

# 1. Einführung

## Philosophie von „Open Source“

Im Jahre 1984 startete Richard Stallman, ein Forscher der MIT Artificial Intelligence Lab, das GNU Projekt. Ein Projekt, das das einfache Ziel hatte, Software für alle kostenlos zugänglich zu machen. Für ihn war wichtig, dass Wissen, das in jedem ausführbaren Programm steht, uneingeschränkt mit anderen teilen zu können. Wäre die Software nicht kostenlos verfügbar, so glaubte er, würden einige wenige Personen potentiell den gesamten Softwaremarkt dominieren.

Während viele Software Produzenten mit kommerziellem Interesse den Quellcode schützen wollen, sieht Stallman ihn als wissenschaftliche Forschungsarbeit, welche allen uneingeschränkt weitergegeben werden soll. Mit dieser wissenschaftlichen Methode kann eine große Zahl von interessierten Personen von Programmen lernen, sie weiterentwickeln oder verbessern.

Nachdem sich immer mehr Anhänger diesem Projekt angeschlossen hatten, bildete sich schließlich die „Free Software Foundation“, welche den neuen Software-Typ „Free Software“ definierte. Stallman vertrat nämlich die Theorie, dass Innovationen ohne frei verfügbarer Software in der Software-Entwicklung nicht möglich seien. Für ihn war es aber klar, dass Personen oder Firmen versuchen würden, mit der frei verfügbaren Software die eigenen Geschäftsinteressen zu verfolgen.

## General Public License (GPL)

Nach eigenen Weiterentwicklungen auf der Basis der frei verfügbaren Software, würden sie die neue Software vor einer weiteren freien Verteilung mit einer Lizenz schützen.

Durch diese Geschäftsstrategie sichern sie sich die Vermarktungsrechte des gesamten Software-Produktes. Um das zu verhindern, reagierte er schließlich mit einer neuartigen Lizenz. Die sogenannte „General Public License“(GPL), welche das Kopieren und Weitergeben von Software nur dann erlaubt, wenn andere nicht daran gehindert werden, das selbe mit dieser zu tun.

GPL legt im speziellen fest, dass Software, die aus Teilen von Software unter GPL bestehen, auch unter GPL lizenziert werden müsse. Damit wird die freie Verfügbarkeit auch für die weiterentwickelte „Free Software“ garantiert.

Im Jahre 1997 stellte eine Gruppe von führenden „Free Software“ Anhängern, unter ihnen Eric Raymond und Tim O'Reilly die neue „**Open Source**“ **Software** vor. Sie befürchteten, dass die in der „Free Software“ enthaltene "Anti-business" Botschaft das Potential von „Free Software“ schmälert. Sie bietet im Bezug auf ihre Lizenzbestimmungen größere Freiheiten, da dieser im Gegensatz zu „Free Software“ auch mit herkömmlicher Software zusammen kombiniert werden darf, ohne dass die dabei weiterentwickelte Software unter GPL lizenziert werden muss. Das heißt, dass Software, die auf „Open Source“ aufbaut, nicht frei verfügbar gemacht werden muss. Somit kann der Quellcode mit Hilfe einer Lizenz geschützt werden und braucht deswegen nicht offengelegt zu werden.

## Grundfreiheiten von „Open Source Software“

„Open Source Software“ kann aber grundsätzlich nach Gehrling mit den folgenden drei Grundfreiheiten definiert werden:

- Freiheit, die Arbeitsweise des Programms zu verstehen und es für eigene Zwecke einzusetzen, dazu wird der Quellcode offengelegt
- Freiheit, das Programm egal für welchen Zweck einzusetzen
- Freiheit, Kopien anzufertigen sowie das Programm zu verändern und weiterzugeben im gewerblichen, wie auch im privaten Gebrauch

Oft werden die beiden Begriffe „Free Software“ und „Open Source“ Software in der Praxis gleichbedeutend verwendet, obwohl sie sich in einem wesentlichen Punkt unterscheiden. „Free Software“ steht grundsätzlich für Freiheit oder freie Verfügbarkeit von Software:

Wie ein öffentliches Gut quasi steht Software jedermann zur Verfügung. Bei der Entwicklung von „Open Source“ Software stehen jedoch wirtschaftliche Interessen stärker im Vordergrund. Die freie Verfügbarkeit der „Open Source“ Software wird soweit eingeschränkt, dass ihre Entwicklung und Vermarktung wirtschaftlich interessant wird. Seit 1997 steigt der Einfluss von „Open Source“ Software im Geschäftsbereich massiv. Sie bietet mittlerweile hochwertige Alternativen im Bereich von Software-Anwendungen an. Es gibt zum Beispiel im Office-Bereich neben der nicht öffentlich zugänglichen Lösung von Microsoft Word, mindestens zwei ernstzunehmende Lösungen, welche als freie Systeme zur Verfügung stehen: Star Office und Open Office.

Bezüglich des Internets wurde nachgewiesen, dass über die Hälfte aller Web-Seiten verschwinden würden, wenn es keine „Free Software“ gäbe. Dieser beträchtliche Teil wird mehrheitlich von Servern mit der frei verfügbaren „Apache“ Server-Software im Internet angeboten.

Ebenfalls könnten Postzustellungsprogramme keine elektronischen Briefe mehr verschicken, denn die dafür notwendige „sendmail“ ist „Open Source“ Software. Auch wichtige Software-Entwicklungstools könnten nicht mehr verwendet werden, da diese unter anderem von Compilern und Debuggern der „Free Software Foundation“ abhängig sind.

## Kommerzielle Software

Der Gegensatz von frei zugänglicher Software ist die **kommerzielle Software**.

Diese Software darf nicht frei für jeden Zweck eingesetzt werden. Sie darf auch nicht studiert oder frei kopiert und verteilt oder verändert werden. Sie ist ein erwerbbares Programmstück, welches unter einer proprietären („Proprietär“ = Eigentümer) Lizenz an andere verteilt wird. Die „copyright“-behaftete Software darf nur dann im begrenzten Rahmen genutzt werden, wenn die Lizenz akzeptiert und ein fest definiertes Nutzungsrecht käuflich erworben wurde. Der größte Vertreter der kommerziellen Software ist **Microsoft**.

Software ist Teil von Informations- und Kommunikationstechnologien (IKT), da sie die notwendigen digitalen Daten für den Einsatz der Hardware bereitstellt.

Sie ist die Schnittstelle zwischen der Hardware – dazu gehören beispielsweise Rechner, Telefone, Netzwerke - und dem Anwender, der sie benutzt.

## Bedeutung der Informations- und Kommunikationstechnologien (IKT) für Entwicklungsländer

Grundsätzlich stellen die IKT Informationen und Dienstleistungen Personen effizient bereit. Folglich könnten die IKT die sozialen und ökonomischen Entwicklungen nicht nur in den Industrieländern, sondern auch in den Entwicklungsländern fördern.

Es ist daher dringend erforderlich, dass alle zur Verfügung stehenden Technologien zur Optimierung der Landwirtschaft genutzt werden – denn die Vorteile der Entwicklungsländer liegen im Bereich der Landwirtschaft.

In diesem Bereich ermöglicht IKT den Austausch und die Massenverteilung von vorteilhaften Landwirtschaftstechniken, Schädlings-Bekämpfung und den Export von landwirtschaftlichen Produkten. Auch in den Bereichen Umweltschutz, Gesundheitswesen, Ausbildung und Forschung führt der Einsatz der wichtigsten Schlüsseltechnologie zu Effizienzsteigerungen und Kostensenkungen.

Nach dem Bericht des Ausschusses für Entwicklung und Zusammenarbeit des Europäischen Parlaments wird mit dem fehlenden Zugang den Entwicklungsländern die Möglichkeit verwehrt, Nutzen aus der zunehmenden Globalisierung bei Handel, Investitionen und Austausch von Dienstleistungen zu ziehen, denn hierfür ist die Einführung von IKT Voraussetzung.

Der fehlende Zugang der Entwicklungsländer zu den IKT und das unzureichende Wissen darüber bringen eine Reihe spezifischer Gefahren mit sich:

- Die stimulierende Wirkung von IKT auf die wirtschaftliche und soziale Entwicklung des Landes kommt den Entwicklungsländern nicht zugute.
- Ausschluss vom privaten Sektor des industrialisierten Teils der Welt, da der Einsatz von moderner Technologie in den Entwicklungsländern nicht gewährleistet werden kann.
- Entwicklungsländer erhalten keinen Zugang zu dem Wissen, welches der öffentliche und der private Sektor über das Internet allgemein verfügbar machen.
- Überalterung des Bildungs- und Ausbildungssystems, da dieses nicht in der Lage ist, die neuen Technologien zu nutzen.

Um ein weiteres Zurückbleiben der Entwicklungsländer zu verhindern, ist eine Entwicklungszusammenarbeit zur Förderung von IKT anzustreben, um damit die sogenannte „Digitale Kluft“ zu vermindern. Die „Digitale Kluft“ definiert den Unterschied bei den Zugangsmöglichkeiten zu den IKT sowie bei den Möglichkeiten, einen eigenen Beitrag zur Information zu leisten, oder Nutzen aus der entwicklungsfördernden Wirkung der IKT zu ziehen. Die Trennlinie zieht sich zwischen den Industrieländern und den Entwicklungsländern hindurch.

„Open Source“ Software bietet nun mit seiner freien Verfügbarkeit und seinem neuartigen Modell der Software-Entwicklung eine Chance zur Verminderung der „Digitalen Kluft“.

Die kommerzielle Software hingegen kann mit gutem Support und Ausbildungsdienstleistungen aufwarten. Hierbei kann der Käufer nicht Anwender und Entwickler gleichzeitig sein – wie bei „Open Source“ Software –, sondern nur Anwender, denn der Quellcode ist nicht öffentlich zugänglich.

Für einen sinnvollen Einsatz von IKT in wenig technisierten Ländern ist die Verfügbarkeit von ausländischen Fachkräften wichtig, da eigene, qualifizierte Personen im Inland fehlen. Diese Arbeit beschäftigt sich mit der Frage, welcher Software-Typ die Entwicklungszusammenarbeit zur Verminderung der „Digitalen Kluft“ stärker begünstigt und damit einen Beitrag zur Förderung von sozialen und wirtschaftlichen Entwicklungen leistet.

## **2. Die digitale Kluft**

In den USA wurde die „Digitale Kluft“ („Digital Divide“) erstmals untersucht.

Diese Bezeichnung erschien in einer Zeit, als die Informations- und Kommunikationstechnologien (IKT) immer stärkeren Einfluss auf die wirtschaftlichen Prozesse ausübte. Im Informationszeitalter entstand dann ein Bruch innerhalb der Bevölkerung, der zwischen denjenigen, die mit dem Internet vernetzt, und denjenigen, die nicht vernetzt waren, hindurchging.

Es entstand hierdurch eine Trennlinie zwischen verschiedenen Ethnien.

Der unterschiedliche Soziale Status, der Ausbildungsstand und die geographische Lage der amerikanischen Bevölkerung trennten weiter die Vernetzten von den Unvernetzten. Aufgrund des fehlenden Zugriffs auf das Netz und somit auf Information sah die amerikanische Bevölkerung eine Ungleichheit, die es zu lösen galt. Die „Digitale Kluft“ ist nicht nur als Symptom der bestehenden sozialen und wirtschaftlichen Ungleichheiten zu sehen, sondern auch als Ursache für einen dramatischen Anstieg dieser Ungleichheiten.

Die „Digitale Kluft“ bedeutet im allgemeinen, dass ein Teil der Bevölkerung aufgrund fehlender technischer Mittel wie Computer oder Internetzugang und dementsprechend mangelnder IKT-Kompetenz von den Vorteilen und Karriereaussichten der Wissens- und Informationsgesellschaft ausgeschlossen ist, womöglich den Anschluss verliert und in Armut bleibt.

Im Zusammenhang mit der Globalisierung und der weltweiten Vernetzung durch IKT entstand die globale „Digitale Kluft“. Diese trennt die gesamte Weltbevölkerung entlang der Trennlinie zwischen den Industrieländern und Entwicklungsländern. Auch innerhalb eines Landes kann eine weitere digitale Kluft zwischen Stadt- und Landregion entstehen.

In der am G8-Wirtschaftsgipfel in Okinawa verabschiedete Charta über die globale Informationsgesellschaft, wurde festgelegt, dass die globale „Digitale Kluft“ überwunden werden muss (G8-Wirtschaftsgipfel in Okinawa (2000)):

„Das Potenzial der IKT trägt zur weltwirtschaftlichen Dynamik bei. Die damit verbundenen wirtschaftlichen Chancen müssen allen Menschen weltweit zugänglich sein, es darf weder innerhalb einzelner Länder, noch zwischen Industrie- und Entwicklungsländern eine digitale Kluft geben.“

## 2.1 Das Ausmaß

Die drei folgenden Beispiele zeigen, dass die „Digitale Kluft“ erheblich ist:

- Ein Drittel der Weltbevölkerung hat noch nie telefoniert.
- In London gibt es mehr „Internet Hosts“ als in ganz Afrika.
- 90% der Internet „Hosts“ befinden sich in Ländern mit höherem Einkommen, wobei ihre Bevölkerung nur 16% der Weltbevölkerung ausmacht.

Die unterschiedliche Zahl von Internet-Nutzern aus Industrieländern und Entwicklungsländern verdeutlicht die ungleiche Nutzung des Internets. In Industrieländern ist das Internet vielen zugänglich, während in Entwicklungsländern das Internet nur einer kleinen Gruppe der Bevölkerung zur Verfügung steht. Über die letzten Jahre hinweg betrachtet, nimmt die „Digitale“ Kluft sogar zu.

### Beispiele:

#### **Geografische Verteilung der Länder mit direktem Internetanschluss**

<u>Region</u>	<u>Zahl angeschlossener Länder</u>
Afrika	8 von 52
Antarktis	1 von 1
Asien	23 von 47
Australien/Ozeanien	4 von 14
Europa	37 von 40
Nord-/Mittelamerika	13 von 21
Lateinamerika	9 von 14

#### **Telefondichte in ausgewählten Staaten**

<u>Staat</u>	<u>Einwohner pro Telefon</u>
Afghanistan	681
Angola	248
Bangladesch	531
Bolivien	53
Kambodscha	keine privaten Telefone
Äthiopien	keine privaten Telefone
Deutschland	2
Japan	2
USA	2

## Unterschiede zwischen Nord und Süd

<u>Bereich</u>	<u>Verhältnis</u> <u>Nord/Süd</u>
Telefone	100: 6
Radios	100:17
Fernsehen	100:12
Informationstechno- logie	100:12

## in % die weltweit verfügbaren Personal-Computer

<u>Personal-Computer in %</u>	<u>Länder</u>
37.92	USA und Kanada
28.77	Europa
8.86	Japan
5.97	Lateinamerika
4.99	China
1.2	Indien
8.49	sonstiges Asien
1.52	Afrika

## 2.2 Dimensionen der „Digitalen Kluft“

Fehlt es in den Entwicklungsländern an Heim-Rechnern, weil viele Menschen sich diese nicht leisten können, so kann kein nützliches Netz aufgebaut werden. Dazu gehören auch die für den Betrieb von Heim-Rechnern notwendige Software, sowie die physische Vernetzung zu anderen Rechnern oder zu einem bestehenden Netz;

kurzum - Hardware und Software - beides erst ermöglicht das Nutzen von Informations- und Kommunikationstechnologien. Besteht einmal die Infrastruktur der Vernetzung von Rechnern, so braucht es sicherlich noch das Wissen mit dieser Hardware und Software umgehen zu können.

Die „Digitale Kluft“ existiert auf verschiedenen Ebenen. Lisa Seron nennt **drei Dimensionen** der „Digitalen Kluft“:

1. Infrastruktur
2. „Know-how“
3. „Inhalt“

Die erste Dimension charakterisiert die physische Verfügbarkeit von Informations- und Kommunikationstechnologien. Dazu gehören alle notwendigen Geräte, welche die gesamte Infrastruktur aufbauen.

Neben der zur Verfügung gestellten Infrastruktur ist auch das „Know-how“ der Menschen, welche die Infrastruktur benutzen, von Bedeutung. Diese zweite Dimension der „Digitalen Kluft“ ist eine entscheidende Barriere gegen einen breitenwirksamen Einsatz von Informations- und Kommunikationstechnologien.

Nach Servon existiert eine weitere Dimension der „Digitalen Kluft“. Mit „Inhalt“ charakterisiert Servon die frei zur Verfügung gestellte Information, zum Beispiel die Texte, welche auf einer Internet-Seite publiziert sind. Nur sprachlich verständliche und brauchbare Web-Seiten, können den Menschen in Entwicklungsländern von Nutzen sein.

### **3. Analyse von „Open Source“ Software im Zusammenhang mit der Entwicklungszusammenarbeit**

In diesem Kapitel wird die Frage behandelt, welche besonderen Eigenschaften „Open Source“ Software aufweist und welche Vorteile für Entwicklungsländer daraus gezogen werden können.

#### **3.1 Besondere Eigenschaften von „Open Source“ Software**

„Open Source“ Software ist Software deren Lizenz erlaubt, den Quellcode egal für welchen Zweck, laufen zu lassen, zu studieren, zu modifizieren oder weiterzugeben. Im Gegensatz zur proprietären Software eröffnen diese besonderen Eigenschaften neue Wege der Software-Entwicklung. Dabei sind zwei markante Eigenschaften von „Open Source“ Software hervorzuheben:

Zum Einen ist das die freie Verfügbarkeit und zum anderen ist das die modulare Architektur. Der bei der Entwicklung entstehende neuartige Aufbau der Software wird in den folgenden Abschnitten erläutert, um sie dann mit der proprietären Software hinsichtlich ihrer Stabilität und Erweiterbarkeit zu vergleichen.

##### **3.1.1 Freie Verfügbarkeit von „Open Source“ Software**

Software ist eine Ansammlung von Befehlen oder eine Reihe von komplexen Formeln. Oft geschieht es, dass Formeln falsch implementiert werden, so kann es sein, dass die Software falsche Resultate liefert, oder dass der laufende Prozess sogar „abstürzt“. Die Stabilität ist ein Kriterium für die Zuverlässigkeit der Software. Sie ist ein Maß dafür, wie fehlerhaft eine Software ist. Die Qualität der Software hängt maßgeblich davon ab. Eine vereinfachte Methode zur Messung der Stabilität von Software besteht darin, die Zahl der Abstürze oder ihre Ausfalldauer zu messen.

Im Folgenden wird erörtert, warum die frei verfügbare Software durchschnittlich mehr Stabilität aufweist. Ein gewisser Raymond umschreibt das „Open Source“ Softwareentwicklungs-Modell als „Bazaar style“. Solche Software ist aus verschiedenartigen

Programmstücken aufgebaut, die meistens von unterschiedlichen Personen entwickelt wurden. Eine populäre Erklärung für die Namensgebung des bekannten „Apache“ Web-Servers deutet darauf hin. Der Name dieser „Open Source“ Serversoftware steht für „A PATCHY Server“ („Ein zusammengeflickter Server“). Ein solcher Softwareentwicklungsprozess findet vor allem in einem Internet-Portal statt - wie beispielsweise „SourceForge.net“, in welchem sich Software-Entwickler weltweit zusammenfinden. Dieses verweist zu vielen verschiedenen Software-Projekten und hält frei zugängliche Software bereit. Der Entwickler selbst nimmt sich was er brauchen kann und fügt vorhandene und neue Komponenten einfach zusammen. Dies bringt den Vorteil, falls gewisse

verwendete Komponenten nicht mehr passen oder neue, bessere Komponenten gefunden werden, sie kurzerhand mit besseren Codefragmenten aus der Software-Sammlung ausgetauscht werden können. Sind aber gewünschte Komponenten nicht auffindbar, so implementiert der Entwickler selber den notwendigen Teil. Er kann aber auch auf dem Internet-Portal, ein neues Projekt anmelden und freiwillige Mitarbeiter anwerben. Diese helfen dann bei der programm-technischen Umsetzung der notwendigen Komponenten.

Die freie Verfügbarkeit ermöglicht nun, dass die Software von vielen, verschiedenen Interessenten begutachtet wird. Die fast unbegrenzte Zahl von Programmierern aus aller Welt überprüft die „Open Source“ Software und korrigiert vorhandene Fehler. Die frei verfügbaren Quellcodes werden dadurch laufend verbessert. Im Falle von kommerzieller Software jedoch ist der Quellcode nur einer relativ kleinen Zahl von Software-Entwicklern zugänglich. Diese sind dann für die Implementierung des Software-Pakets größtenteils selber verantwortlich. Der Quellcode wird nach der Implementierung verschlossen, um sie schließlich selber vermarkten zu können. Ab diesem Zeitpunkt wird dann nur noch teilweise der Quellcode von den Software-Entwicklern verbessert.

### **3.1.2 Modulare Architektur**

Die schnell fortschreitenden Entwicklungen in der Informations- und Kommunikationstechnologie zwingen die Benutzer sich an die neue Technologie anzupassen, um nicht ins wirtschaftliche Hintertreffen zu geraten. Dazu gehört auch stets die Verwendung aktueller Software. Es ist selten nötig die gesamte Software zu erneuern, denn meistens bedürfen nur Teile davon einer Verbesserung.

Die mit unterschiedlichen Softwareteilen zusammengesetzte frei verfügbare Software, bietet nach dem „Bazaar Style“ von Raymond mehr Modularität. Dem Programmierer wird dadurch ermöglicht Teile des Quellcodes einfach auszuwechseln oder neue Softwareteile mit schon bestehenden Teillösungen zusammenfügen.

Im Gegensatz dazu kann im Falle von proprietärer Software der Benutzer nicht selbstständig die Software einfach umeine gewünschte Funktionalität erweitern, denn einerseits ist deren Quellcode nicht frei verfügbar und andererseits sind die einzelnen Softwareteile stark vernetzt.

Daraus folgt, dass mit der Eigenschaft der modularen Architektur Software schneller, einfacher und überhaupt erweiterbar ist. Als gelungenes Beispiel für eine große „Open Source“ Projekt-Entwicklung ist hier das Betriebssystem „Red Hat“ Linux 7.1 erwähnt, das über 30 Millionen physikalische Quellcodelinien besitzt und das von vielen Programmierern und unterschiedlichen Softwaremodulen entwickelt wurde.

## **3.2 Ökonomische Argumente für einen Einsatz von „Open Source“ Software**

### **3.2.1 Investition in das Humankapital des Entwicklungslandes**

Den Menschen in Entwicklungsländern fehlt es an Geld, um Heim-Rechner zu kaufen. Der Nutzen eines Heim-Rechners ist für eine einzige Familie zu klein, im Vergleich zu den Kosten, die sie dafür aufwenden muss. Stehen in einem Entwicklungsland einmal Rechner zur Verfügung, so werden diese auch Software brauchen müssen. Die Rechner brauchen einerseits Betriebssysteme, welche die Aufgabe haben, dafür zu sorgen, dass die

einzusetzende Software die Hardware des Rechners benutzen kann und andererseits braucht es Anwendungs-Software - wie zum Beispiel Textverarbeitungs- oder Tabellenkalkulationsprogramme. Nachdem der Anwender bestimmt hat, welche Programme er braucht und welche Software er dafür auf dem Rechner installieren muss, so kann er sich nun für eine proprietäre Software oder „Open Source“ Software entscheiden.

In diesem Abschnitt werden neben den Kosten für den Erwerb auch die Kosten für den Unterhalt beider Software betrachtet. Gesucht wird dabei nach den Gründen eines Kostenunterschieds und nach dem Nutzen, den die Entwicklungsländer dabei gewinnen können, wenn sie sich für eines der beiden Software-Typen entscheiden.

Nach Kapitel 3.1 ist die frei verfügbare Software mindestens so stabil wie die kommerzielle Software, und sie ist durch ihre modulare Architektur auch leicht erweiterbar. Um die Kosten nun zu vergleichen, wird in diesem Zusammenhang das weitverbreitete Betriebssystem Windows XP von Microsoft dem frei verfügbaren Betriebssystem Linux gegenübergestellt.

Für den Erwerb von Windows XP, bezahlt der Anwender 200 US-Dollar (Stand 2001). Wird hingegen die Linux-Alternative gewählt, so bezahlt der Anwender lediglich ein paar US-Dollar. Entweder bezieht er die frei verfügbare Software über eine Verbindung im Internet, oder er bekommt sie kostenlos auf einer CD-Rom. Folglich bezahlt er nur für die Transaktion der Software, deren Kosten sich auf einige Dollar beschränken – je nach Menge der Daten, die heruntergeladen werden.

Im Folgenden wird davon ausgegangen, dass für Anwendungen der kommerziellen Software immer eine entsprechende frei verfügbare Variante existiert. Auch für Textverarbeitungsanwendungen wie Microsoft Word, Präsentationsanwendungen wie Microsoft Powerpoint oder Tabellenkalkulationen wie das Microsoft Excel, welche unter anderen zusammen in Office Paketen enthalten sind, gibt es „Open Source“ Alternativen. Die „OpenOffice“ Anwendungs-Software, das Pendant zum Microsoft Office Paket, kann auch kostenlos vom Internet heruntergeladen werden.

Es wurde ein mexikanisches Schul-Projekt vorgestellt, welches das Ziel hatte, etwa 140 000 Schulhäuser mit Software zu beliefern. In einer Studie wurde dann schließlich geschätzt, dass für die Installation von Windows 98, Microsoft Office und einem Server mit Windows NT, mindestens 885 US-Dollar für einen Computer Labor investiert werden muss. Daraus folgt, dass für die Belieferung aller Computer Labore sich die Kosten auf 124 Millionen US-Dollar belaufen würden. Wäre ein frei verfügbares Betriebssystem wie Linux verwendet worden, so würden sich die Kosten auf 50 US-Dollar beschränken – nämlich Kosten für eine Sammlung von CD-Roms und einer Anleitung von Red Hat Linux, welche unter der GPL vervielfältigt und an weitere Schulhäuser weitergegeben werden könnte.

Im folgenden Abschnitt werden die Gründe für diesen Kostenunterschied erörtert und die volkswirtschaftlichen Folgen bei der Verwendung von „Open Source“ Software zum Teil aus eigenen Quellen aufgezeigt.

- Ein erster Grund ist die Motivation von „Open Source“ Software Programmierern. Arbeiten freiwillige Programmierer an deren Weiterentwicklung, so erhöht sich der Software Preis nicht, weil die Software-Entwicklung keine Kosten verursachen.

- Als zweiter Grund kann die Vernachlässigung der Installationskosten von „Open Source“ Software genannt werden.

Da „Open Source“ Software auf vielen verschiedenen Plattformen ausführbar ist, ändert sich die für die Installation notwendige Konfiguration immer. Auch die vielen verschiedenen Versionen oder Distributionen von „Open Source“ Software-Anwendungen erschweren die Installation, weil bei jeder anderen Version andere

Einstellungen getätigt werden müssen. Die Installation von proprietärer Software ist hingegen einfacher. Software-Anwendungen, die vom gleichen Unternehmen produziert werden, können problemlos miteinander verknüpft werden. Betrachten wir die Software-Produkte von Microsoft, so sehen wir, dass diese sich erfahrungsgemäss praktisch von selbst installieren. Die Kosten dieser Vereinfachung bei der Software Installation sind im Preis der proprietären Software enthalten; ganz im Gegensatz zu der „OpenSource“ Software Variante.

- Als dritter Grund kann die uneingeschränkte Verteilbarkeit von „Open Source“ Software aufgezählt werden. Nach der von der „Free Software Foundation“ erklärten GPL oder nach der Definition von „Open Source“ kann Software unter GPL und „Open Source“ Software kostenlos vervielfältigt und uneingeschränkt an andere weitergegeben werden. Demzufolge kann die einmal erworbene Software auf CD-Rom, mehrfach kopiert und an jedem Rechner in einer Universität oder Schule im Entwicklungsland installiert werden, ohne dabei weitere Kosten zu verursachen, abgesehen von den Installationskosten. Will eine Schule an jedem Arbeitsplatz eine proprietäre Software installieren, so müsste sie für jeden Rechner die gleiche Lizenz kaufen. So werden die Kosten für die Verwendung von proprietärer Software um die Zahl der zu installierenden Rechner vervielfacht.

- Als weiterer Grund kann der unterschiedliche Unterhalt der Software erwähnt werden. Tauchen Probleme bei der Installation der Software auf, oder bestehen Fragen bezüglich der Optimierung von mehreren Software-Programmen, die verwendet werden, so steht dem Anwender ein Supportdienst vom Software-Hersteller zur Verfügung. Betrachten wir die Support Web-Seite von Microsoft, so kann sich jeder Erwerber entweder direkt über email zu einem Spezialisten wenden, oder indirekt aus dem Archiv mit bereits behandelten Fällen einen Ratschlag holen. Microsoft bietet im weiteren auch kostenpflichtige Support-Verträge an. Diese erweitern die Support-Leistungen. Betrachten wir die von „OpenSource“ Software-Entwicklern zur Verfügung gestellte Support Web-Seite, so kann der Anwender auch hier auf Support-Leistungen zählen. Tipps erhält er von archivierten, bereits behandelten Fällen, die er selber nachliest, oder er bekommt diese direkt von anderen Anwendern, die seine auf der Web-Seite publizierte Frage gelesen haben. Der angebotene, kostenlose Support auf der Web-Seite wird ausschließlich von anderen Anwendern gewährleistet. Im weiteren sind auch hier kostenpflichtige Support-Verträge für erweiterte Support-Leistungen möglich. Falls Spezialisten für den Support eingesetzt werden, so verursachen diese laufende Kosten, welche die Preise der proprietären Software steigen lassen. Im Gegensatz dazu ist der Support von „Open Source“ vor allem eine von anderen Anwendern ausgeübte, freiwillige Tätigkeit. Im Grunde genommen, helfen sich die Anwender gegenseitig beim Support der Software ohne dass dabei weitere Kosten entstehen.

Die oben erwähnten Gründe erklären zwar den Kostenunterschied zwischen der kommerziellen und frei verfügbaren Software in Industrieländern; ob nun Entwicklungsländer von dem kostengünstigeren Software-Typ profitieren können, wird im nachfolgenden Abschnitt behandelt.

Die zweite Dimension der „Digitalen Kluft“ das „Know-how“ ist eine entscheidende Barriere gegen einen breitenwirksamen Einsatz von Informations- und Kommunikationstechnologien (IKT). Man erklärte weiter, dass IKT infolge ihres Querschnittscharakters rasch in allen wirtschaftlichen Prozessen an Einfluss gewinnt. Die Nachrekrutierung von IKT-Kräften braucht Zeit, schon daher, weil der Aufgabenbereich sehr viel komplexer ist. Es braucht dementsprechend mehr qualifizierte Fachleute, welche die Infrastruktur aufbauen und unterhalten können, wenn der Einsatz von IKT in den Entwicklungsländern vergrößert

werden soll. Investitionen in Ausbildung im IKT-Sektor können dem Mangel an qualifizierten Fachkräften entgegenwirken.

Falls die Unternehmer in den Entwicklungsländern die Chance wahrnehmen könnten, „Open Source“ Software zu verwenden, so könnten sie Geld sparen und es anderweitig investieren. Bei der Installation von Software braucht es jedoch im Falle von „Open Source“ Software Personen, welche die Fähigkeit besitzen, die Software auch richtig zu konfigurieren oder im Allgemeinen einen Software-Support zu leisten. In den Entwicklungsländern fehlt es aber an qualifizierten Fachkräften, die Support-Leistungen erbringen können. Um von der frei verfügbaren Software dennoch profitieren zu können, müssten deshalb Personen im Entwicklungsland mit solchen Fähigkeiten ausgebildet und das notwendige Wissen müsste ihnen vermittelt werden. Ein Support mit rekrutierten Fachkräften aus dem Entwicklungsland selbst, würde nicht nur nach Arbeitsplätze schaffen, sondern auch verhindern, dass durch einen Einsatz von ausländischen Fachkräften, Geld ins Ausland fließt und damit nicht im Entwicklungsland selbst investiert wird, sondern in Sachkapital von Industrieländern. Im Folgenden bietet sich der offengelegte Quellcode geradezu dafür an, die Technologie, die im Quellcode steckt, zu erlernen und anzuwenden.

Investiert nun das Entwicklungsland in Ausbildung von qualifizierten Fachkräften, so wird sich längerfristig das Humankapital des Landes steigern.

Durch die Verwendung von „Open Source“ Software und die damit verbundene Investition in Ausbildung von qualifizierten Fachkräften ist das vorhandene Humankapital im Entwicklungsland höher als ohne diese – oder mit kommerzieller Software. Das Humankapital verschiebt sich schließlich.

### **3.2.2 Software-Anpassungen ans lokale Umfeld**

1991 begann die schnelle Entwicklung eines Betriebssystems GNU/Linux durch die Veröffentlichung des ersten Quellcodes. Linus Torvalds, der Schöpfer des Betriebssystemkerns von Linux, stellte seinen Quellcode unter die GPL der „Free Software Foundation“. Seitdem wurde der Quellcode aus einer gemischten Gruppe von computer-begeisterten Entwicklern ständig verbessert und weiterentwickelt. Was die Personen bewog freiwillig an solchen Projekten mitzumachen wird nachfolgend erklärt. Auch das Ansehen ist ein wichtiger Motivationsfaktor.

Sind viele Entwickler miteinander vernetzt, und jeder kann dabei ohne Einschränkung die Arbeit des anderen sehen, so ist das Ansehen größer, wenn eine gute Arbeit zur Verfügung gestellt wird. Je mehr Software-Entwickler den erarbeiteten Quellcode mit anderen teilen und je mehr die Arbeit unter den Programmierern geteilt wird, desto höher das Ansehen, das sich einer verdienen kann. Es wird auch die Motivation gesteigert, wenn Software-Entwickler Fehler im Quellcode finden oder einfach ihre Implementierung des Softwareprogramms beenden können. Viele Programmierer investieren viel Zeit und Geld nur um ihr Softwareprogramm zum Laufen zu bringen. Welchen Nutzen die Menschen in den Entwicklungsländern davon ziehen können, wird im nachfolgenden Abschnitt erörtert.

Da es für Menschen in Entwicklungsländern schwierig ist, eine Ausbildung zu bekommen oder sogar eine computer-spezifische Ausbildung, wird ihr Umgang mit importierter Software stark erschwert. Aufgrund zum Teil ganz anderer Schriftsätze in Entwicklungsländern braucht es an ihre Sprache angepasste Software-Anwendungen, welche die Menschen lesen und verstehen können, um sie schließlich verwenden zu können. Software-Anpassungen sind deshalb für den nachhaltigen Erfolg von Informations- und Kommunikationstechnologie in den Entwicklungsländern wichtig.

Anpassungsfähigkeit internationaler Informations- und Kommunikationstechnologien (IKT) an lokale Gegebenheiten stellt einen kritischen Faktor für den nachhaltigen Erfolg der Informations- und Kommunikationstechnologien in Entwicklungsländern dar. Die IKT umfasst jene Erfindungen und Produkte, die es Menschen ermöglichen, sich mit Hilfe von Computern zu informieren.

Die Entwicklungsländer profitieren von Informations- und Kommunikationstechnologien dann, wenn sie Software an ihre Umgebung anpassen können.

Die weltweite Verteilung von Internet Hosts zeigt, wer Web-Seiten anbietet und in welcher Sprache sie verfasst sind. Mehr als 75 Prozent der Seiten, die im Internet zugänglich sind, kommen aus Nord-Amerika.

Der Nutzen ist, dass die Menschen in den Entwicklungsländern eigene Anpassungen an lokale Gegebenheiten durchführen können, und dass angepasste, importierte Software immer zur Verfügung steht, selbst dann, wenn ausländische Software-Hersteller angepasste Software aus Rentabilitätsgründen nicht anbieten können.

Da zum Beispiel die Software-Anwendungen meist für Unternehmungen und Personen in Industrieländern produziert werden, sind die Texte in den Anwendungen in einer, in Industrieländer gesprochenen Sprache verfasst. Sprachanpassungen importierter Software werden von Industrieländern nicht durchgeführt, weil Software-Märkte in Entwicklungsländern zu klein für große Software-Anbieter sind.

Falls Betriebe oder Personen von angepasster Software trotzdem profitieren möchten, so ermöglichen Open Source Software-Entwickler solche Anpassungen, weil für die lokale Bevölkerung keine Anpassungskosten anfallen. Dadurch ist es möglich Software-Anwendungen in entlegenen Dörfern in ihrer eigenen Sprache zu verfassen.

Im Grunde genommen können alle Software-Anpassungen von freiwilligen Software-Entwicklern umgesetzt werden, auch jene die für ausländische Software-Unternehmen nicht rentabel sind.

## **4. Fazit**

Die freie Verfügbarkeit von „Open Source“ Software vereinfacht nicht nur deren Zugriff, sondern ihre Quellcodes können auch ohne Einschränkung eingesetzt und verteilt werden, da deren Lizenzvereinbarungen diese nicht verbieten.

Der Einsatz der frei verfügbaren Software ermöglicht Investitionen in das Humankapital zu leisten. Moderne Technologie, wie die kommerzielle Software (Microsoft), muss aus dem Ausland importiert werden, damit der technologische Fortschritt im Entwicklungslande garantiert wird. Für die frei verfügbare Software müssen jedoch nur die Transaktionskosten aufgewendet werden. Es kann deshalb, statt in das Sachkapital von ausländischen Software-Firmen zu investieren, direkt in die Ausbildung von qualifizierten inländischen Fachkräften investiert werden.

Das Humankapital des Landes erhöht sich dadurch und damit auch die volkswirtschaftliche Produktion.

Die modulare Architektur vereinfacht die Modifizierbarkeit und Erweiterbarkeit von frei verfügbarer Software. Dadurch können leichter Software-Anpassungen an lokale Gegebenheiten durchgeführt werden.

Der offengelegte Quellcode und die starke Motivation der Software-Entwickler ermöglichen zusätzlich, dass zum Beispiel Sprachanpassungen der Software auch dann durchgeführt werden, wenn keine Software-Firma sie gewinnbringend umsetzen kann.

Der Einsatz von „Open Source“ Software entschärft die berechtigte Kritik soweit, dass zum einen sie an lokale Gegebenheiten angepasst werden können, und zum anderen das Entwicklungsland die Möglichkeit bekommt auch als Software-Produzent im globalen Markt aufzutreten, weil die Technologie im offengelegten Quellcode erlernbar ist.

Neben den Investitionen in Schulung von qualifizierten Fachkräften und damitin „Know-how“, braucht es aber auch Investitionen in Hardware Ressourcen, und diese sind nur dann gerechtfertigt, wenn die darauf verwendete Software für die Befriedigung der Grundbedürfnisse der Menschen in Entwicklungsländern eingesetzt wird. Es braucht Entwicklungshilfe für den Aufbau der Infrastruktur.

## **5. Quellenangaben**

<http://www.europarl.eu.int/meetdocs/committees/deve/20010409/428813de.doc>

<http://www.gnu.org/philosophy/free-sw.html>

[http://www.rainer-gerling.de/aktuell/open\\_bt.html](http://www.rainer-gerling.de/aktuell/open_bt.html)

[http://www.auswaertiges-amt.de/www/de/aussenpolitik/friedenspolitik/g8/okinawa/index\\_html](http://www.auswaertiges-amt.de/www/de/aussenpolitik/friedenspolitik/g8/okinawa/index_html)

<http://www.abipur.de/hausaufgaben/alt/files/pdf/288.pdf>

<http://www.opensource.org/advocacy/fuzz-revisited.pdf>

<http://www.opensource.org/docs/definition.php>

<http://www.netaction.org/articles/freesoft.html>

<http://www.heise.de/tp/deutsch/inhalt/te/8408/1.html>

<http://www.dietmar-schmitt.de/essays/Entwicklung/PlaedoyerFuerOpenSource.html>

<http://www.input-consulting.com/download/hungernde-internet.pdf>

<http://www.gnu.org/gnu/thegnuproject.html>

[http://www.dwheeler.com/oss\\_fs\\_why.html#conclusions](http://www.dwheeler.com/oss_fs_why.html#conclusions)

[http://www.home.uni-osnabrueck.de/uafemann/Internet\\_Und\\_Dritte\\_Welt/inet-3w/inet-3w.htm](http://www.home.uni-osnabrueck.de/uafemann/Internet_Und_Dritte_Welt/inet-3w/inet-3w.htm)