird ein Kriterienkatalog vorgestellt, anhand dessen eine Migrationsentscheidung evaluiert werden kann. Dieser Kriterienkatalog berücksichtigt insbesondere die Faktoren, die bei einer Software-Migration in einer öffentlichen Verwaltung wichtig sind. Eine Anwendung auf private Organisationen ist denkbar, selbst wenn dort einige der Kriterien nicht zutreffen. Generell sollten die Entscheider über eine Migration (z. B. die Politiker) festlegen, welche Kriterien ihnen besonders wichtig oder unwichtig sind und die Faktoren des Kriterienkatalogs entsprechend gewichten.

Die bedeutendsten Kriterien bei einer Software-Migration in der öffentlichen Verwaltung sind die Kosten (kurz- und langfristige), die Qualität, Verfügbarkeit und Sicherheit der Software, die Fähigkeiten der IT-Abteilung und die Motivation der Mitarbeiter sich auf die neue Software einzulassen, sowie die Herstellerunabhängigkeit. Die Kriterien sind je nach Migration von unterschiedlicher Bedeutung. Beim LiMux-Projekt ist die Herstellerunabhängigkeit das entscheidendste Kriterium zugunsten der Migration auf Open-Source-Software gewesen.

Bei den Kosten ist zu beachten, dass die wirtschaftlichste und nicht unbedingt die billigste Alternative gewählt werden sollte, denn nur, wenn der reibungslose Betrieb mit der neuen Software gewährleistet werden kann, kann eine Migration als erfolgreich gelten. Darüber hinaus muss die IT-Abteilung in der Lage sein eine solche Migration durchzuführen. Außerordentlich wichtig ist es die Mitarbeiter mitzunehmen, wenn diese nicht ausreichend motiviert sind, z. B. durch offensive Informationspolitik und Schulungen, kann eine Migration nicht gelingen. Ein wichtiger Faktor in der öffentlichen Verwaltung ist die Herstellerunabhängigkeit, denn insbesondere hier will man selbst entscheiden können, wann und wie man eine Migration durchführt, und man möchte sich nicht unbegrenzt an einen Hersteller binden. Außerdem ist es in der öffentlichen Verwaltung von Bedeutung, dass Steuergelder bevorzugt für Produkte und Dienstleistungen von kleinen und mittelständische Betrieben aus der Region oder zumindest aus dem selben Land ausgegeben werden.

Damit eine Migration ein Erfolg werden kann, ist es wichtig eine Strategie zu entwickeln, wie die Migration durchzuführen ist und wann die einzelnen Schritte, z. B. Auswahl, Testen und Rollout der Software, vollzogen werden sollen.

### 4.1. Kriterienkatalog

Es folgt der Kriterienkatalog, anhand dessen eine Softwremigration evaluiert werden und eventuelle Schwierigkeiten aufgedeckt werden können. Die Kriterien lassen sich entweder mit ja oder nein beantworten, z .B. „Ist die Migration notwendig?“, oder bedürfen einer einer Bewertung, z. B. „Wie gut ist die Benutzbarkeit der Software?“

Der Kriterienkatalog ist in 6 Oberkategorien gegliedert: Strategie (Art und Weise der Migration), Kosten (kurz- und langfristig), Software (Verfügbarkeit und Qualität), Personal (Motivation), IT-Abteilung (Kompetenz) und Politik (politischer Wille und Ziele).

### **4.1.1. Strategie**

Ohne eine gut durchdachte Strategie ist das Gelingen eines großen Projektes, wie eine vollständige Migration, unmöglich. Deshalb zu aller erst die Kriterien anhand der die Qualität der Migrationsstrategie evaluieren lässt:

#### **Notwendigkeit der Migration**

Ist die Migration notwendig? Ist beispielsweise die Software veraltet oder gibt es keinen Herstellersupport mehr?

#### **Zeitpunkt der Migration**

Konnte der Zeitpunkt gewählt werden? Wenn ja, wurde der Zeitpunkt gut gewählt?

#### **Evaluation von Alternativen**

Wurden alle relevanten Alternativen evaluiert, um die wirtschaftlichste auswählen zu können?

#### **Entscheidung für je eine Alternative**

Wurde für jede essenzielle Software eine Alternative gefunden?

#### **Entwicklung einer Basis-Systemkonfiguration und eines Installationsverfahrens**

Wurde eine Basis-Systemkonfiguration und eine standardisiertes Installationsverfahren erarbeitet, um die Migration reibungslos durchführen zu können?

#### **Testen**

Wurde die Software ausgiebig getestet, ob sie für den organisationskritischen Einsatz geeignet ist?

#### **Pilotanwender / Rollout**

Wurden passende Pilotanwender ausgewählt, sowie die passende Rollout-Strategie entwickelt?

#### **Ziele realistisch definieren**

Sind die Ziele realistisch definiert? Ist es möglich den Zeitplan einzuhalten?

#### **Notfallstrategie (was passiert, wenn etwas nicht wie geplant läuft?)**

Gibt es eine Notfallstrategie? Ist diese Notfallstrategie anwendbar?

#### **Make or buy (selber entwickeln oder einkaufen?)**

Wurde entschieden, welche Software selbstentwickelt und welche Software eingekauft werden soll? Ist diese Entscheidung wirtschaftlich?

### **4.1.2. Kosten**

Eine Softwaremigration ist nur sinnvoll, wenn zumindest langfristig durch sie die Kosten nicht steigen. Eine Entscheidung für eine solche Migration lässt sich nur verteidigen, wenn sie wirtschaftlich ist, deshalb nun die Kriterien für eine Evaluation der Migrationskosten:

#### **Kurzfristige Kosteneinsparung**

Werden durch die Migration kurzfristig Kosten eingespart?

**Lizenzkosten bei der Anschaffung**

Fallen bei der Anschaffung der neuen Software Lizenzkosten an? Können diese Lizenzkosten gegenüber der letzten Beschaffung reduziert werden?

**Hardwarekosten**

Benötigt die neue Software neue Hardware? Fallen Kosten für die Hardware an?

**Neuentwicklung von Software**

Muss neue Software entwickelt werden? Fallen dadurch höhere Ausgaben an?

**Anpassung von Soft- und Hardware**

Muss die bestehende Software- und Hardwarelandschaft an die neuen Gegebenheiten angepasst werden? Fallen dadurch Kosten an?

**Reduzierte Produktivität während der Umstellung und Einarbeitung**

Wird durch die Umstellung der Software eine umfangreiche Einarbeitung nötig? Sinkt dadurch die Produktivität?

**Anpassung von bestehenden Strukturen, z. B. Vorlagen und Daten**

Müssen bestehende Strukturen (z. B. Vorlagen und Daten) an die neuen Gegebenheiten angepasst werden? Fallen durch diese Anpassung Kosten an?

**Langfristige Kosteneinsparung**

Werden durch die Migration langfristig Kosten eingespart?

**Wartungskosten**

Wie beeinflusst die Migration die laufend anfallenden Wartungskosten?

**Laufende Lizenzkosten**

Fallen laufende Lizenzkosten an?

**Supportkosten**

Wie beeinflusst die Migration den Aufwand für den Support der Nutzer?

**Ausfallkosten**

Sind durch die Migration höhere Ausfallkosten zu erwarten?

**4.1.3. Software**

Neben der Migrationsstrategie und den Kosten ist die Verfügbarkeit von Alternativen und die Qualität dieser Software für den Erfolg einer Migration entscheidend. Ohne adäquate Software kann auch eine noch so gut geplante Migration nicht gelingen, deshalb hier die Kriterien für die Evaluation der Software, die durch die Migration eingeführt werden soll:

**Verfügbarkeit**

Ist für jeden Anwendungsbereich eine adäquate Software verfügbar?

**Langfristige Verfügbarkeit von Patches und Support**

Ist für die Software die langfristige Verfügbarkeit von Patches und Support gewährleistet?



**Anpassbarkeit**

Lässt sich die Software an die eigenen Bedürfnisse anpassen?

**Komplexität**

Ist das Anpassen der Software aufwändig?

**Flexibilität**

Wie flexibel lässt sich die Software anpassen? Lässt sich die Software an sich ändernde Bedürfnisse anpassen?

**Sicherheit**

Gewährleistet die Software die notwendige Sicherheit?

**Zugangskontrolle**

Verfügt die Software über eine Zugangskontrolle, sodass nur berechtigte Personen sie benutzen können?

**Datenschutz**

Lässt sich mit der Software der Datenschutz gewährleisten? Können die Nutzer nur auf die Daten zugreifen, für die sie berechtigt sind?

**Malware-Anfälligkeit**

Wie Anfällig ist die Software für Viren, Trojaner, Spyware etc.?

**Benutzbarkeit**

Wie gut ist die Benutzbarkeit der Software?

**Erwartungskonformität**

Verhält sich die Software so, wie es der Anwender erwartet?

**Dokumentation**

Ist die Software dokumentiert? Gibt es z .B. Handbücher?

**In deutscher Sprache verfügbar**

Ist die Software in deutscher Sprache verfügbar?

**Ausgereiftheit**

Ist die Software hinreichend ausgereift?

**Stabilität**

Ist die Software stabil genug für den organisationskritischen Einsatz?

**Ausfallzeiten**

Wie hoch sind die zu erwartenden Ausfallzeiten?

**Reparaturzeiten**

Wie lange dauern die zu erwartenden Reparaturzeiten?

**Eigene Erfahrungen mit ähnlicher Software**

Kann man auf Erfahrungen mit ähnlicher Software in der eigenen Organisation zurückgreifen?

### **Erfahrungen anderer (öffentlicher) Einrichtungen mit der gleichen oder ähnlicher Software**

Gibt es andere Organisationen, die schon Erfahrungen mit gleicher oder ähnlicher Software gemacht haben?

#### **Wartbarkeit**

Lässt sich die Software gut warten? Wie aufwändig sind diese Wartungen?

#### **Hardwareanforderungen**

Wie groß sind die Hardwareanforderungen der Software?

### **4.1.4. Personal**

Da bei einer Softwaremigration Menschen betroffen sind, ist es wichtig diese von vornherein mit einzubinden und zu informieren, denn nur motivierte Mitarbeiter sind produktive Mitarbeiter. Es folgen die Kriterien, anhand derer evaluiert werden kann, ob das Personal bei der Migration hinreichend berücksichtigt wird:

#### **Motivation**

Wie wirkt sich die Migration auf die Mitarbeitermotivation aus? Werden motivationsfördernde Maßnahmen ergriffen?

#### **Verbesserung oder Verschlechterung der Arbeitsbedingungen**

Verändern sich die Arbeitsbedingungen durch die Migration?

#### **Information**

Werden die Beschäftigten über die Veränderungen informiert?

#### **Einbindung des Personals**

Wird das Personal bei der Migration mit eingebunden? Können die Beschäftigten Einfluss auf die Migration nehmen?

#### **Schulungen**

Werden die Beschäftigten ausreichend für die neue Software geschult?

#### **Support**

Wie hoch ist der Supportbedarf der Mitarbeiter durch die Migration?

### **4.1.5. IT-Abteilung**

Um neue Software erfolgreich einzuführen, bedarf es einer dazu fähigen und motivierten IT-Abteilung, somit hier die Kriterien zur Evaluation, ob die IT-Abteilung im Stande ist die Migration durchzuführen:

#### **Erfahrung mit der neuen oder ähnlicher Software**

Haben die Beschäftigten der IT-Abteilung Erfahrung mit der neuen oder zumindest mit ähnlicher Software?

#### **Genügend Personal vorhanden oder akquirierbar**

Ist der Bedarf an IT-Personal für die Migration gedeckt? Falls nein, lassen sich genügend Fachkräfte anwerben?

**Fähigkeiten**

Besitzt die IT-Abteilung die Fähigkeit eine Migration erfolgreich durchzuführen?

**Schulungen**

Werden die IT-Mitarbeiter für die Migration geschult?

**Motivation**

Sind die in der IT-Abteilung beschäftigten motiviert die Migration durchzuführen?

**4.1.6. Politik**

Eine Migrationen in einer öffentlichen Verwaltung hängt von der Unterstützung der Politik ab. Ohne politische Entscheidung für eine Migration, gibt es keine Migration. Für die Politik ist in der Regel die Migration selbst von untergeordneter Bedeutung und die Auswirkungen der Migration stehen im Vordergrund. So ist z. B. die Herstellerunabhängig und die Mittelstandsförderung von herausragender Bedeutung. Nachfolgend die Evaluationskriterien anhand derer die Unterstützung der Politik für die Migration evaluiert werden kann:

**Herstellerunabhängigkeit**

Ist durch die einzusetzende Software eine Herstellerunabhängig gewährleistet?

**Offene Standards**

Werden offene Dokumentenstandards verwendet, sodass die Daten auch langfristig noch lesbar sind?

**Mittelstandsförderung**

Lassen sich durch den Einsatz der neuen Software gezielt kleine und mittelständische Unternehmen fördern?

**Ansehen in der Bevölkerung**

Welche Auswirkungen hat die Migration auf das Ansehen der öffentlichen Einrichtung in der Bevölkerung?

**Entscheidungswille**

Steht die Politik hinter der Migration der Verwaltung?

**Politische Agenda**

Stehen die einzelnen politischen Parteien langfristig hinter der Migration?

**Mehrheiten**

Ist die Mehrheit, die hinter der Migration steht, verlässlich?

**4.2. Fazit Kriterien für eine Software-Migration**

Anhand des Kriterienkatalogs lässt sich erkennen, dass der Erfolg einer Softwaremigration in einer öffentlichen Verwaltung von vielen Faktoren abhängt. Ohne eine Strategie, ohne Wirtschaftlichkeit, ohne verfügbare und ausgereifte Softwarealternativen, ohne motiviertes Personal und ohne eine fähige IT-Abteilung lässt sich eine Softwaremigration nicht erfolgreich durchführen. Und ohne die Unterstützung der Politik findet keine Migration in einer öffentlichen Verwaltung statt.

## 5. Open-Source in der öffentlichen Verwaltung

Die Stadt München ist nicht die einzige öffentliche Verwaltung, die Client-seitig auf Open-Source-Software umsteigt. Nachfolgend wird eine Auswahl von Projekten (Schwäbisch Hall, Treuchtlingen, Wienux und NIVADIS) vorgestellt, bei denen entweder die Migration bereits erfolgreich abgeschlossen ist oder schon begonnen hat. Die Reihenfolge der Vorstellung der Projekte ergibt sich aus der Vergleichbarkeit mit dem LiMux-Projekt. Die Migration von Schwäbisch Hall ist am ehesten Vergleichbar mit der in München, denn in Schwäbisch Hall wurden die Clients von Windows NT auf Linux migriert. In Treuchtlingen wurde hingegen auf Linux-Thin-Clients migriert, in Wien wird es den Beschäftigten freigestellt, ob sie Linux oder Windows benutzen möchten und bei der Niedersächsischen Polizei wird nur eine Fachanwendung (NIVADIS) plus OpenOffice.org, Mozilla und Ximian-Evolution verwendet.

### 5.1. Schwäbisch Hall

Die Migration der Stadt Schwäbisch Hall gestaltete sich die Migration (vgl. [Bräu05]) wie folgt:

Die Stadt Schwäbisch Hall<sup>38</sup> (ca. 36.000 Einwohner; 225 PC-Arbeitsplätze in der Verwaltung) ist einer der Vorreiter einer Migration auf Open-Source-Software. Im Jahr 2001 kündigte die Firma Microsoft einseitig den Support für das in Schwäbisch Hall als Client-Betriebssystem verwendete Windows NT. Gleichzeitig brachen in jenem Jahr die Gewerbesteuererinnahmen der Stadt ein, sodass evaluiert wurde, ob durch den Einsatz von Open-Source-Software Geld gespart werden könnte.

Die neue Software sollte folgenden Kriterien genügen: Herstellerunabhängigkeit, Sicherheit, Einsparungen und Wettbewerb. Open-Source-Software war in allen diesen Bereichen die günstigere Wahl gegenüber der proprietären Alternative von Microsoft. Beispielsweise war bei der Herstellerunabhängigkeit ausschlaggebend, dass eine Migration auf neuere Microsoft Produkte 250.000 Euro gekostet hätte und dass Microsoft jeder Zeit auch für diese Software den Support wieder aufkündigen könnte und somit erneut eine Zwangsmigration anstünde. Andererseits ist eine ablösende Migration auch mit Risiken verbunden, z. B. lässt sich der Supportbedarf für die neue Software schwer abschätzen. Nichtsdestotrotz konnte die Open-Source-Alternative dadurch punkten, dass die bestehende Hardware weiterverwendet werden konnte, was bei einer Migration auf Windows XP nicht der Fall gewesen wäre, und dass bei Open-Source-Software die regionale Wirtschaft gestärkt werden kann, weil keine Lizenzgebühren ins Ausland fließen.

Ende 2001 wurde beschlossen, alle Clients auf Open-Source-Software zu migrieren. Ende 2004 war die Migration vollzogen. Auf den Clients ist u. a. SuSE Linux Desktop und OpenOffice installiert. Fachanwendungen, die nicht unter Linux selbst oder als Terminalemulation (für die Mainframe-Legacy-Anwendungen) verfügbar sind, werden in Schwäbisch Hall auf Microsoft Windows Terminal Servern betrieben, was zwar zu Lizenzkosten für den Terminal Server führt, aber die Administration vereinfacht.

Neben den bestehenden Mainframes, wurden für Benutzerverwaltung, Datei-Server, Druck-Server, E-Mail und Groupware SuSE Linux OpenExchange Server und SuSE Linux Enterprise Server eingerichtet.

---

38 <http://www.schwaebischhall.de/> (zuletzt besucht am: 24.04.2007)

## 5.2. Treuchtlingen

Noch früher als Schwäbisch Hall migrierte die bayrische Kleinstadt Stadt Treuchtlingen<sup>39</sup> (ca. 14.000 Einwohner; 50 PC-Arbeitsplätze in der Verwaltung) auf Open-Source-Software (vgl. [Treuchtl04] und [Schnei04]). Bereits im Jahr 2000 wurde die Client-Migration von Windows NT auf Linux in Treuchtlingen begonnen. Zu dieser Entscheidung kam es u. a. dadurch, dass aufgrund von kollidierenden DLL manche Beschäftigte mehrere Windows NT-Systeme parallel verwenden mussten, um einen reibungslosen Arbeitsablauf gewährleisten zu können. Des Weiteren war die Motivation da, durch eine Migration auf Open-Source-Software, Geld einzusparen.

In Treuchtlingen werden Thin-Clients von Sun Microsystems verwendet, die sich besonders durch ihre einfache Wartbarkeit auszeichnen, denn die Software läuft auf dem (Linux-)Server, der Thin-Client ist ausschließlich für die Darstellung zuständig. Ein weiterer Vorteil der Thin-Clients von Sun ist, dass die Benutzer eine Smartcard erhalten, mit der sie z. B. ihre Session unterbrechen und an einem beliebigen anderen Client fortsetzen können, weil sämtliche Anwendungen auf dem Server weiterlaufen.

In Treuchtlingen kommt u. a. folgende Software zum Einsatz: Linux auf den Servern und Clients, OpenOffice und eine Groupware, die in PHP implementiert ist und somit als Webservice läuft. Anwendungen, die (noch) nicht direkt für Linux oder als Webservice verfügbar sind, werden von unter VMware Server laufenden Windows bereitgestellt.

Die Migration wurde im Juni 2002 vollendet. Seit der Umstellung konnten in Treuchtlingen ca. 40% der jährlichen EDV-Betriebskosten eingespart werden.

## 5.3. Wienux

Eine grundsätzlich andere Herangehensweise (vgl. [Lutz06], [Lutz04k] und [Lutz04l]), als die anderen vorgestellten Open-Source-Migrationsprojekte, verfolgt die österreichische Hauptstadt Wien<sup>40</sup>, sie setzt bei der Migration auf die Freiwilligkeit der Anwender, d. h. die Beschäftigten können selber entscheiden, ob sie OpenOffice.org und/oder Wienux (ein an die Wiener Bedürfnisse angepasstes Debian Linux) verwenden wollen. Bislang wird auf den circa 18.000 PCs der Stadt hauptsächlich Microsoft Windows 2000 verwendet, für die Wien noch Supportverträge mit Microsoft hat, die erst 2010 auslaufen, sodass hier der Migrationsdruck noch nicht akut ist.

Auf den Servern werden in Wien schon seit 1989 Open-Source-Programme (z. B. FreeBSD oder Linux) verwendet. Um die Möglichkeiten eines Einsatzes von Open-Source-Software auf den Client-Computern zu evaluieren, hat die IT-Abteilung der Stadt Wien, die Magistratsabteilung 14 (Automationsunterstützende Datenverarbeitung, Informations- und Kommunikationstechnologie; kurz MA 14), eine Studie erarbeitet, die als Kurz- [Lutz04k] und als Langversion [Lutz04l] verfügbar ist.

Im Rahmen der Studie wurde insbesondere untersucht, ob die eingesetzten Anwendungen unter Linux benutzbar sind bzw. eine Linux-Alternative verfügbar ist. Für die insgesamt 1100 Anwendungen, die z.T. nur auf wenigen Computern installiert sind, ergab sich folgendes Ergebnis:

---

39 <http://www.treuchtlingen.de/> (zuletzt besucht am: 24.04.2007)

40 <http://www.wien.gv.at/> (zuletzt besucht am: 24.04.2007)

Software-Kategorie	Anzahl	Anteil
Plattformunabhängig	90	8%
Alternative verfügbar	175	16%
Neu zu entwickeln	500	46%
Keine sinnvolle Alternative	100	9%
Unbestimmter Status	235	21%
<b>Summe</b>	<b>1.100</b>	<b>100%</b>

*Tabelle 3: Klassifizierung der in Wien eingesetzten Software.*

*Quelle: vgl. [Lutz06] S. 75*

Programme die nur auf einzelnen Computern installiert sind, wurden nicht untersucht, daher die 235 Anwendungen mit unbestimmtem Status. „Keine sinnvolle Alternative“ ergab sich zumeist bei Graphik-Programmen. Bei den 500 neu zu entwickelnden Programmen handelt es sich um Eigenentwicklungen, welche im Rahmen der regulären Produktlebenszyklen sukzessive durch plattformunabhängige Neuentwicklungen abgelöst werden sollen.

Insgesamt ergibt sich, dass 7.500 der 16.000 PCs kein Microsoft Office bedingen und somit OpenOffice.org ohne großen Aufwand eingesetzt werden kann. Von diesen 7.500 Arbeitsplätzen verwenden 4.800 ausschließlich plattformunabhängige Software oder die verwendete Software kann leicht durch Linux- Alternativen ersetzt werden, sodass diese Computer relativ leicht auf Linux migriert werden können.

Um eine Migration der 7.500 PCs auf OpenOffice.org und der 4.800 PCs auf Linux durchführen zu können, rechnet die Studie für den 5 jährigen Betrachtungszeitraum mit Kosten von 1,1 Millionen Euro, wovon allerdings nur 3.500 Euro ausgabenwirksame Kosten sind.

Letztendlich hat man sich in Wien dazu entschieden bei der Migration auf Freiwilligkeit zu setzen. Das bedeutet in der Praxis, dass auf allen 16.000 PCs OpenOffice.org installiert wird, aber Microsoft Office nur deinstalliert wird, wenn der jeweilige Anwender dies wünscht. Darüber hinaus können die Beschäftigten auf Linux umsteigen, wenn ihre verwendete Software dies zulässt und sie dies wünschen. Durch die Freiwilligkeit der Migration erhofft sich die Stadt Wien nur einen minimal erhöhten Schulungsaufwand, weil erwartet wird, dass in erster Linie computerversierte Anwender umsteigen, die kaum Support benötigen.

Durch diese Art der Migrationen werden die Kosten niedrig und die Wahrscheinlichkeit von Komplikationen minimiert, aber trotzdem werden Erfahrungen mit Open-Source-Software gesammelt und bei der für das Jahr 2008 anvisierten Upgrade von Microsoft Office und Microsoft Windows werden entsprechend weniger Lizenzen benötigt, was zu niedrigeren Ausgaben führt.

## 5.4. NIVADIS

Ein grundsätzlich anderes Szenario als in Schwäbisch Hall, Treuchtlingen und Wien gibt es in Niedersachsen, wo die Polizei mit gut 11.500 PC-Arbeitsplätzen auf Linux migriert ist (vgl. [Nivadis04]). Neben den betriebswirtschaftlichen Gründen, niedrigen Hardwareanforderungen und der Herstellerunabhängigkeit war für die niedersächsische Polizei vor allem die Sicherheit der Software entscheidend. Der Unterschied zu den anderen Projekten liegt darin, dass die Vielfalt der eingesetzten Programme bei der Polizei weitaus geringer ist als bei einer Stadtverwaltung.

Bereits im Jahr 2001 wurde entschieden, die bislang eingesetzten 23 Programme und mehrere Papierprozesse in einem Programm zu konsolidieren. Dieses Programm namens NIVADIS (Niedersächsische Vorgangsbearbeitungs-, Analyse-, Dokumentations- und Informations-System) ist in JAVA implementiert worden, sodass es z. B. sowohl unter Windows als auch Linux lauffähig ist. 2002 wurde beschlossen, dass nicht

nur auf NIVADIS umgestellt werden sollte, sondern, dass sämtliche PCs auf Linux migriert werden sollten. Heute wird bei der niedersächsischen Polizei neben NIVADIS und Linux OpenOffice.org als Office-Suite, Mozilla als Browser und Ximian-Evolution als E-Mail- und Groupware-Anwendung verwendet.

Durch den Einsatz von Open-Source-Software und dem damit verbundenen Wegfall von Lizenzkosten werden innerhalb der ersten 10 Jahre Einsparungen in Höhe von 117 Millionen Euro erwartet.

### **5.5. Fazit Open-Source in der öffentlichen Verwaltung**

Bereits vor dem LiMux-Projekt sind verschiedene öffentliche Verwaltungen erfolgreich auf Open-Source-Software migriert. Allerdings sind diese nicht annähernd so komplex, wie die Verwaltung einer Millionen-Metropole.

Am ehesten lässt sich die Migration der Stadt Schwäbisch Hall mit München vergleichen, weil auch dort aufgrund der Supportkündigung für Windows NT seitens Microsofts die Clients auf Linux umgestellt worden. Jedoch wurden in Schwäbisch Hall nur 225 PCs auf Linux migriert, während es in München 14.000 sind.

Treuchtlingen und Wien zeigen, dass es auch noch andere Arten der Migration auf Open-Source-Software gibt. In Treuchtlingen wurde anstatt auf den vorhandenen Fat-Clients (PCs) Linux zu installieren auf Linux-Thin-Clients migriert und in Wien wurde die Umstellung den Beschäftigten freigestellt, was den Supportbedarf begrenzt, weil zu erwarten ist, dass nur Beschäftigte umsteigen, die sich das zutrauen. Solche Menschen brauchen in der Regel kaum Support. Trotz das nicht alle Umsteigen, werden immerhin Lizenzkosten für die gespart, die Umsteigen.

Nicht alle öffentliche Verwaltungen haben solch vielfältige und komplexe Softwareanforderungen wie München, Schwäbisch Hall, Treuchtlingen, und Wien. Bei der Niedersächsischen Polizei beschränkt sich der Softwarebedarf auf eine Fachanwendung, eine Office-Suite, einen Browser und einen E-Mail- und Groupware-Client. Mit der Wahl von OpenOffice.org, Mozilla und Evolution, die alle sowohl für Windows als auch für Linux verfügbar sind, war hier die Herausforderung die Migration der einzigen Fachanwendung, die durch die Neuentwicklung NIVADIS in JAVA bewältigt wurde.

## 6. Das LiMux-Projekt

In den folgenden Kapiteln wird das LiMux-Projekt beschrieben und analysiert. Hierbei wird besonders auf folgende Quellen zurückgegriffen: die Webseite des Projektes<sup>41</sup>, auf Informationen, die durch ein Gespräch vor Ort mit dem stellvertretenden Projektleiter Manfred Lubig-Konzett (für das vollständige Interview siehe Anhang: „Interview vom 15.11.2006 mit Manfred Lubig-Konzett“, ab S. 59) erlangt wurden und den Artikel „Das Projekt LiMux - Freie Software für die Münchner Verwaltungsclients“ [Hoe06] aus dem Open-Source-Jahrbuch 2006 [OSJb06].

Der Name LiMux steht für „Linux in München“.

### 6.1. Ausgangszustand

Vor dem Beginn des LiMux-Projektes wurden bei der Münchner Stadtverwaltung auf den Clients ausschließlich Microsoft Windows NT 4 als Betriebssystem und Microsoft Office 97 und 2000 als Bürosoftwarepakete eingesetzt (vgl. [Hoe06], S. 59f.). Neben dem Betriebssystem und dem Office-Paket wurden noch diverse Fachanwendung eingesetzt, die eine besondere Herausforderung bei einer Migration darstellen.

Auf den Servern wurden diverse Unix-Betriebssysteme und BS2000<sup>42</sup> von der Firma Siemens eingesetzt. BS2000 ist ein Betriebssystem für Großrechner, auf die von Clients per Terminalemulation zugegriffen werden kann. Daneben wurden Datei- und Druckerdienste auf Basis von Novell Netware<sup>43</sup> angeboten und es kamen Datenbankserver von Oracle zum Einsatz (vgl. ebd., S. 60).

Insgesamt gab es (Stand 2001) bei der Stadtverwaltung von München etwa 14.000 PCs (vgl. ebd., S. 60f.) für ca. 16.000 Benutzer, d. h. teilweise teilen sich mehrere Beschäftigte einen PC, was z. B. auf Teilzeitarbeit zurückzuführen ist, und teilweise benötigen die Beschäftigten keinen PC, z. B. mobiles Personal. Es gibt ca. 170 Client-Server-Fachverfahren, ca. 170 Großrechner-basierte Fachverfahren und ca. 300 verschiedene Softwareprodukte auf den Clients. Diese große Vielfalt von unterschiedlichen Anwendungen lässt sich vor allem darauf zurückführen, dass die Münchner Stadtverwaltung dezentral organisiert ist. Es gibt insgesamt 17 IT-Sachgebiete in 12 Referaten (Behörden der Stadt München) und 5 Eigenbetrieben, die für Planung und Betrieb ihrer Fachanwendung, Datei- und Druckerdienste und den Anwendersupport verantwortlich sind. Dies hat zur Folge, dass in den unterschiedlichen Referaten z. T. unterschiedliche Lösungen für die gleichen Problemstellungen zum Einsatz kommen, sodass sich durch Konsolidierung der Lösungen ein Einsparpotenzial ergibt. Allerdings haben sich die Beschäftigten an ihre individuellen Lösungen gewöhnt, sodass bei einer Konsolidierung kurzfristig mehr Supportbedarf anfällt.

Im Jahr 2001 evaluierte die Stadt München den Einsatz der alternativen Bürosoftware StarOffice (vgl. ebd., S. 61). Zu diesem Zeitpunkt wurde jedoch festgestellt, dass StarOffice zu Microsoft Office 97 keine hinreichende Kompatibilität bot.

Angeregt durch diese Evaluation beschloss der Münchner Stadtrat (das Parlament der Stadt München) am 14. November 2001 mehrheitlich die Untersuchung der technischen und wirtschaftlichen Auswirkungen eines stadtweiten Einsatzes von alternativen Produkten zu Microsoft Office und Microsoft Windows für die Client-Computer der Stadt (vgl. ebd., S. 61f.). Das Einbeziehen des Betriebssystems bei der Evaluation wurde ins-

41 <http://www.muenchen.de/Rathaus/dir/limux/89256/index.html> (zuletzt besucht am: 24.04.2007)

42 <http://www.fujitsu-siemens.de/products/bs2000/index.html> (zuletzt besucht am: 24.04.2007)

43 <http://www.novell.com/de-de/products/netware/> (zuletzt besucht am: 24.04.2007)



besondere deshalb beschlossen, weil Microsoft das Ende des Supports für Windows NT angekündigt hatte. Da Microsoft Windows proprietäre Software ist, kann weder ein anderes Unternehmen noch die eigene IT-Abteilung mit der Pflege (z. B. Entwicklung von Patches) von Windows beauftragt werden, sodass nichts anderes übrig bleibt, als entweder auf ein neueres Windows oder auf ein alternatives Betriebssystem (z. B. Linux, das in Kapitel 2.2.1 erläutert wird) umzusteigen.

Weil das zuständige Amt für Informations- und Datenverarbeitung sich personell nicht in der Lage sah, eine solche umfassende stadtweite Untersuchung durchzuführen, wurde dem Stadtrat vorgeschlagen die Evaluation der Client-Software als Studie auszu-schreiben, was am 17. April 2002 beschlossen wurde (vgl. ebd., S. 62).

Die Ausschreibung für die Studie wurde von der Firma UNILOG Integrata Unternehmensberatung GmbH<sup>44</sup> gewonnen (vgl. ebd.).

## **6.2. Auswahlprozess**

Die UNILOG Studie [CliStu03] wurde als Entscheidungsvorlage für den Münchner Stadtrat erarbeitet. In der Studie wurden Alternativen für die zukünftige Client-Strategie vorgestellt, diese nach technischer Machbarkeit abgewogen, deren wirtschaftliche Auswirkungen betrachtet und die qualitativ-strategischen Konsequenzen bewertet. Als Alternativen wurden eine rein proprietäre Lösung und eine reine Open-Source-Lösung, sowie drei Mischlösungen, betrachtet.

---

<sup>44</sup> <http://www.unilog-integrata.de/> (zuletzt besucht am: 24.04.2007)

### 6.2.1. Qualitative Betrachtungen

Die Alternativen waren folgende:

Alternative	Client-Betriebssystem	Büro- und Kommunikationssoftware	Fachapplikationen
Microsoft Windows XP + Microsoft Office XP (XP/XP)	Microsoft Windows XP	Microsoft Office XP + Microsoft Outlook + Microsoft Internet Explorer	native Windows Lösungen
Microsoft Windows XP + OpenOffice.org (XP/OSS)	Microsoft Windows XP	OpenOffice.org + Mozilla Suite	native Windows Lösungen
Linux + OpenOffice.org (LX/OSS)	Linux	OpenOffice.org + Mozilla Suite	native Linux Lösungen oder Webservices
Linux + OpenOffice.org + PC-Emulation (LX/OSS/VM)	Linux	OpenOffice.org + Mozilla Suite	unter Windows NT, welches in einer Virtual Machine (VMware) unter Linux betrieben wird
Linux + OpenOffice.org + Microsoft Windows Terminal Server (LX/OSS/TS)	Linux	OpenOffice.org + Mozilla Suite	unter Microsoft Windows Terminal Server

Tabelle 4: Migrationsalternativen der Stadt München. Quelle: vgl. [Hoe06], S. 62f.

Einige Fachapplikationen bleiben bei allen fünf Alternativen der Client-Migration unberührt, weil diese hauptsächlich auf dem Server ablaufen. Hierbei handelt es sich um webbasierte und um BS2000-Anwendungen, die 2002 insgesamt etwa 8% aller Fachanwendung ausgemacht haben. Darüber hinaus kann bei Fachverfahren, die sich in der Planung befinden oder demnächst neu ausgeschrieben werden, die neue Client-Infrastruktur berücksichtigt werden, sodass hier keine gesonderten Migrationskosten anfallen. Diese machten 2002 insgesamt ca. 9% aller Fachanwendungen aus. Tatsächlich sind somit nur 83% aller Fachanwendungen bei einer Migration zu betrachten (vgl. [CliStu03], S. 7f.).

Die Münchner Stadtverwaltung zeichnete sich durch eine dezentrale Organisation aus (vgl. ebd., S. 10). Aufgrund dieser dezentralen Organisation gab es keinen zentralen Help-Desk und keine zentrale Softwareverteilung. Es gab noch nicht einmal eine Standardisierung der PC-Arbeitsplätze. Einzig das Betriebssystem wurde vorgeschrieben, und es wurde verlangt, dass alle Mitarbeiter Zugang zu E-Mail und Intranet haben können. Der Zugriff auf das Internet war nur möglich über einen durch eine Firewall geschützten Zugang.

Bei einer Migration von Windows auf Linux gibt es zwei Möglichkeiten, wie diese durchgeführt werden kann. Entweder sie wird hart oder weich durchgeführt. Bei einer

harten Migrationen muss eine Ersatzbeschaffung für Teile der Hardware vor Ende ihres technischen Lebenszyklus durchgeführt werden, und es muss (noch) nicht vorhandene Software für Linux selbst entwickelt werden, was erhebliche Kosten verursacht. Bei einer weichen Migration werden die Lebenszyklen der Hardware berücksichtigt, d. h. wird eine bestimmte Hardware nicht von Linux unterstützt, wird diese erst nach Ende ihres Lebenszyklus durch Linux-kompatible Hardware ersetzt. Vergleichbares gilt für die Fachanwendungen, die nach Möglichkeit erst bei einem ohnehin vorgesehenen Versionswechsel als Linux-Version ausgeschrieben werden. Das bedeutet, dass eine weiche Migration zwar deutlich länger dauert als eine harte, aber dafür deutlich kostengünstiger durchzuführen ist (vgl. ebd., S. 11).

Um die weiche Migration zu vereinfachen, gibt es mehrere Möglichkeiten, wie mit noch nicht als Linux-Version verfügbarer Software umzugehen ist. Diese kann z. B. in einer Virtual Machine weiterhin unter dem alten Windows betrieben werden oder durch einen Microsoft Windows Terminal Server zu Verfügung gestellt werden (vgl. ebd., S. 11).

Wird eine Migration von Windows NT zu Windows XP, respektive von Microsoft Office 97 zu XP, einer Migration von Windows NT zu Linux, respektive Microsoft Office zu OpenOffice.org, gegenübergestellt, fällt auf, dass nicht nur bei der Migration auf Linux/OpenOffice.org z. T. erhebliche Anpassungen nötig sind, sondern auch bei einer Migration von einer Version zu einer höheren von Microsoft Produkten. So muss bei einem Wechsel von Windows zu Linux 100% der vorhandenen Software ausgetauscht oder erheblich angepasst werden, während bei einem Umstieg von Windows NT zu XP immerhin noch 20% auszutauschen sind. Bei einer Migration von Microsoft Office 97 auf OpenOffice.org 2.0 fallen 70% der ursprünglichen Kosten der Office-Konfiguration an, z. B. für die Anpassung von Vorlagen und Makros. Bei einem Wechsel von Microsoft Office 97 zu 2003 fallen jedoch auch noch einmal 20% der ursprünglichen Kosten an, denn auch Microsoft Office 2003 ist nicht 100%-kompatibel zu Microsoft Office 97. Bei den Hardware-Anforderungen werden für alle Lösungen ein 500MHz-Prozessor und mindestens 256MB RAM festgestellt. Dem zufolge müssten bei jeder Migrationsalternative ca. 7000 PCs ersetzt und ca. 5000 PCs ein RAM-Upgrade erhalten, d. h. nur ca. 2000 PCs erfüllten 2002 diese Anforderungen (vgl. ebd., S. 12-15).

Für alle Handlungsalternativen wurde festgestellt, dass sie technisch umsetzbar sind, aber sich wirtschaftlich und strategisch unterschiedlich auswirken (vgl. [Hoe06], S. 63).

### **6.2.2. Wirtschaftlichkeitsbetrachtung**

Alle Migrationsalternativen wurden im Rahmen der Client-Studie gemäß den WiBe21-Richtlinien auf ihre Wirtschaftlichkeit untersucht. WiBe21 sind Empfehlungen der Bundesverwaltung zur Durchführung von Wirtschaftlichkeitsbetrachtung (vgl. [CliStu03], S. 16).

Die Wirtschaftlichkeitsbetrachtung der Alternativen wurde nur für die unmittelbaren Kosten der Migration durchgeführt. Beispielsweise wurde davon ausgegangen, dass die Netzinfrastruktur (LAN und WAN-Anbindung) hinreichend schnell und zuverlässig ist (vgl. ebd.).

Bei der Wirtschaftlichkeitsbetrachtung wurden sowohl haushaltswirksame (z. B. Ausgaben für externe Dienstleistungen) als auch nicht haushaltswirksame (Eigenleistungen) Kosten betrachtet.

Es wurden folgende Kostenarten klassifiziert:

- Feinkonzept und Projektadministration
- Hardwarekosten inkl. Installation

- Schulungskosten inkl. Teilnehmerkosten
- Softwarekosten (z. B. Lizenzen)
- Migrationskosten der Fachverfahren
- Anpassung von Makros und Formularen

Der Betrachtungszeitraum der Untersuchung wurde auf 5 Jahre festgesetzt (vgl. ebd., S. 17).

Für die Windows-basierten Alternativen (Microsoft Windows XP + Microsoft Office XP (XP/XP) und Microsoft Windows XP + OpenOffice.org (XP/OSS)) wurden folgende Zahlen ermittelt:

Alternativen	XP/XP		XP/OSS	
	hw.	n. hw.	hw.	n. hw.
<b>Einführungskosten</b>				
Personal (Projektadministration & Feinkonzept)	878.910	817.500	1.019.525	981.870
Hardware	548.992	34.416	548.992	34.416
Lizenzen	2.260.626	47.395	860.208	47.395
Migration Software	2.525.116	833.161	5.250.680	1.732.459
Umstellung Formulare & Makros	235.965	157.310	825.879	550.586
Schulung	3.585.610	10.408.219	4.783.564	13.839.595
Einarbeitung	0	3.091.627	0	4.292.608
<b>Betriebskosten</b>				
Betriebskosten	6.064.123	2.693.185	2.284.667	2.693.185
<b>Summe</b>	<b>16.099.342</b>	<b>18.082.813</b>	<b>15.573.515</b>	<b>24.172.114</b>
<b>Gesamtsumme</b>	<b>34.182.155 €</b>		<b>39.745.629 €</b>	
<b>Kapitalwert (Berechnung n. WiBe 21)</b>	<b>31.303.370 €</b>		<b>37.045.780 €</b>	

Tabelle 5: Kostenaufschlüsselung Windows XP-Client-Alternativen.

Quelle: [CliStu03], S. 18

*hw.* = haushaltswirksam (externe Aufwendungen); *n. hw.* = nicht haushaltswirksam (interne Aufwendungen)

Für die Linux-basierten Alternativen (Linux + OpenOffice.org (Linux/OSS), Linux + OpenOffice.org + VMware (Linux/OSS/VM) und Linux + OpenOffice.org + Microsoft Windows Terminal Server (Linux/OSS/TS)) ergaben sich folgende Zahlen:

Alternativen	Linux/OSS		Linux/OSS/VM		Linux/OSS/TS	
	hw.	n. hw.	hw.	n. hw.	hw.	n. hw.
<b>Einführungskosten</b>						
Personal (Projektadministration & Feinkonzept)	1.305.158	1.184.220	1.019.525	981.870	1.305.158	1.184.220
Hardware	875.498	83.392	875.498	83.392	11.287.111	603.973
Lizenzen	1.560.259	233.889	2.126.676	23.698	3.769.092	23.698
Migration Software	10.060.283	3.319.386	1.262.558	416.580	1.262.558	416.580
Umstellung Formulare & Makros	825.879	550.586	825.879	550.586	825.879	550.586
Schulung	5.459.585	15.665.731	5.459.585	15.665.731	5.459.585	15.665.731
Einarbeitung	0	4.931.756	0	4.931.756	0	4.931.756
<b>Betriebskosten</b>						
Betriebskosten	-690.996	404.400	1.276.697	444.840	1.104.903	1.617.600
<b>Summe</b>	<b>19.395.666</b>	<b>26.373.360</b>	<b>12.846.418</b>	<b>23.098.453</b>	<b>25.014.286</b>	<b>24.994.144</b>
<b>Gesamtsumme</b>	<b>45.769.026 €</b>		<b>35.944.871 €</b>		<b>50.008.430 €</b>	
<b>Kapitalwert (Ber. n. WiBe 21)</b>	<b>43.167.498 €</b>		<b>33.762.122 €</b>		<b>46.560.401 €</b>	

Tabelle 6: Kostenaufschlüsselung Linux-Client-Alternativen. Quelle: [CliStu03], S. 18

*hw.* = haushaltswirksam (externe Aufwendungen); *n. hw.* = nicht haushaltswirksam (interne Aufwendungen)

Bei den Betriebskosten für Linux/OSS ergeben sich Einsparungen aufgrund des Wegfalls von OEM-Lizenzen, die bereits im städtischen Haushaltsplan eingeplant waren.

Über den 5-jährigen Betrachtungszeitraum gesehen ist somit die rein proprietäre Lösung (XP/XP) die günstigste. Diese Lösung hat die geringsten Hardware- und Migrationskosten sowie den geringsten Schulungs- und Einarbeitungsaufwand. Wesentlicher Kostentreiber bei dieser Lösung sind die Lizenzkosten von insgesamt 7,65 Millionen Euro, die sich, aufgrund der Lizenzbedingungen der Firma Microsoft, sowohl in den Einführungskosten als auch in den Betriebskosten niederschlagen.

Werden nur die haushaltswirksamen Ausgaben betrachtet, so ist die Linux/OSS/VM die preisgünstigste Alternative. Diese verursacht etwa 3 Millionen Euro weniger haushaltswirksame Ausgaben als die zweitgünstigste Alternative (XP/XP). Von den Gesamtkosten in den ersten 5 Jahren her ist die Linux/OSS/VM-Alternative nur ca. 2 Millionen Euro, also etwa 5%, teurer als die günstigste Lösung (vgl. [CliStu03], S. 19).

Wenn man die Betriebskosten betrachtet, so wird deutlich, dass alle Open-Source-Alternativen hier deutlich billiger sind als die Closed-Source-Varianten. Somit lässt sich erwarten, dass langfristig gesehen die Open-Source-Lösungen günstiger sein werden. Die nicht haushaltswirksamen Betriebskosten betragen bei der Linux/OSS- und der Linux/OSS/VM-Lösung nur etwa 400.000€ während diese bei der XP/XP- und der XP/OSS-Lösung fast 3.000.000€ betragen. Besonders negativ fallen bei der Linux/OSS-Lösung die Kosten für die Softwaremigration auf, während bei der Linux/OSS/TS-Variante die Hardwarekosten und bei der XP/XP-Alternative die Lizenzkosten besonders hoch sind.

Alles in Allem sind die XP/XP- und die Linux/OSS/VM-Lösungen die günstigsten Alternativen mit der ausgeglichensten Kostenverteilung.

### 6.2.3. Strategische Auswirkungen

Da sich die Auswirkungen einer Migration nicht ausschließlich auf die monetäre Situation beschränken, wurde im Rahmen der Studie auch die strategischen Auswirkungen betrachtet. Um die strategischen Auswirkungen zu beurteilen, wurde ein Kriterienkatalog zusammen mit der Stadt München erarbeitet. Dieser ist in sechs unterschiedlich gewichtete Kategorien (Kriteriengruppen) unterteilt, die wiederum in gewichtete Unterkategorien (Kriterien) gegliedert sind. Die sechs Kategorien sind:

- Aufwand<sup>45</sup> für die Einhaltung von Gesetzen und Verwaltungsvorschriften
- Auswirkungen auf die IT-Sicherheit
- Auswirkungen auf die Mitarbeiter
- Auswirkungen auf die IT-Organisation
- Auswirkungen auf externe Adressaten
- Erfüllung weiterer strategischer Punkte

Die Gewichtung der Kategorien und der Unterkategorien ist so normiert, dass sie jeweils 100 ergeben, sodass insgesamt 10.000 Punkte zu vergeben sind. 0 Punkte für eine Unterkategorie bedeutet, dass keine Verbesserung gegenüber der bisherigen Situation möglich ist und die höchstmögliche Punktzahl der Unterkategorie bedeutet, dass dieses Kriterium zu 100% erfüllt wird (vgl. [CliStu03], S. 20-23).

---

<sup>45</sup> Hierbei handelt es sich ausschließlich um den nicht-monetären Aufwand, also Arbeitszeit der eigenen Beschäftigten.

Die folgende Tabelle zeigt das Ergebnis der Gewichtung. Anhand dieser lässt sich erkennen, welche Kriterien besonders wichtig und welche weniger wichtig waren.

Nr.	Kriterium	Gewichtung		Max. Punkte	
		K-Gruppe	Kriterien	K-Gruppe	Kriterien
<b>1.</b>	<b>Aufwand für die Einhaltung von Gesetzen und Verwaltungsvorschriften</b>	<b>14</b>		1.400	
1.1	Aufwand für die Einhaltung gesetzlicher Vorgaben		30		420
1.2	Aufwand für die Erfüllung der Anforderungen an Datenschutz und Datensicherheit		40		560
1.3	Aufwand für die Erfüllung interner Regeln und Vorgaben		20		280
1.4	Aufwand für die Erfüllung von Auflagen und Empfehlungen		10		140
<b>2.</b>	<b>Auswirkungen auf die IT-Sicherheit</b>	<b>29</b>		2.900	
2.1	Bedeutung für die Realisierung weiterführender Sicherheitslösungen		20		580
2.2	Auswirkungen auf das Bedrohungspotenzial		40		1.160
2.3	Auswirkungen auf die Stabilität des Systems		30		870
2.4	Beurteilung des Pilot-Projekt-Charakters		10		290
<b>3.</b>	<b>Auswirkungen auf die Mitarbeiter</b>	<b>19</b>		1.900	
3.1	Auswirkungen auf die Attraktivität der Arbeitsbedingungen		67		1.273
3.2	Auswirkungen auf die Qualitätssicherung / Qualitätserweiterung		33		627
<b>4.</b>	<b>Auswirkungen auf die IT-Organisation</b>	<b>9</b>		900	
4.1	Auswirkungen auf die Komplexität der Systemumgebung		17		153
4.2	Auswirkungen auf das IT-Betriebsmanagement		50		450
4.3	Auswirkungen auf das IT-Personalmanagement		33		297
<b>5.</b>	<b>Auswirkungen auf externe Adressaten</b>	<b>5</b>		500	
5.1	Auswirkungen auf die Kommunikationspartner		67		335
5.2	Auswirkungen auf das Image		33		165
<b>6.</b>	<b>Erfüllung weiterer strategischer Punkte</b>	<b>24</b>		2.400	
6.1	Einhaltung offener Standards		22		528
6.2	Beurteilung der Herstellerunabhängigkeit		14		336
6.3	Beurteilung der Beschaffungsabhängigkeit		7		168
6.4	Beurteilung der Flexibilität des IT-Einsatzes		14		336
6.5	Beurteilung der Beschaffungskontinuität		4		96
6.6	Einhaltung des Grundsatzes „einheitliches Betriebssystem“ für alle Clients		25		600
6.7	Beurteilung des Schutzes getätigter Investitionen		14		336
		<b>100</b>		<b>10.000</b>	<b>10.000</b>

Tabelle 7: Strategische Kriterien. Quelle: vgl. [CliStu03], S. 21f.

Bei der Festlegung des Katalogs der strategischen Kriterien, wurden die „Auswirkungen auf die IT-Sicherheit“ und die „Erfüllung weiterer strategischer Punkte“ als besonders wichtig hervorgehoben. Von den Kriterien wurden insbesondere folgende hoch gewichtet: die „Auswirkung der Attraktivität der Arbeitsbedingungen“ mit maximal 1273

Punkten (19\*67), die „Auswirkungen auf das Bedrohungspotenzial“ mit maximal 1160, und die „Auswirkungen auf die Stabilität des Systems“ mit maximal 870.

Als weniger wichtig wurde insbesondere die „Beurteilung der Beschaffungskontinuität“ mit nur 96 Punkten bewertet.

Nr.	Kriterium	Max. Punkte	XP/XP	XP/OSS	LX/OSS	LX/OSS VM	LX/OSS TS
1.	Aufwand für die Einhaltung von Gesetzen und Verwaltungsvorschriften	<b>1.400</b>	476	462	756	770	798
2.	Auswirkungen auf die IT-Sicherheit	<b>2.900</b>	2.407	2.465	2.697	2.378	2.291
3.	Auswirkungen auf die Mitarbeiter	<b>1.900</b>	823	505	125	380	443
4.	Auswirkungen auf die IT-Organisation	<b>900</b>	181	166	601	465	315
5.	Auswirkungen auf externe Adressaten	<b>500</b>	134	66	66	83	83
6.	Erfüllung weiterer strategischer Punkte	<b>2.400</b>	1.272	1.409	1.973	1.884	1.850
	<b>Summe</b>	<b>10.000</b>	<b>5.293</b>	<b>5.073</b>	<b>6.218</b>	<b>5.960</b>	<b>5.780</b>
	<b>Platz</b>		<b>4</b>	<b>5</b>	<b>1</b>	<b>2</b>	<b>3</b>

Tabelle 8: Ergebnis der qualitativ-strategischen Betrachtung. Quelle: [CliStu03], S. 23

Das Ergebnis der Untersuchung bezüglich der qualitativ-strategischen Kriterien zeigt, dass alle Linux-Varianten strategisch günstiger sind als die Windows-Alternativen, wobei die LX/OSS-Lösung die meisten Punkte erhält. Diese Lösung kann insbesondere bei der IT-Sicherheit, der Erfüllung weiterer strategischer Punkte (u. a. Einhaltung offener Standards und Herstellerunabhängigkeit) und den Auswirkungen auf die IT-Organisation überzeugen, während die XP/XP-Lösung bei den Auswirkungen auf die Mitarbeiter und auf externe Adressaten und die LX/OSS/TS bei der Einhaltung von Gesetzen und Vorschriften am besten abschneiden.

Alles in allem liegt die LX/OSS-Lösung mit 6.218 von 10.000 möglichen Punkten knapp vor der LX/OSS/VM-Alternative mit 5.960 Punkten. Es folgt die LX/OSS/TS-Variante mit 5.780 Punkten. Weit abgeschlagen folgen schließlich die XP/XP- mit 5.293 und die XP/OSS-Lösung mit 5.073 Punkten.

Hieraus lässt sich erkennen, dass die Linux-Alternativen langfristig, also strategisch, den Windows-Lösungen überlegen sind.

## 6.2.4. Empfehlung

Nach der Analyse der Alternativen stellte sich heraus, dass die verschiedenen Lösungen unterschiedliche Vor- und Nachteile haben. Die XP/XP-Lösung ist in dem Betrachtungszeitraum, den Unilog Integrata in Absprache mit der Stadt München auf 5 Jahre festgesetzt hat, die preisgünstigste Alternative. Dies liegt vor allem an ihren niedrigen Schulungskosten, die je nach Handlungsalternative 50% bis 70% der Gesamtkosten ausmachen. Werden jedoch nur die haushaltswirksamen Ausgaben betrachtet, so ist die LX/OSS/VM-Alternative die günstigste Lösung, weil bei ihr die sehr geringen Lizenzkosten die haushaltswirksamen Ausgaben für die Schulungen mehr als ausgleichen (vgl. [CliStu03], S. 25ff.).

Rein betriebswirtschaftlich gesehen ist die XP/XP-Lösung bei einem 5-jährigen Betrachtungszeitraum die günstigste Alternative. Bei einer längerfristigen Betrachtung



wäre allerdings zu erwarten, dass die Linux-Alternativen günstiger werden, weil für sie (fast) keine laufenden Lizenzkosten anfallen und der Supportbedarf längerfristig nachlässt.

Wird der Kapitalwert, den man durch Abzinsen der zukünftigen Investitionen erhält, dem strategischen Nutzwert gegenübergestellt, so ergibt sich folgendes:

Alternative	XP/XP	XP/OSS	LX/OSS	LX/OSS/VM	LX/OSS/TS
Kapitalwert in €	31.303.370	37.045.780	43.167.498	33.762.122	46.560.401
Nutzwert	5.293	5.073	6.218	5.960	5.780
<b>Kapitalwert pro Nutzwertpunkt in €</b>	<b>5.914</b>	<b>7.303</b>	<b>6.942</b>	<b>5.665</b>	<b>8.055</b>

Tabelle 9: Kapitalwert pro Nutzwert. Quelle: [Hoe06], S. 64

Beim Kapitalwert pro Nutzwert ist die LX/OSS/VM-Lösung mit 5.665 € pro Nutzwertpunkt die günstigste Lösung, gefolgt von der XP/XP-Alternative mit 5.914 € pro Nutzwertpunkt. Weit abgeschlagen folgen die LX/OSS- mit 6.942, XP/OSS- mit 7303 und LX/OSS/TS-Lösung mit 8.055 € pro Nutzwertpunkt.

Die LX/OSS/VM-Lösung kann hier aufgrund ihres recht hohen Nutzwertes, der sich insbesondere durch die Herstellerunabhängigkeit und die Einhaltung von offenen Standards ergibt, ihren gegenüber der XP/XP-Lösung etwas höheren Kapitalwert mehr als wettmachen. Insgesamt ist die LX/OSS/VM-Lösung sowohl beim Kapitalwert (hinter XP/XP) als auch beim Nutzwert (hinter LX/OSS), die zweitbeste Lösung, was sie insgesamt zur ausgewogensten Lösung macht.

Die Migrationsstudie kam folglich zu dieser Empfehlung:

*„Obschon sich die Variante XP/XP in der Prüfung der technischen Machbarkeit und in der monetären Wirtschaftlichkeitsbetrachtung als die sinnvollste Lösung für die LHM<sup>46</sup> erwiesen hat, führt die Gesamtbetrachtung – unter Einbeziehung der Ergebnisse der qualitativ-strategischen Betrachtung – zu der Client-Konfiguration LX/OSS/VM.“* Quelle: [CliStu03], S. 31

Im Detail lautet die Empfehlung (vgl. ebd.):

- Migration des Client-Betriebssystem von Windows NT auf Linux
- Migration der Office-Software von Microsoft Office 97 auf OpenOffice.org
- Die Migration als „weiche“ Migration durchführen, d. h. es soll bei der Migration auf eine Windows-Emulation z. B. VMware, zurückgegriffen werden

Das Vorhaben der Stadt München, möglicherweise auf Linux zu migrieren, blieb lange Zeit von der Öffentlichkeit unbeachtet. Dies änderte sich jedoch schlagartig dadurch, dass der Vorstandsvorsitzende der Firma Microsoft Steve Ballmer seinen Ski-Urlaub Anfang 2003 unterbrach, um dem Oberbürgermeister der Landeshauptstadt München Christian Ude (SPD) zu treffen. Bei diesem Treffen bot Ballmer dem Oberbürgermeister an, für den Fall einer Migration auf Windows XP, kostenlose personelle Unterstützung bereitzustellen. Dieses Angebot veranlasste die Firmen IBM und Novell dazu, für den Fall einer Migration auf Linux, ebenfalls kostenlose Unterstützung anzubieten. Diese neuen Gesichtspunkte wurden von der Firma Unilog kurzfristig in die Clientstudie mit eingearbeitet, was aber an den Handlungsempfehlungen nicht viel änderte. Die XP/XP-Lösung blieb die kostengünstigste Alternative knapp vor der LX/OSS/VM-Lösung und die LX/OSS/VM-Lösung blieb die Lösung mit dem besten Kapitalwert pro Nutzwertpunkt (vgl. [Hoe06], S. 63f. und [CliStu03], S. 33-36).

46 LHM = Landeshauptstadt München

### 6.2.5. Entscheidung des Stadtrats

Am 28. Mai 2003 beschloss der Stadtrat der Landeshauptstadt München mit großer Mehrheit, zukünftig freie Software auf den Clients einzusetzen und Client-Server-Anwendungen nur noch als Webanwendungen zu entwickeln oder auszuschreiben (vgl. [Hoe06], S. 64f.). Vor allem letzteres soll zukünftige Migrationen deutlich günstiger machen. Von dieser Entscheidung erwartet der Stadtrat eine größere Herstellerunabhängigkeit, mehr Wettbewerb am Softwaremarkt und langfristig eine Entlastung des städtischen Haushalts.

Darüber hinaus wurde die Verwaltung beauftragt, ein Feinkonzept für die Migration zu erarbeiten, das detailliert die technischen, wirtschaftlichen und zeitlichen Eckpunkte der Migration umfasst und dieses nach einem Jahr dem Stadtrat vorzulegen (vgl. ebd., S. 65).

Das Feinkonzept der Stadtverwaltung wurde zusammen mit den Firmen IBM und Novell erstellt und am 16. Juni 2004 vom Stadtrat beschlossen. Der Beschluss definiert folgende Projektziele (vgl. ebd., S. 65ff.):

- Migration der Clients ohne Gefährdung des Dienstbetriebes
- Aufbau von Hersteller-, Betriebssystem- und Office-unabhängiger Lösungen
- Migration der Fachverfahren
- Konsolidierung und sinnvoller Ersatz der Client-Standardanwendungen
- Konsolidierung und Migration von Makros, Vorlagen und Formularen
- Einführung eines Systemmanagements für den LiMux-Basis-Client

und folgende Rahmenbedingungen:

- Umstellung durch überwiegend eigenes Personal
- weiche Migration durch Ausnutzung der Produktlebenszyklen mit dezentraler Zuständigkeit
- Umstellung der unproblematischen Bereiche zuerst
- Eigenverantwortung jedes Referates für sein Migrationskonzept und dessen Umsetzung innerhalb eines vorgegebenen Migrationspfades durch die LiMux-Projektleitung

Von besonderer Wichtigkeit ist bei der Migration, dass diese intensiv vorbereitet wird, damit ein reibungsloser Ablauf des Dienstbetriebs jeder Zeit gewährleistet werden kann. Zu der Vorbereitung gehört insbesondere die sorgfältige Auswahl, das Anpassen und das Testen der neuen Software, die Konsolidierung von sich überschneidenden Funktionen und die Information und Einbindung der Beschäftigten (der Nutzer).

### 6.3. Implementierungsprozess

Da es über den Implementierungsprozess noch keine wissenschaftlich zitierfähigen Quellen gibt, wird als Grundlage für dieses Kapitel auf die Website des LiMux-Projektes<sup>47</sup> und auf Informationen, die der Autor durch einen Besuch vor Ort gewonnen hat, zurückgegriffen.

Der Implementierungsprozess begann mit dem Feinkonzept und soll seinen Abschluss in der Umstellung von mindestens 80% der Client-Computern von Windows auf Linux

---

47 <http://www.muenchen.de/Rathaus/dir/limux/89256/index.html> (zuletzt besucht am: 24.04.2007)

finden. Bei einigen Clients lohnt sich die Umstellung auf Linux nicht oder sie ist lizenzrechtlich nicht möglich. Die stadteigene Druckerstraße wird von einer Windows-spezifischen Druckersteuerungs-Software betrieben. Hier wäre es wirtschaftlich unsinnig, diese einzelne Software auf Linux portieren zu lassen. Darüber hinaus werden einige PCs weiterhin mit Windows betrieben, weil deren Nutzer das Programm AutoCAD benötigen, für das es zur Zeit keine adäquate Linux-Alternative gibt und bei dem die Lizenz verbietet, es in einer Virtual Machine oder auf einem Terminal-Server zu betreiben. Aber auch, wenn diese Computer weiterhin unter Windows betrieben werden, so lassen sich hier immerhin die Lizenzgebühren für Microsoft Office einsparen, denn OpenOffice.org läuft auch unter Windows.

Bevor die Migration begonnen wurde, wurde diese sorgfältig vorbereitet. Zentrale Punkte der Vorbereitung waren und sind die Entwicklung eines Basis-Clients und eines E-Learning-Systems, die Konsolidierung von Makros und Vorlagen und die Portierung der Fachanwendung (ggf. durch den Einsatz von Wine<sup>48</sup>).

Die gesamte Migration war Ende 2004 in Gefahr zu scheitern, weil die EU erwogen hatte, eine Richtlinie zu verabschieden, welche die Patentierung von Software erlauben sollte. Aus diesem Grund beauftragte die Stadt München die Kanzlei Frohwitter damit, diesbezüglich ein Gutachten zu erstellen. Das „Kurzgutachten Rechtliche Bedingungen und Risiken der Landeshauptstadt München für den Einsatz von Open-Source Software“ [Fro04] kommt zu dem Schluss, dass das Risiko der Stadt, aufgrund von Softwarepatenten verklagt zu werden, bei Open-Source-Software nicht höher ist als bei proprietärer Software und dass dieses Risiko insgesamt gering ist (vgl. [Fro04], S. 38 und S. 40).

Die Softwarepatent-Richtlinie wurde mittlerweile vom Europäischen Parlament abgelehnt, somit besteht in der EU keine Möglichkeit, Softwarepatente zu verletzen.

### 6.3.1. Akzeptanzförderung

Der allerwichtigste Punkt bei einer Software-Migration ist die Akzeptanz durch die Anwender. Wenn ein Anwender sich quer stellt, kann der beste Support ihm nicht helfen. Um dies zu vermeiden, wurden von der Leitung des LiMux-Projekts von Anfang an akzeptanzfördernde Maßnahmen ergriffen. Es wird laufend über den Fortschritt und die Schwierigkeiten berichtet. Zu diesem Zweck werden ein E-Mail-Newsletter, Flyer, das Intranet, Pressemitteilungen und Informationsveranstaltungen verwendet. Während der Newsletter, die Flyer und das Intranet nur eine ungefähre Vorstellung des LiMux-Projektes erzeugen können, können die Beschäftigten der Stadt München auf den Informationsveranstaltungen den LiMux-Basis-Client selbst ausprobieren.

Des Weiteren ist der Ansatz der Migration, nicht nur langfristig Kosten einzusparen, sondern alles besser zu machen, d. h. der Anwender soll in Zukunft einfacher und effizienter mit seinen Computer-Programmen umgehen können.

Ein weiteres Beispiel zur Akzeptanzförderung ist, dass die Hardwareanforderungen bezüglich des Arbeitsspeichers von 256MB auf 512MB angehoben wurden, damit alle Anwender des LiMux-Basis-Clients einen flüssigen Arbeitsablauf genießen können.

### 6.3.2. LiMux-Basis-Client

Ein integraler Bestandteil der Vorbereitung der Migration war die Entwicklung des LiMux-Basis-Client. Entgegen anfänglicher Planungen basiert der LiMux-Basis-Client auf Debian und nicht auf dem SuSE-Linux-Enterprise-Desktop. Um den Basis-Client

<sup>48</sup> <http://www.winehq.com/> (zuletzt besucht am: 24.04.2007)

effizient auf den Clients zu installieren, wird das Programm FAI - Fully Automatic Installation<sup>49</sup>, das von der Universität Köln entwickelt wurde, eingesetzt. Der LiMux-Basis-Client besteht aus folgenden Komponenten:

- Betriebssystem: Debian-Linux
- Desktop-Environment: KDE
- Dateimanager: Konqueror
- Office: OpenOffice.org
- E-Mail-Client: Mozilla Thunderbird
- Browser: Mozilla Firefox
- Bildbearbeitung: GIMP

Die Anwendungen werden grundsätzlich alle mit deutschsprachiger Benutzeroberfläche eingesetzt.

Besonderes Augenmerk wurde bei der Entwicklung des Basis-Clients auf die Einfachheit und Benutzbarkeit gelegt. Um diesen Forderungen nachzukommen, gibt es für jede Aufgabe nur genau ein Programm, z. B. gibt es für die Bildbearbeitung nur GIMP, und der Desktop wurde schlicht gehalten. Es wird beabsichtigt, den Basis-Client hinsichtlich seiner Benutzbarkeit vom TÜV zertifizieren zu lassen.

Für die Benutzerverwaltung und die Administration der zu diesem Zweck eingesetzten LDAP<sup>50</sup>-Server wird das Open-Source-Programm Gosa<sup>51</sup> (GONICUS System Administrator) der Firma Gonicus verwendet.

### 6.3.3. WollMux

Ein weiteres essenzielles Bauteil des LiMux-Basis-Client ist der so genannte WollMux. Der WollMux ist ein in JAVA geschriebenes Programm, das sich in OpenOffice.org integriert. Der WollMux ermöglicht es, sehr einfach Textbausteine und Daten in ein OpenOffice.org Dokument einzufügen, um z. B. einen, den formalen Anforderungen der Stadt München entsprechenden, Serienbrief zu erzeugen.

Der WollMux lässt sich sowohl unter Linux als auch unter Windows zusammen mit OpenOffice.org einsetzen, sodass bereits OpenOffice.org effektiv unter Windows verwendet werden kann.

Da die meisten bisher benutzten Microsoft Office Makros und Vorlagen (insgesamt etwa 12.000), durch den WollMux effizient ersetzt werden können, erwartet die Leitung des LiMux-Projektes, dass die Ausgaben für die Umstellung der Makros und Vorlagen um bis zu 50% unterschritten werden können, d. h. diese Ausgaben werden voraussichtlich von geplanten knapp 1,4 Millionen Euro auf etwa 700.000 Euro sinken. Die Umstellung auf den WollMux wird außerdem durch Incentives unterstützt, d. h. wenn ein Anwender die Funktionalität seiner alten Makros und Vorlagen auf den WollMux portiert und auf den WollMux umsteigt, erhält er eine Prämie.

Die Stadt München plant, soweit lizenzrechtlich möglich, sämtliche, im Rahmen des LiMux-Projektes entwickelte Software als Open-Source freizugeben, also auch den WollMux. Dies macht es in Zukunft, insbesondere für andere Kommunen, leichter, auf Open-Source-Software umzusteigen, weil ein Großteil der notwendigen Software dann

<sup>49</sup> <http://www.informatik.uni-koeln.de/fai/> (zuletzt besucht am: 24.04.2007)

<sup>50</sup> LDAP = Lightweight Directory Access Protocol

<sup>51</sup> <https://gosa.gonicus.de/> (zuletzt besucht am: 24.04.2007)

bereits in München entwickelt wurde und nur noch den eigenen Bedürfnissen angepasst werden muss.

Bei dem Einsatz in der öffentlichen Verwaltung zeigt Open-Source seine große Stärke, denn eine öffentliche Verwaltung hat keinen wirtschaftlichen Vorteil davon, Software unter Verschluss zu halten. Und warum sollten für die Entwicklung von funktionsgleicher Software mehrfach Steuergelder ausgegeben werden? Der Einsatz und die Entwicklung von Open-Source-Software in der öffentlichen Verwaltung hilft, Steuergelder zu sparen, weil hier, wenn eine Software erst einmal entwickelt wurde, nur noch die Kosten für die individuelle Anpassung anfallen und keine Lizenz- oder Entwicklungskosten.

### 6.3.4. Fachanwendungen

Als Fachanwendungen werden bei der Stadt München sämtliche Programme bezeichnet, die nicht Teil des Betriebssystems (inkl. graphischer Oberfläche etc.) sind.

Für einen Teil der Fachanwendungen der Stadt München ändert sich nichts, da diese weiterhin auf den BS2000-Servern laufen werden, auf die, sowohl von Windows als auch von Linux, per Terminal-Emulation zugegriffen werden kann. Bei einigen dieser Terminal-Anwendungen wäre eine graphische point-and-click Benutzeroberfläche mehr als hinderlich. So schafft ein Mitarbeiter bei der KFZ-Zulassungsstelle in München bis zu 100 KFZ-Zulassungen pro Stunde, was nur durch sehr effiziente Tastenkombinationen möglich ist. Die BS2000 Anwendungen werden alle von der Stadt München selbst gewartet.

Bei allen anderen Fachverfahren (etwa 300) wurden die Hersteller der jeweiligen Software angeschrieben und befragt, ob sie diese auch als Webanwendung oder als native Linuxanwendung anbieten. Doch der Rücklauf der Anfragen war enttäuschend, denn nur etwa 20% der Hersteller stellten diese in Aussicht oder haben bereits solche Lösungen im Angebot (vgl. [Hoe06], S. 70). Also wurde nach Open-Source-Alternativen gesucht. Für viele Fachverfahren ließen sich Alternativen, z. T. sogar mehrere, finden. In München werden z. B. Microsoft Visio durch Kivio und Photoshop durch GIMP ersetzt.

Fachanwendungen, für die es (noch) keine adäquate Open-Source-Anwendung gibt, werden ausgeschrieben oder per Emulation betrieben, soweit dies lizenzrechtlich möglich ist. München verfolgt folgende Präferenzen: bevorzugt werden Webanwendung gefolgt von nativen Linux-Lösungen. Wenn dies beides nicht möglich ist, wird versucht, die entsprechende Anwendung per Wine (Windows-Emulation) unter Linux zum Laufen zu bringen. Ist dies nicht möglich, so wird auf VMware oder Microsoft Windows Terminal Server zurückgegriffen.

Bei der Ausschreibung für die Softwareentwicklungen oder Softwareanpassungen werden kleine und mittelständische Unternehmen bevorzugt, weil diese laut der Leitung des LiMux-Projektes am flexibelsten sind und am besten die Anforderungen der Stadt München erfüllen können. Bei großen Unternehmen besteht häufig das Problem, dass diese nur nach ihrem eigenem Schema programmieren können, was nicht unbedingt den Anforderungen Münchens entspricht. Außerdem betreibt die Stadt München durch diese Vergabep Praxis eine aktive Mittelstandsförderung, die ausdrücklich politisch gewollt ist.

Bei einigen Programmen verbieten allerdings die Lizenzen, dass sie unter einer Emulation oder auf einem Terminal Server laufen. Dies betrifft bei der Stadt München z. B. das Programm AutoCAD, sodass die Nutzer dieses Programms (einige hundert, also weniger als 5% Beschäftigten) weiterhin einen Windows-PC einsetzen müssen.

Für die Migration der Fachverfahren gibt es folgende Migrationsszenarien:

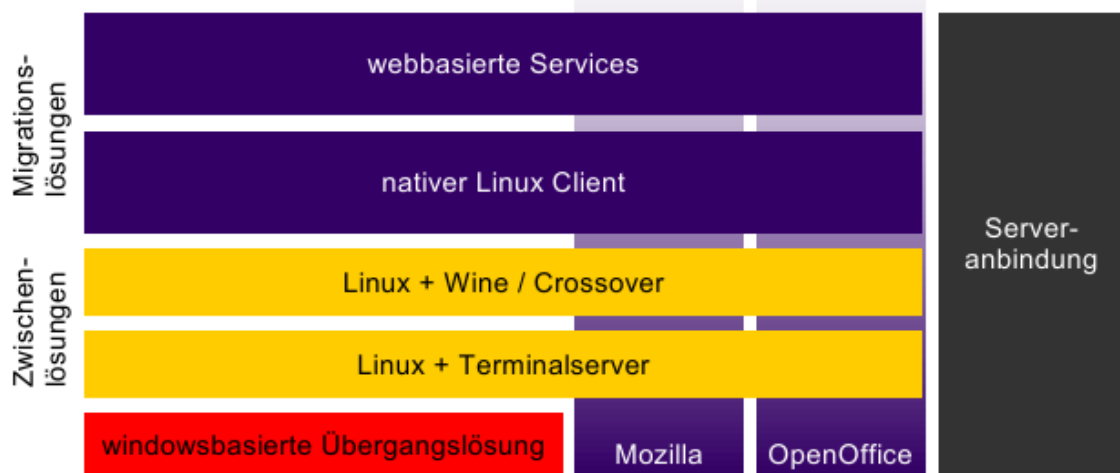


Abbildung 1: Migrationsszenarien der Fachverfahren

Quelle: <http://www.muenchen.de/Rathaus/dir/limux/ueberblick/147210/feinkonzept.html>  
(zuletzt besucht am: 24.04.2007)

Die windowsbasierte Übergangslösung (Windows selbst oder in Verbindung mit VMware), die Zwischenlösungen mit dem Microsoft Windows Terminal Server oder der Windows Emulation und die Migrationslösungen mit nativen Linux Anwendungen oder Webservices.

In jedem Fall sollen Mozilla (Firefox und Thunderbird) und OpenOffice.org (inklusive WollMux) zum Einsatz kommen und auf Serverdienste (z. B. Drucken oder Authentifizieren) zugegriffen werden können.

#### 6.3.4.1. Übergangsszenario

Zu Beginn der Migration laufen noch alle 14.000 PCs der Stadt München unter Microsoft Windows, sie fallen somit alle unter die Kategorien Übergangslösung, d. h. sie sollen soweit wie möglich und so schnell wie möglich durch Linux-Lösungen abgelöst werden. Allerdings lassen sich mittelfristig nicht alle Computer auf Linux umstellen, weil dies durch Lizenzbedingungen einiger der eingesetzten Programme untersagt ist, z. B. AutoCAD (siehe oben). Doch von diesen Lizenzbedingungen sind nur einige hundert PCs betroffen. Auch wenn auf diesen Computern weiterhin Windows laufen wird, so sollen dort trotzdem die Open-Source-Anwendungen verwendet werden, die auf dem LiMux-Basis-Client eingesetzt werden, vorausgesetzt, sie sind als Windows-Version verfügbar. Diese sind z. B. OpenOffice.org, der WollMux, Mozilla Firefox, Mozilla Thunderbird und GIMP. Obgleich durch das Weiterverwenden von einigen Windows-PCs die Betriebssystemlandschaft heterogener wird, wodurch der Administrationsaufwand steigt, wird dieser Aufwand durch den Einsatz der gleichen Anwendungen möglichst niedrig gehalten.

Bei fortschreitender Migration werden noch nicht auf Linux oder Microsoft Windows Terminal Server migrierte Fachanwendungen unter Windows betrieben, die in Virtual Machines (VMware) unter Linux laufen, sodass sie weiterhin verfügbar sind. Dies ist natürlich nur möglich, wenn die Lizenzbedingungen dies zulassen.

### **6.3.4.2. Zwischenlösung**

Bei den Zwischenlösungen werden die Fachanwendungen entweder per Microsoft Windows Terminal Server bereit gestellt oder per Wine an Linux adaptiert.

Bei dem Einsatz des Microsoft Windows Terminal Server laufen die entsprechenden Fachanwendung unter einem Windows Server, auf die vom LiMux-Basis-Client zugegriffen werden kann. Der Einsatz des Microsoft Windows Terminal Servers führt aber dazu, dass weiterhin Lizenzgebühren anfallen.

Bei der Verwendung von Wine wird für jede Fachanwendung, die so auf Linux migriert werden soll, Wine an diese Anwendung angepasst, sodass diese bestmöglich unter Linux lauffähig ist. Wine bildet die Microsoft Windows API<sup>52</sup> nach, d. h. die Anwendung führt ihre Systemaufrufe wie unter Windows aus und Wine übersetzt diese für Linux. Der Vorteil von Wine gegenüber einer Virtual Machine ist, dass nur die Windows API nachgebildet wird und nicht ein kompletter Computer virtualisiert werden muss. Außerdem ist Wine Open-Source und somit lizenzkostenfrei, während bei dem Einsatz von VMware sowohl Lizenzgebühren für VMware selbst als auch für das virtualisierte Windows anfallen. Außerdem sind die unter Wine laufenden Programme auf ihre jeweilige Wine-Umgebung beschränkt, d. h. ihre Sicherheitslücken können nicht das gesamte System oder andere Programme in Gefahr bringen.

Langfristig sollen aber neben den Übergangslösungen auch die Zwischenlösungen verschwinden. Dies soll insbesondere dadurch erreicht werden, dass Fachverfahren, die am Ende ihres natürlichen Lebenszyklus sind und wieder ausgeschrieben werden müssen, als Webservice oder native Linux-Anwendung ausgeschrieben werden.

### **6.3.4.3. Migrationslösung**

Das langfristige Ziel des LiMux-Projektes ist es, alle Fachanwendungen entweder als native Linux-Anwendungen oder als Webservices anzubieten. Bei der Migration werden Webservices bevorzugt, weil auf den Clients nur ein Webbrowser vorhanden sein muss, um diese benutzen zu können, sodass zukünftige Migrationen der Clients ohne viel Aufwand und kostengünstig durchführbar sind.

Bei den Migrationslösungen werden auf den LiMux-Basis-Clients ausschließlich native Linux-Anwendungen betrieben, die zum allergrößten Teil Open-Source-Software sein werden. Eine Ausnahme ist z. B. der PDF-Reader, hier wird auch bei den Migrationslösungen der proprietäre Adobe Reader verwendet, weil dieser zur Zeit der einzige ist, der PDF-Formulare unterstützt.

Alles im allem zielen die unterschiedlichen Migrationsszenarien darauf ab, dass die Migration möglichst glatt verläuft und der laufende Betrieb nicht unnötig gestört wird.

## **6.4. Aktueller Zustand des Projekts**

Nach mehr als zweijähriger Vorbereitungszeit, in der unter anderem die Migrationsszenarien erarbeitet, der LiMux-Basis-Client erstellt und getestet und der WollMux programmiert wurden, sind im September 2006 die ersten Benutzer von Windows auf Linux umgestellt worden. Der erste Bereich, in dem der LiMux-Basis-Client produktiv eingesetzt wurde, war der Kernbereich des Direktoriums, insbesondere das Büro des Oberbürgermeisters. Dadurch, dass der oberste Benutzer der Stadt, der Oberbürgermeister, als einer der Ersten ausschließlich den LiMux-Basis-Client verwendet, werden die mögliche Vorbehalte der Beschäftigten gegenüber der neuen Software minimiert. Wenn

---

52 API= Application Programming Interface; zu deutsch: Anwendungs-Programmier-Schnittstelle

der Oberbürgermeister das neue System verwendet, dann muss es schon hinreichend ausgereift sein.

Bisher wurden einige hundert Benutzer auf den LiMux-Basis-Client migriert, bislang allerdings ausschließlich die einfachen Fälle, die z. B. nur eine Textverarbeitung und ein E-Mail-Programm brauchen. Bis jetzt läuft die Migration relativ reibungslos, d. h. die bereits migrierten Benutzer brauchen kaum Support, was insbesondere auf die intensive Vorbereitung zurückzuführen ist, die neben öffentlichen Präsentationen des LiMux-Basis-Clients auch eine offene Informationspolitik und Schulungen, sowie auf E-Learning im Intranet umfasst.

Neben der Migration der einfachen Fälle werden zur Zeit die weiteren Migrationsszenarien vorbereitet, so wurde z. B. das Programm Microsoft Project 2002, für das noch keine Linux-Alternative gefunden wurde, mit einem extra angepassten Wine unter Linux lauffähig gemacht.

In 2007 sollen 50% aller PCs der Stadt München auf Linux umgestellt werden und bis Ende 2008 sollen es 80% sein. Die restlichen folgen erst, wenn die dafür benötigten Fachverfahren oder entsprechende Alternativen unter Linux verfügbar sind, was durch Ausschreibungen am Ende des Einsatzzyklus der jeweiligen noch nicht migrierten Software erreicht werden soll.

Darüber hinaus soll die Benutzbarkeit des LiMux-Basis-Clients soweit verbessert werden, dass er vom TÜV zertifiziert werden kann.

Des Weiteren sollen alle Entwicklungen der Stadt München, insoweit die Lizenzbedingungen dies zulassen, ab Mitte 2007 als Open-Source freigegeben werden.

Die Resonanz auf das LiMux-Projekt ist bislang fast ausschließlich positiv. So steht die Stadt München im Kontakt mit andern Städten, die auch auf Open-Source setzen wollen. Unter diesen Städten sind z. B. Wien, Paris, Prag, Stockholm und Johannesburg. Neben diesen Städten sind auch die Anwender des LiMux-Basis-Clients positiv überrascht, so gab es z. B. Reaktionen, wie „*das sieht ja aus wie Windows*“, in anderen Worten der Umstellungsaufwand ist nicht so hoch, wie viele erwarten hatten.

## **6.5. Anwendungen des Kriterienkatalogs auf das LiMux-Projekt**

Im folgenden wird der Kriterienkatalog aus Kapitel 4.1 (siehe S. 22) auf das LiMux-Projekt angewendet. Es wird analysiert, welche Auswirkungen die Migration auf Open-Source-Software auf die Ausgaben, die Beschäftigten, die IT und die Politik der Stadt München hat.

### **6.5.1. Strategie**

<b>Kriterium</b>	<b>Evaluation des Kriteriums</b>	<b>Erfüllung des Kriteriums</b>
Notwendigkeit der Migration	Die Migration war notwendig, weil Microsoft den Support für Microsoft Windows NT eingestellt hat.	ja
Zeitpunkt der Migration	Der Zeitpunkt der Migration konnte durch die Kündigung des Supports seitens Microsoft nicht anders gewählt werden.	ja
Evaluation von Alternativen	Es wurde eine Studie [CliStu03] angefertigt, die die Alternativen evaluiert hat.	ja



<b>Kriterium</b>	<b>Evaluation des Kriteriums</b>	<b>Erfüllung des Kriteriums</b>
Entscheidung für je eine Alternative	In München wurde für fast jede eingesetzte Software eine Alternative (als Webservice, nativ, emuliert, virtualisiert oder per TS) gefunden oder erstellt. Ausnahme: AutoCAD, bei dem die Lizenzbedingungen ein Windows voraussetzen.	größtenteils
Entwicklung einer Basis-Systemkonfiguration und eines Installationsverfahren	Im Rahmen des LiMux-Projektes wurde der LiMux-Basis-Client entwickelt, der per FAI automatisch installiert wird.	ja
Testen	Bevor mit den ersten Installationen in 2006 begonnen wurde, wurde die Software intensiv getestet.	ja
Pilotanwender / Rollout	Wahl einer schrittweisen Migration und des Oberbürgermeister-Büros als Pilotanwender.	ja
Ziele realistisch definieren	Begonnen wurde die Migration im Jahr 2003. In 2006 wurden die ersten Clients umgestellt und 2008 soll die Migration abgeschlossen sein. Bislang wurden alle Ziele eingehalten.	ja
Notfallstrategie (was passiert, wenn etwas nicht wie geplant läuft)	Aufgrund der schrittweisen Migration ist im Notfall das Zurückgehen auf einen vorherigen Zustand jederzeit möglich.	ja
Make or buy (selber entwickeln oder einkaufen)	Damit die Kosten für externe Entwicklungen nicht explodieren, wird das Anpassen der Software von der Stadt München selbst übernommen. Bei den Fremdvergaben für die Entwicklung von Software, wird ausschließlich auf kleine, flexible Unternehmen zurückgegriffen.	ja

*Tabelle 10: LiMux - Kriterienkatalog – Strategie*

Zur Strategie der Stadt München lässt sich festhalten, dass sowohl die Migration sorgfältig vorbereitet wurde als auch alle Alternativen evaluiert wurden. Darüber hinaus wurden realistische Ziele gesetzt und durch das intensive Testen, mit der Wahl des Oberbürgermeisters als Pilotanwender und einer schrittweisen Migration wurde sichergestellt, dass die Migration möglichst reibungslos durchzuführen ist.

### 6.5.2. Kosten

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
<b>Kurzfristige Kosteneinsparung</b>	Die kurzfristigen Kosten der Migration auf Open-Source-Software sind höher als sie bei einer Beibehaltung von proprietärer Software wären.	keine Einsparung
Lizenzkosten (Anschaffung)	Die Lizenzkosten reduzieren sich für Open-Source-Software auf Null. Einzig für weiterhin verwendete proprietäre Software (z. B. AutoCAD) fallen Lizenzgebühren an.	große Einsparungen
Hardwarekosten	Die bestehende Hardware kann größtenteils weiterverwendet werden.	kaum erhöhte Ausgaben
Neuentwicklung von Software	Durch die Migration auf Linux müssen viele Programme neu entwickelt werden.	erhöhte Ausgaben
Anpassung von Soft- und Hardware	Fast die gesamte Software muss an die neuen Gegebenheiten angepasst werden. Alte Hardware muss z. T. ersetzt werden.	erhöhte Ausgaben
Reduzierte Produktivität während der Umstellung und Einarbeitung	Durch die Umstellung auf ein anderes System ist mit erhöhten Schulungsbedarf zu rechnen.	kurzfristig verminderte Produktivität
Anpassung von bestehenden Strukturen, z. B. Vorlagen und Daten	Die bestehenden Strukturen müssen an die neue Software angepasst werden, es müssen z.B. sämtlich Vorlagen von Microsoft Office auf den WollMux umgestellt werden.	erhöhte Ausgaben und kurzfristig verminderte Produktivität
<b>Langfristige Kosteneinsparung</b>	Langfristig werden die Kosten durch eine Migration auf Open-Source-Software sinken.	ja
Wartungskosten	Durch die Konsolidierung der bestehenden Softwarelandschaft auf ein System ist mit niedrigeren Wartungskosten zu rechnen.	niedriger
Laufende Lizenzkosten	Nur für die verbliebene proprietäre Software (z. B. Microsoft Windows Terminal Server) fallen weiterhin Lizenzkosten an. Langfristig soll diese durch Open-Source-Software ersetzt werden.	sehr niedrig
Supportkosten	Durch die Konsolidierung der bestehenden Softwarelandschaft auf ein System ist mit niedrigeren Supportkosten zu rechnen.	niedriger
Ausfallkosten	Open-Source-Software zeichnet sich durch seine hohe Stabilität aus, sodass niedrige Ausfallkosten zu erwarten sind.	niedriger

Tabelle 11: LiMux - Kriterienkatalog – Kosten

Durch die Entwicklung neuer Software und den hohen Anpassungsbedarf der neuen Software und der bestehenden Vorlagen und Makros, sowie durch die verminderte Produktivität, die mit einer großen Softwaremigration einher geht, ist kurzfristig mit höheren Kosten zu rechnen, auch wenn die Kosten für Softwarelizenzen aufgrund des hauptsächlichlichen Einsatzes von Open-Source-Software stark reduziert werden.

Langfristig ist jedoch mit Kosteneinsparungen zu rechnen, weil durch die Konsolidierung die Wartungs- und Supportkosten und durch den Einsatz von Open-Source-Software die laufenden Kosten, z. B. die Lizenz- und Ausfallkosten, sinken.

### 6.5.3. Software

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
<b>Verfügbarkeit</b>	Da Open-Source-Software frei ist, ist deren Verfügbarkeit unbegrenzt gewährleistet. Allerdings ist nicht für jeden Anwendungsbereich eine hinreichend gute Open-Source Anwendung verfügbar.	größtenteils ja
Langfristige Verfügbarkeit von Patches und Support	Im Gegensatz zu proprietärer Software lässt sich Open-Source-Software im Notfall selber patchen.	ja
<b>Anpassbarkeit</b>	Dadurch, dass der Quelltext von Open-Source-Software frei ist, lässt sich diese Software unbegrenzt Anpassen	hervorragend
Komplexität	Die Komplexität der Anpassung richtet sich danach, was angepasst werden soll. Eine Quelltext-Veränderung ist sehr komplex, während z. B. das Ändern der Sprache recht einfach ist.	unterschiedlich
Flexibilität	Aufgrund des offenen Quelltextes lässt sich Open-Source-Software sehr flexibel an die eigenen Bedürfnisse anpassen.	hervorragend
<b>Sicherheit</b>	Open-Source-Software zeichnet sich durch eine hohe Sicherheit aus, z. B. werden Sicherheitslücken sehr schnell behoben.	gut
Zugangskontrolle	Der Zugang zu den Computern wird von LDAP-Servern verwaltet, somit ist eine Zugangskontrolle sichergestellt.	ja
Datenschutz	Die Nutzer haben nur Zugriff auf Daten, für die diese authentifiziert sind.	gewährleistet
Malware-Anfälligkeit	Es gibt praktisch keine Viren, Trojaner etc. für Linux.	nicht vorhanden
<b>Benutzbarkeit</b>	Beim LiMux-Projekt wird sämtliche Software auf Benutzbarkeit optimiert.	gut

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
Erwartungskonformität	Das LiMux-Projekt versucht alle Anwendungen erwartungskonform zu machen und z. B. unsinnige Fehlermeldungen zu vermeiden.	gut
Dokumentation	Eine E-Learning-Plattform und Dokumentation ist für jeden Beschäftigen der Stadt München im Intranet verfügbar.	gut
In deutscher Sprache verfügbar	Alle Anwendungen bei der Stadt München werde grundsätzlich auf deutsch betrieben.	ja
<b>Ausgereiftheit</b>	Sämtliche Software, die im Rahmen des LiMux-Projektes eingesetzt wird, wurde vorher auf seine Ausgereiftheit untersucht. Es wird nur Software eingesetzt, die hinreichend ausgereift ist.	hinreichend
Stabilität	Open-Source-Software zeichnet sich in der Regel durch eine hohe Stabilität aus.	gut
Ausfallzeiten / Reparaturzeiten	Die maximalen Ausfall- und Reparaturzeiten sind mit externen Dienstleistern, die die Wartung durchführen, vertraglich vereinbart.	gut
Eigene Erfahrungen mit ähnlicher Software	Vor dem LiMux-Projekt wurde in München nur auf einigen Servern Open-Source-Software eingesetzt.	wenig
Erfahrungen anderer (öffentlicher) Einrichtungen mit der gleichen oder ähnlicher Software	Es gibt bereits einige andere Gemeinden (die im Kapitel Open-Source in der öffentlichen Verwaltung ab S. 28 vorgestellt werden), die auf Open-Source-Software migriert sind, aber keine in der Größe der Stadt München.	einige, aber nicht in vergleichbarer Größe
Wartbarkeit	Open-Source-Software lässt sich aufgrund ihres offenen Quelltextes sehr gut warten.	sehr gut
Hardwareanforderungen	Die Hardwareanforderungen von Open-Source-Software sind relativ niedrig. Ein minimales Linux benötigt z. B. nur 4MB RAM (vgl. [DuInf], S. 387).	niedrig

*Tabelle 12: LiMux - Kriterienkatalog – Software*

Durch den Einsatz von Open-Source-Software besteht eine größere Zukunftssicherheit. Weil der Quelltext von Open-Source-Software verfügbar ist, lässt sich die Software zeitlich unbegrenzt an veränderte Gegebenheiten anpassen und Sicherheitslücken können jeder Zeit behoben werden. Außerdem zeichnet sich Open-Source-Software durch flexible Anpassbarkeit, gute Benutzbarkeit und hohe Sicherheit, sowie Stabilität aus.

Allerdings sind (noch) nicht für alle Anwendungsgebiete Open-Source-Alternativen verfügbar, was eine größere Heterogenität der Softwarelandschaft zur Folge hat. Diese heterogene Softwarelandschaft verursacht einen höheren Wartungsaufwand und eine größere Komplexität.

Außerdem hat es bislang noch keine so große Open-Source-Migration, wie in München, gegeben, sodass kaum auf Erfahrungen zurückgegriffen werden kann.

#### 6.5.4. Personal

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
<b>Motivation</b>	In München werden eine Reihe von motivationsfördernden Maßnahmen durchgeführt, z. B. 512MB RAM für jeden PC und eine offensive Informationspolitik.	wird gefördert
Verbesserung oder Verschlechterung der Arbeitsbedingungen	Das Ziel des LiMux-Projektes ist es, „alles“ besser zu machen, d. h. sämtliche Arbeitsabläufe werden nach Möglichkeit optimiert.	verbessert
Information	Informiert wird per Intranet, Newsletter, Informationsständen und Flyern. Es wird auch über Probleme informiert.	gut
Einbindung des Personals	Die Einbindung des Personals zwecks Feedback ist Teil der offensiven Informationspolitik.	gut
<b>Schulungen</b>	Jeder Beschäftigte wird für die neue Software geschult.	gut
<b>Support</b>	Bislang gab es bei den bereits umgestellten kaum Supportbedarf.	gut

*Tabelle 13: LiMux - Kriterienkatalog – Personal*

Beim LiMux-Projekt steht die Motivation der Mitarbeiter im Vordergrund. Sie werden intensiv über die Migration informiert und z. B. durch Informationsveranstaltungen mit eingebunden. Ziel ist es, die Arbeitsbedingungen der Beschäftigten durch die Migration zu verbessern. Außerdem wird jeder Beschäftigte intensiv für die neue Software geschult und erhält bei Bedarf Unterstützung. Bislang ist allerdings der Supportbedarf sehr niedrig, was wiederum dafür spricht, dass die Migration gut vorbereitet wurde.

#### 6.5.5. IT-Abteilung

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
<b>Erfahrung mit der neuen oder ähnlicher Software</b>	In München wurde bereits vor dem LiMux-Projekt Open-Source-Software auf den Servern eingesetzt. Außerdem stehen die Mitarbeiter der IT-Abteilung Open-Source-Software sehr aufgeschlossen gegenüber.	vorhanden
<b>Genügend Personal vorhanden oder akquirierbar</b>	Zum größten Teil vorhanden.	ja

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
Fähigkeiten	Bislang läuft die Migration ohne größere Probleme, was für die Fähigkeiten der IT-Abteilung spricht.	ja
Schulungen	Aufgrund der komplett neuen Systemumgebung werden auch die IT-Mitarbeiter geschult.	ja
Motivation	Das Klima in der IT-Abteilung des Direktoriums ist hervorragend.	hervorragend

*Tabelle 14: LiMux - Kriterienkatalog – IT-Abteilung*

Die Hauptabteilung Informationstechnologie der Stadt München stand der Migration auf Open-Source-Software von Anfang an positiv gegenüber und bis heute ist die Migrationsfreude dort ungebrochen. Das Arbeitsklima in der Abteilung scheint hervorragend und die Beschäftigten erwecken einen, sehr positiven, fähigen und motivierten Eindruck, was ein Grundstein für den potenziellen Erfolg des LiMux-Projektes ist.

### 6.5.6. Politik

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
<b>Hersteller-unabhängigkeit</b>	Da Open-Source-Software von jedem frei ersetzbar und veränderbar ist, ist eine Herstellerunabhängigkeit gegeben.	ja
Offene Standards	In München wird u. a. OpenOffice.org eingesetzt, das als Dateiformat das von OASIS und ISO standardisierte Open-Document-Format verwendet. Die Verwendung von offenen Standards gewährleistet eine Anwendungsunabhängigkeit.	ja
<b>Mittelstandsförderung</b>	Mit der Erstellung und Wartung von Software werden von der Stadt München ausschließlich kleine und mittelständische Unternehmen beauftragt, weil diese flexible auf die Bedürfnisse der Stadt reagieren können.	ja
<b>Ansehen in der Bevölkerung</b>	Das LiMux-Projekt ist mittlerweile auch außerhalb Münchens bekannt, was vor allem mit dem Besuch des Microsoft CEOs Steve Ballmer zu tun hat. Das Feedback, das das LiMux-Projekt von anderen Gemeinden (z. B. Wien und Paris) und von der Bevölkerung (z. B. auf der Messe Systems) erhält, ist überwiegend positiv.	gut

Kriterium	Evaluation des Kriteriums	Erfüllung des Kriteriums
Entscheidungswille	Der Stadtrat hat sich erst, nachdem die Client-Studie [CliStu03] den Umstieg auf Open-Source-Software empfohlen hat, für diese Migration entschieden, obwohl die IT-Abteilung des Direktoriums dies schon vorher empfohlen hatte.	vorhanden
Politische Agenda	Für den Einsatz von Open-Source-Software haben in München die SPD, die Grünen und die FDP gestimmt, dagegen war einzig die CSU.	überwiegend pro Open-Source
Mehrheiten	Die große Mehrheit des Münchner Stadtrats unterstützt das LiMux-Projekt.	pro Open-Source

*Tabelle 15: LiMux - Kriterienkatalog – Politik*

Für die Mehrheit der Politiker der Stadt München stand von Anfang an die Herstellerunabhängigkeit bei der Migration im Vordergrund. Sie wollen in Zukunft selber entscheiden können, wann und wie eine Migration durchzuführen ist. Dies lässt sich nur durch den Einsatz von Open-Source-Software und das Einhalten von offenen Standards erreichen. Nur bei Open-Source-Software kann man selbst entscheiden, wie lange das eingesetzte Produkt gepflegt wird, und nur offene Standards gewährleisten, dass man in Zukunft mit einem anderen Programm die selben Daten problemlos bearbeiten kann. Somit ist folglich die Entscheidung für Open-Source-Software und offene Standards die einzig logische.

Darüber hinaus ist für die Politik die Förderung von regionalen kleinen und mittelständischen Betrieben von großer Bedeutung. Dieses Ziel ist unvereinbar mit dem Einsatz von Software eines US-amerikanischen Großunternehmens (z. B. Microsoft). Folglich ist hier die Wahl von Open-Source-Software die richtige, denn der Support dieser Software wird auch von kleinen Unternehmen aus Deutschland angeboten.

Auch ist die Unterstützung durch die Politik relativ verlässlich, denn eine überwältigende Mehrheit (SPD, Grüne, FDP und Splittergruppen) der Münchner Stadträte steht hinter dem LiMux-Projekt. Einzig die CSU ist gegen die Migration der Stadt München auf Open-Source-Software.

Des Weiteren ist das Feedback der Bevölkerung und anderer Gemeinden durchweg positiv, was wiederum die politische Unterstützung festigt.

## **6.6. Chancen und Risiken**

Nach der Analyse des LiMux-Projektes lassen sich folgende Chancen und Risiken erkennen:

### **6.6.1. Chancen**

Die Chancen für das LiMux-Projekt bestehen darin, dass die zukünftigen Einsparungen größer sein könnten als erwartet, dass die Beschäftigten mit der neuen Software produktiver arbeiten als zuvor und dass der Wartungsaufwand geringer ist.

Des Weiteren könnten andere Städte und Gemeinden sich ein Beispiel an München nehmen und ebenfalls auf Open-Source-Software umsteigen. Dieses wird insbesondere

dadurch vereinfacht, dass die Stadt München plant, sämtliche Softwareentwicklungen für das LiMux-Projekt als Open-Source freizugeben. Somit wird der Aufwand für andere öffentliche Verwaltungen, auf Open-Source-Software umzusteigen, deutlich vermindert, weil diese auf die für München entwickelte Software zurückgreifen können. Auf diese Art können auch andere öffentliche Verwaltungen vom LiMux-Projekt profitieren und Steuergelder sparen.

Der vermehrte Einsatz von Open-Source-Software in der öffentlichen Verwaltung könnte die Verbreitung dieser insgesamt erhöhen, weil so die freie Software in der Öffentlichkeit bekannter wird und weil gezeigt wird, dass Open-Source-Software funktioniert und benutzbar ist.

### **6.6.2. Risiken**

Die Risiken des LiMux-Projektes liegen darin, dass die Migration teurer werden könnte als erwartet, dass die Beschäftigten nicht mit der neuen Software arbeiten wollen und dass die einzusetzende neue Software nicht das leistet, was man von ihr erwartet.

Diesen Risiken wird in München versucht, durch eine intensive Vorbereitung und detaillierte Planung der Migration, sowie durch offensives Informieren und Einbinden der Beschäftigten entgegenzuwirken.



## 7. Fazit

Alles in allem lässt sich feststellen, dass eine Migration auf Open-Source-Software in der öffentlichen Verwaltung möglich ist. Dies lässt sich aus den erfolgreichen Migrationen der Städte Schwäbisch Hall und Treuchtlingen und der niedersächsischen Polizei folgern. Allerdings ist bei einer solchen Migration zu beachten, dass (noch) nicht für jeden Anwendungsbereich eine Open-Source-Alternative zur Verfügung steht, sodass auch nach der Migration noch vereinzelt proprietäre Software eingesetzt werden muss. Doch je mehr Kommunen den Umstieg auf Open-Source-Software vollziehen, desto mehr werden diese Lücken durch Eigenentwicklungen geschlossen, die als Open-Source freigegeben werden. Es gibt für öffentliche Einrichtungen keinen Grund, ihre Eigenentwicklungen verschlossen zu halten, denn sie haben erstens keinen Wettbewerbsvorteil davon und zweitens, was von Steuergeldern bezahlt wurde, gehört prinzipiell allen und sollte nicht nochmals von Steuergeldern finanziert entwickelt werden müssen, dies wäre Steuergeldverschwendung.

Open-Source-Software bietet ein hohes Einsparpotenzial für die öffentliche Verwaltung, denn die Aufgaben, die z. B. Gemeinden zu erledigen haben, unterscheiden sich von Gemeinde zu Gemeinde nur minimal, sodass sie die gleiche Software mit kleineren Anpassungen einsetzen können. Ist diese Software erst einmal entwickelt und als Open-Source veröffentlicht, so sind die Kosten für die Softwareumstellung anderer Gemeinden sehr niedrig.

Allerdings muss eine Softwaremigration sorgfältig vorbereitet und eine geeignete Strategie für die Migration entwickelt werden, damit sie ein Erfolg werden kann. Die intensive, etwa 3-jährige Vorbereitung der Migration der Stadt München ist ein Hauptgrund, warum die Migration dort bislang ohne größere Komplikationen verläuft. Weitere essenzielle Bedingungen für eine erfolgreiche Migration sind fähiges und motiviertes IT-Personal, sowie die Mitnahme der Beschäftigten, was durch eine offensive Informationspolitik und durch gute Schulungen zu erreichen ist. Diese Bedingungen werden bei dem LiMux-Projekt alle erfüllt, sodass dort erwartungsgemäß die Migration auf Open-Source-Software bislang ein voller Erfolg ist.

Des Weiteren muss bei einer Migration auf Open-Source-Software mit kurzfristig höheren Kosten gerechnet werden, die z. B. für die Entwicklung und Anpassung der neuen Software, sowie für Schulungen anfallen. Allerdings ist zu erwarten, dass langfristig diese Kosten mehr als ausgeglichen werden, weil für Open-Source-Software keine Lizenzkosten anfallen. Außerdem ist man, wenn man auf Open-Source-Software umgestiegen ist, herstellerunabhängig, d. h. man kann frei entscheiden, wann, wie und auf welche Software bei der nächsten Migration umgestiegen wird. Außerdem kann man bei Bedarf die Open-Source-Software an sich ändernde Gegebenheiten selbst anpassen und Sicherheitslücken notfalls selbst patchen.

Darüber hinaus ist eine Migration von Open-Source-Software mit offenen Datenformaten auf andere Open-Source-Software deutlich einfacher und somit billiger als eine Migration von proprietärer Software mit proprietären Datenformaten auf Open-Source-Software.

Ein weiterer Vorteil von Open-Source-Software gegenüber proprietärer Software sind ihre niedrigen Hardwareanforderungen. Ein Linux-Betriebssystem mit graphischer Oberfläche benötigt nur mindestens einen Pentium PC und 256MB RAM, diese Anforderungen werden von den meisten Computern in der öffentlichen Verwaltung erfüllt. Um das neuste Betriebssystem aus dem Hause Microsoft, Windows Vista, flüssig betreiben zu können, bedarf es hingegen mindestens eines Gigahertz-Prozessors, 1GB

RAM und einer leistungsfähigen Grafikkarte, diese Anforderung kann kaum ein Computer der öffentlichen Verwaltung erfüllen. Somit sind die Hardwarekosten bei einer Migration auf Windows Vista ein erheblicher Kostenfaktor. Darüber hinaus hat Microsoft der neusten Version von Microsoft Office, Microsoft Office 2007, eine völlig neue Oberfläche verpasst, sodass OpenOffice.org 2 den älteren Microsoft Office Versionen deutlich ähnlicher ist und somit die Schulungskosten bei einer Migration auf Microsoft Office 2007 deutlich höher ausfallen dürften als die Schulungskosten bei einem Umstieg auf OpenOffice.org 2.

Während die Migrationskosten für eine Migration auf Microsoft Produkte stetig steigen, sinken die Kosten für eine Migration auf Open-Source-Software. Die Migrationskosten bei einer Migration auf Microsoft Produkte steigen insbesondere durch steigende Hardwareanforderungen, steigende Schulungskosten und steigende Lizenzkosten. Die Migrationskosten bei einer Migration auf Open-Source-Software sinken, insbesondere durch verbesserte Verfügbarkeit von Alternativen, sinkende Schulungskosten, die mit einer größeren Verbreitung der Open-Source-Software einhergehen, und durch den Erfahrungsaustausch zwischen den öffentlichen Einrichtungen, z. B. München und Wien.

Ein weiterer Vorteil der Migration auf Open-Source-Software ist, dass die Steuergelder für regionale kleine und mittelständische Unternehmen ausgegeben werden können, während beim Einsatz von Microsoft Produkten ein Großteil der Gelder in die USA fließen.

Insgesamt ist folglich die Migration auf Open-Source-Software für öffentliche Einrichtungen möglich, und sie wird immer attraktiver.

Somit ist Open-Source-Software eine Alternative für die öffentliche Verwaltung.

## Anhang

### **Interview vom 15.11.2006 mit Manfred Lubig-Konzett (stv. Projektleiter LiMux)**

(Eine Einverständniserklärung zur Veröffentlichung dieses Interviews liegt vor.)

**Frage:** Was war der Hauptgrund, auf Open-Source-Software umzusteigen?

**Antwort:** Der Umstieg war eigentlich nicht geplant, aber die Firma Microsoft ließ den Support für Windows NT auslaufen, sodass ein Migrationszwang bestand.

Die Stadt München war sehr zufrieden mit Windows NT, weil es weitgehend stabil und sicher lief.

2003 gab der Münchner Stadtrat den Auftrag, Alternativen zu evaluieren. Eine entsprechende Studie [CliStu03] wurde von der Firma Unilog Integrata erstellt, die zu dem Schluss kam, dass eine Migrationen auf Open-Source-Software möglich und strategisch am günstigsten ist.

**Frage:** Welche Kriterien waren bei der Migration entscheidend?

**Antwort:** Herstellerunabhängigkeit, eigenständige Entscheidungsmöglichkeiten, wann eine Migration durchgeführt werden soll, Mittelstandsförderung, Support und Sicherstellung des laufenden Betriebs.

Die Wartung der Software wird von Dienstleistern gemacht, und es gibt definierte Vorgaben, wie häufig Fehler auftreten dürfen und wie schnell sie behoben werden müssen.

Die Sicherstellung des laufenden Betriebs ist besonders wichtig in den Dienststellen mit Publikumsverkehr, z. B. der Kfz-Zulassungsstelle. Dort bearbeitet ein Beschäftigter bis zu 100 Zulassungen pro Stunde, was nur möglich ist durch eine textbasierte Eingabe, diese erfolgt per Terminal-Emulation an einem BS2000-Server. Die Terminal-Emulation ist sowohl für Windows als auch für Linux nativ verfügbar.

**Frage:** Was waren die schwerwiegendsten Gegenargumente?

**Antwort:** Softwarepatente. Die EU plante zeitweise Softwarepatente einzuführen, dies wurde als problematisch angesehen.

Bei der Stadt München gibt es 17 unabhängige IT-Betriebseinheiten. Es besteht eine zentrale Beschaffung und die Strategie wird von oben vorgegeben.

Es bestanden folgende Vorbehalte:

1. Die Migration der bestehenden Fachverfahren (insgesamt ca. 300), sind zum großen Teil nur für Windows vorhanden. Eine Umstellung würde Zeit und Geld kosten.
2. Der Bestand an Microsoft Office Formularen, Makros, Dokumentenformaten und Formblättern. Insgesamt gibt es etwa 12.000 Vorlagen und Makros.
3. Infrastruktur, Benutzerverwaltung, Softwareverteilung (Patches) und Integration in die Backend-Struktur (z. B. Novell Netware).

**Frage:** Wie konnten diese überwunden werden?

**Antwort:** Die Softwarepatente wurden letztendlich vom Europäischen Parlament abgelehnt, nachdem sich u. a. die Münchner Politik, allen voran der Oberbürgermeister Ude

(SPD), dagegen eingesetzt hat. Es wurde auch ein Gutachten [Fro04] zu den Softwarepatenten angefertigt.

Akzeptanzförderung: z. B. 512MB RAM für alle PCs und laufende Informationen per Papier, per Intranet, per Mail-Verteiler (Newsletter) oder per Informationsveranstaltungen, bei denen die Beschäftigten den LiMux-Basis-Client selbst ausprobieren können.

Alle Firmen, die die ca. 300 Fachverfahren erstellt haben, wurden angeschrieben, ob sie diese auch als Linux-Version anbieten oder dies planen. Das Resultat war enttäuschend, denn nur etwa 10% boten dies an oder planten, es anzubieten. Also wurde nach Alternativen gesucht, wobei folgende Präferenz vorgegeben wurde: am liebsten Webservices, dann native Linux-Lösungen, Wine-basierte Lösungen, VMware-basierte Lösungen und notfalls auch Microsoft Windows Terminal Server-Lösungen.

Es wurden z. B. folgende Alternativen gefunden: Microsoft Visio wird durch Kivio ersetzt und Photoshop durch GIMP. Z. T. gab es sogar mehrere Alternativen, z. B. Dia und Kivio, sodass eine Entscheidung für eine getroffen werden musste. Andere Produkte können auch weiterhin verwendet werden, so ist z. B. der Adobe Reader, sowohl für Windows als auch für Linux verfügbar.

Letztendlich wurde nach intensiven Recherchen für jedes Produkt eine Alternative oder zumindest eine Zwischenlösung gefunden. Microsoft Projekt 2002 wird z. B. unter Wine betrieben.

Für alle Anwendungen gilt, dass sie grundsätzlich mit einer deutschen Oberfläche betrieben werden.

Für die Erstellung und Wartung der Software werden ausschließlich kleine und mittelständische Unternehmen herangezogen, weil diese München besser „verstehen“ und flexibler sind als Großunternehmen. Die Vergabe erfolgt durch Ausschreibung mit Pflichten- (wie) und Lastenheft (was durchzuführen ist).

Die Konsolidierung der Makros und Vorlagen wird anhand des neu entwickelten WollMux realisiert. Der WollMux ist ein in JAVA geschriebenes Programm, das sich in OpenOffice.org einklinkt und das komfortable Erstellen von Dokumenten, z. B. anhand von Textbausteinen erlaubt. Jeder Beschäftigte, der seine Vorlagen und Makros auf den Wollmux umstellt, erhält eine Prämie.

Mittlerweile wird erwartet, dass durch den Wollmux etwa 50% der Kosten für die Umstellung von Makros und Vorlagen eingespart werden können.

Gemäß dem Open-Source-Gedanken arbeitet die Stadt München mit anderen Kommunen zusammen. Das Projekt ist auf globales Interesse gestoßen. Es bestehen z. B. Kontakte nach Stockholm, Prag, Spanien, Paris, Johannesburg und Wien.

Es lassen sich leider nicht alle Programme umstellen, so verbietet z. B. die Lizenz von AutoCAD den Betrieb auf einem Terminal Server, sodass Beschäftigte, die dieses Programm benötigen (wenige hundert), weiterhin mit Windows arbeiten müssen.

**Frage:** *Wie haben die Beschäftigten der Stadt auf die Entscheidung reagiert?*

**Antwort:** Die Endbenutzer waren erstaunt, mit wie wenig Aufwand die Umstellung verbunden ist.

**Frage:** *Wie werden sie bei der Migration mit eingebunden?*

**Antwort:** Durch Informationsveranstaltungen, durch das Einbinden von Führungskräften, durch Pressemitteilungen und durch ein E-Learning-Angebot im Intranet mit Ton- und Video-Hilfen, sodass eine Selbsthilfe möglich ist.

Außerdem wird eine Neuorganisation der IT vorgenommen. Das Motto ist „wir wollen alles besser machen“.

Darüber hinaus erhält jeder Beschäftigte eine 12,5 Tage lange Schulung. Die Schulungen machen insgesamt etwa 1/3 der Gesamtkosten der Migration aus.

**Frage:** *Wie viel Support brauchen die bereits umgestellten Beschäftigten?*

**Antwort:** Der Supportbedarf läuft gegen Null. Nur ohnehin Schulungsresistente machen Probleme. Die Schulungen zielen hauptsächlich auf bisherige Schulungsdefizite ab.

**Frage:** *Wie sieht die Umstellung praktisch aus, d. h. wie wird z. B. eine konkrete Fachanwendung migriert?*

**Antwort:** Wenn eine Neuausschreibung für ein Fachverfahren ansteht, so wird dieses entweder als Webservice oder als native Linux-Anwendung ausgeschrieben, sodass langfristig auch die Übergangslösungen mit Wine, VMware oder Microsoft Windows Terminal Server abgelöst werden.

**Frage:** *Was wurde bisher erreicht?*

**Antwort:** Bislang wurden etwa 550 Personen, allen voran der Oberbürgermeister, auf Linux umgestellt.

**Frage:** *Gab es positive Überraschungen?*

**Antwort:** Reaktionen wie „das sieht ja aus wie Windows“.

Die Stadt München erhält immense Unterstützung (außer von Microsoft). Auf der Systems 2006, wo das LiMux-Projekt mit einem Stand vertreten war, gab es über 1000 Kontakte. Die Resonanz war äußerst positiv.

Beim Projekt darf auch mal was schief gehen. Es wird aber sehr offen, transparent und ohne Beschönigungen damit umgegangen.

**Frage:** *Wurde die Kostenplanung bislang eingehalten, über- oder unterschritten?*

**Antwort:** Die Kosten blieben bislang im Rahmen. Insbesondere im Office-Bereich wird mit einer 50% Unterschreitung des Budgets gerechnet.

**Frage:** *Wie sehen die nächsten Schritte aus?*

**Antwort:** 2007 wird mit der Massenumstellung begonnen. Ende 2007 sollen 50% umgestellt sein. Mitte 2007 soll begonnen werden, die Eigenentwicklungen an die Community zurückzugeben, d. h. diese sollen als Open-Source freigegeben werden.

Im Jahr 2006 wurde vor allem entwickelt und die Migration vorbereitet, z. B. GOsa (GOnicus System Administrator), FAI (Fully Automatic Installation) und LDAP (Lightweight Directory Access Protocol) angepasst.

Besonders im Vordergrund stehen die Qualitätssicherung und die Usability, welche vom TÜV zertifiziert werden soll. Bei der Usability ist vor allem die Ergonomie, vernünftige Fehlermeldungen und die Erwartungskonformität von Bedeutung.

**Frage:** *Welche Lehren können aus dem LiMux-Projekt für andere migrationswillige öffentliche Verwaltungen gezogen werden?*

**Antwort:** Die reine Umstellung ist kein Problem, wenn die Schulung und die Kommunikation stimmen. Es ist wichtig, allen klar zu machen, dass die Migration sinnvoll ist. Es sollte vorher das Systemmanagement geklärt werden, d. h. wie funktioniert die Administration, die Konfiguration, die Patch-Verwaltung und der Anmeldedienst.

Es gibt keine Lösung von der Stange.

Die Systemverwaltung kostet. Es ist eigenes Personal nötig, da sonst die Kosten davon laufen. Man sollte nur etwa 1/3 extern vergeben. Wenn das Personal schon in der Unix-Szene Erfahrung hat, sollte man auf jeden Fall umstellen. Wenn die IT-Abteilung nur aus einem MCSE (Microsoft Certified Systems Engineer) besteht, dann sollte man es lieber lassen. Aber generell ist eine Migration auf Open-Source-Software zu empfehlen, man sollte aber bedenken, dass neue Strukturen Zeit und Aufwand brauchen.

*Vielen Dank für das Interview!*

## Literaturverzeichnis

- [Alt05] Altenhörner, R. (2005), *Open Source Software - definition, licensing models and organisational consequences (introduction)*, World Library and Information Congress: 71th IFLA General Conference and Council "Libraries - A voyage of discovery" August 14th - 18th 2005, Oslo, Norway.
- [Apache05] Apache Foundation (2005), *About the Apache HTTP Server Project*, [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html), zuletzt besucht am: 24.04.2007.
- [Bräu05] Bräuner, H. (2005), *Linux im Rathaus Ein Migrationsprojekt der Stadt Schwäbisch Hall*, Open Source Jahrbuch 2005, S. 37-50, Lehmanns Media.
- [BroCIT03] k. A. (2003), *Der Brockhaus Computer und Informationstechnologie*, F. A. Brockhaus.
- [Brockh06] k. A. (2006), *Brockhaus: Enzyklopädie in 30 Bänden, 21. Auflage*, Bibliographisches Institut & F. A. Brockhaus AG.
- [CliStu03] k. A. (2003), *Projekt Client Studie der Landeshauptstadt München*, UNILOG Integrata Unternehmensberatung GmbH.
- [Dist07] k.A. (2007), *Distrowatch.com*, <http://www.distrowatch.com/>, zuletzt besucht am: 24.04.2007.
- [DuInf] Claus, V., Schwill, A. (2006), *Duden Informatik A-Z, 4. Auflage*, Dudenverlag.
- [FeFi01] Feller, J., Fitzgerald, B. (2001), *Understanding Software Development*, Addison-Wesley.
- [Fro04] Rechtsanwalt Sedlmaier, R., Patentanwalt Gigerich, J. (2004), *Kurzgutachten Rechtliche Bedingungen und Risiken der Landeshauptstadt München für den Einsatz von Open-Source Software*, FROHWITTER Patent- und Rechtsanwälte, <http://www.ris-muenchen.de/RII/RII/DOK/SITZUNGSVORLAGE/517379.pdf>, zuletzt besucht am: 24.04.2007.
- [Gan04] Ganten, P. (2004), *Erfolgsfaktoren bei der Einführung von Linux in Unternehmen*, Open Source Jahrbuch 2004, S. 249-268, Lehmanns Media.
- [Gra04] Grassmuck, V. (2004), *Freie Software Zwischen Privat-und Gemeineigentum, 2.Auflage*, Bundeszentrale für politische Bildung (bpb).
- [HaHo03] Hang, J., Hohensohn, H. (2003), *Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte*, C-LAB.
- [Hoe06] Hoegner, W. (2006), *Das Projekt LiMux - Freie Software für die Münchner Verwaltungsclients*, Open Source Jahrbuch 2006, S. 59-72, Lehmanns Media.
- [Ker03] Kern, U. et al. (2003), *Studien zu den Einsatzmöglichkeiten von OpenOffice*, Fachhochschule für Ökonomie und Management.
- [KiMa95] Kimball, S., Mattis, P. (1995), *ANNOUNCE: The GIMP*, <http://groups.google.com/group/comp.os.linux.development.apps/msg/b5a9a98ef1e9fd4d?output=gplain>, zuletzt besucht am: 24.04.2007.
- [Lutz04k] Lutz, B. et al. (2004), *Studie OSS. Open Source Software am Arbeitsplatz im Magistrat Wien. - Kurzfassung*, MA 14-ADV.
- [Lutz04l] Lutz, B. et al. (2004), *Studie OSS. Open Source Software am Arbeitsplatz im Magistrat Wien. - Langfassung*, MA 14-ADV.

- [Lutz06] Lutz, B. (2006), *WIENUX und OpenOffice - die freiwillige Migration im Magistrat der Stadt Wien*, Open Source Jahrbuch 2006, S. 73-82, Lehmanns Media.
- [MiLeiFa06] Bundesministerium des Inneren (2006), *Migrationsleitfaden 2.1 - Leitfaden für die Migration der Basissoftwarekomponenten auf Server- und Arbeitsplatz-Systemen*, KBSt.
- [Moz03] Mozilla Foundation (2003), *About the Mozilla Foundation*, <http://www.mozilla.org/foundation/>, zuletzt besucht am: 24.04.2007.
- [Netc07] Netcraft Ltd (2007), *Web Server Survey Archives*, [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html), zuletzt besucht am: 24.04.2007.
- [Netsc98] Netscape (1998), *Netscape announces mozilla.org, a dedicated team and web site supporting development of free client source code.*, <http://wp.netscape.com/newsref/pr/newsrelease577.html>, zuletzt besucht am: 24.04.2007.
- [Nivad04] k. A. (2004), *Polizei in Niedersachsen setzt bei Neuentwicklung auf Open-Source*, Splitter - IT-Nachrichten für die Berliner Verwaltung Nr. 2/2004, S. 4ff., Landesbetrieb für Informationstechnik (LIT) Berlin.
- [OS99] k. A. (1999), *Open Source kurz & gut*, O'Reilly.
- [OsiOsd06] Open Source Initiative (2006), *The Open Source Definition*, <http://www.opensource.org/docs/definition.php>, zuletzt besucht am: 24.04.2007.
- [OSJb06] Lutterbeck, B. et. al. (2006), *Open Source Jahrbuch 2006*, Lehmanns Media.
- [Ren05] Renner, T. et al. (2005), *Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit - Eine Studie der Fraunhofer-Gesellschaft*, Fraunhofer IRB Verlag.
- [Schnei04] Schneider, M. (2004), *Tausend Tage Tux in Treuchtlingen*, <http://www.pro-linux.de/berichte/tausend-tage-tux.html>, zuletzt besucht am: 24.04.2007.
- [Treucht04] k. A. (2004), *Die Stadt Treuchtlingen schafft Handlungsspielraum mit ThinClients*, Splitter - IT-Nachrichten für die Berliner Verwaltung Nr. 2/2004, S. 14f., Landesbetrieb für Informationstechnik (LIT) Berlin.
- [Trid97] Tridgell, A. and the Samba Team (1997), *A bit of history and a bit of fun*, <http://www.rxn.com/services/faq/smb/samba.history.txt>, zuletzt besucht am: 24.04.2007.
- [Wie04] Wieland, T. (2004), *Stärken und Schwächen freier und Open-Source-Software im Unternehmen*, Open Source Jahrbuch 2004, S. 107-119, Lehmanns Media.



*„Ich versichere, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Alle Quellen, die dem World Wide Web entnommen oder in einer sonstigen digitalen Form verwendet wurden, sind der Arbeit beigefügt“*

(Benjamin Stukenberg)