

Open Source in der Hochschulausbildung

Prof. Dr. T. Romeyke

28. April 2009

Inhaltsverzeichnis

Abkürzungsverzeichnis	2
1 Einleitung	2
2 Anforderungen an die Ausbildung im Bereich der Wirtschaftsinformatik	2
2.1 Praktische Relevanz der Ausbildungsinhalte	2
2.2 Verbindung von Forschung und Lehre	3
3 Mögliche Auswirkungen durch Nutzung von quelloffenen Produkten in der Lehre	3
3.1 Wirtschaftliche Aspekte	3
3.2 Didaktische Aspekte	4
3.3 Organisatorische Vorteile	4
3.4 Strukturpolitische Vorteile	4
4 Unterschiedliche Projektstrukturen	5
5 Kriterien zur Auswahl geeigneter Systeme	6
5.1 Dokumentation	6
5.2 Qualität	6
5.3 Verbreitung	6
6 Konkrete Beispiele	7
6.1 Betriebssysteme	7
6.2 Enterprise Ressource Management	8
6.3 Dokumentenmanagement	8
6.4 Customer Relationship Management	8
6.5 Mathematik	9
6.6 Freie Software - freie Daten: Ein Projekt mit OpenStreetmap	9
7 Umsetzungsmöglichkeiten	9
7.1 Download-Link	10
7.2 Eigene Live-CD	10
7.3 Virtuelle Maschine	10
7.4 USB-Stick	11
8 Fazit	12

Abkürzungsverzeichnis

OSS	Open Source Software
DMS	Dokumentenmanagementsystem
CRM	Customer Relationship Management
TCO	Total Cost Of Ownership

1 Einleitung

Der Einsatz von Open Source Software (OSS) ist heute in vielen Unternehmen betrieblicher Alltag. Dieser Artikel untersucht, unter welchen Aspekten die Nutzung von OSS in der Hochschullehre betrachtet werden kann und welche Möglichkeiten sich eröffnen.

Die Ausbildung an einer Hochschule orientiert sich an verschiedenen Aspekten. So sollen einerseits die besten, innovativsten oder auch wirtschaftlichsten Methoden im jeweiligen Fachgebiet vermittelt werden - nach welcher Skala hier auch immer gemessen werden kann. Andererseits ist in vielen Fällen auch eine gewisse Orientierung an den in der Praxis verwendeten Methoden sinnvoll - die unmittelbare Einsetzbarkeit der Absolventen in einem Unternehmen ist insbesondere für viele zum Bachelor-Abschluß führende Studiengänge an Fachhochschulen und Universitäten ein wesentliches Ziel in der Ausbildung.

2 Anforderungen an die Ausbildung im Bereich der Wirtschaftsinformatik

2.1 Praktische Relevanz der Ausbildungsinhalte

Im Bereich der zum Bachelor führenden Studiengänge mit Anteilen aus der Wirtschaftsinformatik (zum Beispiel Wirtschaftsingenieurwesen und Betriebswirtschaftslehre) sind bei der Erarbeitung von Curricula eine Reihe von Entscheidungen zu treffen. In einer Dimension betrifft dies die gleiche Art von Entscheidungen, die in anderen Disziplinen auch zu fällen sind: Über welche Grundlagen muß ein Absolvent verfügen können, um Problemstellungen aus der Praxis klassifizieren und lösen zu können; welche Methoden und Verfahren muß er anwenden können, damit er in seinem Fachgebiet kompetent arbeiten kann? Doch in Fächern mit IT-Bezug kommt noch eine weitere Entscheidungsdimension hinzu: Wenn denn entschieden ist, dass eine Kompetenz in der Planung und Modellierung von Applikationen beispielsweise zur rechnerbasierten Unterstützung der Geschäftsprozessabwicklung notwendig ist, so folgt daraus einerseits die Vermittlung von theoretischen Zusammenhängen, andererseits ist es sowohl für die Lernmotivation als auch den Lernerfolg überaus sinnvoll, die praktische Anwendung des erarbeiteten Wissens zu üben. In aller Regel gibt es eine Fülle von Produkten durchaus unterschiedlicher Qualität, die dafür in Frage kommen. Da eine parallele Nutzung verschiedener Systeme wegen des damit erforderlichen Lern- und Betriebsaufwands für jedes einzelne System meist nicht in Frage kommt, muß eine Auswahl erfolgen: Neben kommerziellen Systemen, die zumeist noch als proprietäre Produkte vermarktet werden, stehen heute vielfach auch qualitativ vergleichbare Systeme auf Basis von OSS zur Wahl.¹ Sieht man von vergleichsweise wenigen Ausnahmen wie beispielsweise dem Apache-Webserver² ab, sind die OSS-Alternativen im Unternehmenseinsatz allerdings zur Zeit nur unterdurchschnittlich vertreten.

¹Vgl. Widmer und Bähler (2006), S. 165ff.

²Der aktuelle Anteil der durch den Apache-Webserver betriebenen Sites beträgt ca. 46% (Vgl. Netcraft).

2.2 Verbindung von Forschung und Lehre

Verfolgt eine Ausbildung das Ziel, Absolventen *praxisfertig* zu verabschieden, spricht dieser Punkt für den Einsatz eines etablierten - und damit zumeist proprietären - Produktes in der Ausbildung. Und auch eine Forderung nach der Ausbildung in der *leading edge* widerspricht nicht unbedingt dieser Wahl: Gerade die Hersteller der großen, weitverbreiteten Produkte unterhalten vielfach gemeinsame Projekte mit Hochschulen, die einen schnellen Wissenstransfer von der Hochschule in die Praxis fördern sollen.

3 Mögliche Auswirkungen durch Nutzung von quelloffenen Produkten in der Lehre

Trotz der im vorhergehenden Abschnitt dargestellten Überlegungen kann sich jedoch ein verstärkter Einsatz von OSS in der Lehre als sehr sinnvoll herausstellen. In den folgenden Abschnitten werden einzelne Aspekte untersucht, die durch die Wahl der in der Lehre eingesetzten Produkte beeinflusst werden können.

3.1 Wirtschaftliche Aspekte

Ein nicht unbedeutender Aspekt ist heute die finanzielle Seite einer derartigen Auswahl: Hochschulen müssen mit sehr knappen Ressourcen auskommen. Auf den ersten Blick ist der Wegfall von Lizenzkosten bei Open-Source-Systemen deshalb ein Vorteil. Doch abgesehen davon, dass auch an Hochschulen ein Aufwand für den Betrieb von Applikationen eingeplant werden muß, ist inzwischen allgemein bekannt, dass im Rahmen einer Betrachtung der Total Cost Of Ownership (TCO) in aller Regel die Lizenzkosten nur eine untergeordnete Rolle spielen.³ Hinzu kommt, dass für die Anwendung in der Hochschullehre für praktisch alle relevanten Produkte spezielle Lizenzmodelle angeboten werden, die deren Kosten in die Nähe der Open-Source-Alternativen bringen. Im Bereich der Software besteht für einen Hersteller kein physisches Distributionsproblem mehr: Jedes Produkt lässt sich ohne Qualitätsverlust beliebig häufig kopieren und nutzen. Wenn ein Hersteller also eine rechtliche Regelung mit einer Hochschule als ausreichend erachtet, entstehen ihm durch die Nutzung der Software im Rahmen der Ausbildung keine zusätzlichen Kosten.⁴ Als Auswahlkriterium taugt dieser Punkt demnach nur in besonderen Fällen.

Der vorgenannte finanzielle Aspekt hat jedoch eine weitere Facette, die berücksichtigt werden kann: Es darf vermutet werden, dass die Unternehmen eher nicht aus Großzügigkeit oder bildungspolitischer Verantwortung auf mögliche Lizenzeinnahmen verzichten. Vielmehr bauen sie darauf, dass die mit ihrem Produkt vertrauten Absolventen im späteren Berufsleben im Zweifelsfalle bei diesem Produkt bleiben werden und den Aufwand für das Erlernen eines Konkurrenzproduktes nach Möglichkeit vermeiden wollen. Und selbst, wenn die Umstände des Arbeitsplatzes zum Lernen eines weiteren Systems zwingen, so neigt der Mensch dazu, ein neues Produkt in jedem Detail mit dem Gewohnten zu vergleichen und andere Abläufe als umständlich und weniger gut einzuschätzen. Etwas überspitzt lassen sich die durch den Hersteller verbilligt abgegebenen Lizenzen deshalb dem Bereich *Vertriebsunterstützung* zuordnen.

Damit stellt sich die Frage, ob eine Hochschule eine wie auch immer geartete Vertriebsunterstützung für kommerzielle Unternehmen leisten sollte. Neu ist diese Frage nicht, auch in anderen Disziplinen ist eine Zusammenarbeit mit Unternehmen und die vergünstigte Bereitstellung von

³Vgl. Laudon u. a. (2006), S. 301f.

⁴Die - mitunter auf beiden Seiten - nicht unerheblichen Kosten der Lizenzverwaltung seien an dieser Stelle einmal außer acht gelassen.

beispielsweise Produktionsmaschinen für Unterrichtszwecke üblich und auch sinnvoll. Neu ist allerdings mit der Informatik die Verfügbarkeit von zur Lehre geeigneten Produkten ohne (oder mit nur gering ausgeprägtem) kommerziellen Herstellerinteresse. Möglich wird damit tatsächlich eine akademische Ausbildung, die ohne kommerziell gefärbte, werbende Effekte die im theoretischen Teil erarbeiteten Kenntnisse an einem im Markt vertretenen Produkt spiegeln und untersuchen kann.

Ein wesentlicher Kostenblock kann durch den Support von in der Lehre eingesetzten Anwendungen ausgelöst werden. Das gängige Modell des Anwendungssupports zielt nur auf die hochschuleigenen Systeme, da die proprietären Produkte i.d.R. nur dort installiert werden dürfen. Damit ist der Supportbedarf zum einen geringer und zum anderen besser planbar als im Falle von vielen Einzelinstallationen auf Geräten von Studierenden. Tendenziell wird demnach im Falle des Einsatzes von OSS-Produkten ein höherer Supportbedarf die Folge sein. Doch auch die Art des Supports wird sich ändern: Nicht unbedingt der Experte für eine einzelne Applikation ist hier gefragt, sondern eher jemand mit Überblick über die Gesamtinstallation und deren Querwirkungen. Noch komplexer kann es im Falle plattformabhängiger Anwendungen werden, wenn plötzlich Supportanfragen zu Installationen auf verschiedenen Betriebssystemen kommen.

3.2 Didaktische Aspekte

Ein positiver Nebeneffekt ist bei der Wahl von OSS darin zu sehen, dass die verwendete Software in den weitaus meisten Fällen auf Grund der verwendeten Lizenz und der Verfügbarkeit des Sourcecodes an spezielle Anforderungen angepaßt werden kann. Damit lassen sich tiefergehende Untersuchungen aus dem theoretischen Vorlesungsteil in die Praxis übertragen. Eingeschränkt ist die tatsächliche Anwendbarkeit dieser Option allerdings in den meisten Fällen dadurch, dass je nach Studiengang und Vorbildung der Studierenden die Einarbeitung in komplexe Softwaresysteme mit erheblichem Zeitaufwand verbunden ist. In den Fällen jedoch, in denen trotz dieser Einschränkungen zum Beispiel die Initiierung eines Addons oder eines speziellen Moduls möglich ist, kann dies zu einer erheblichen Motivierung der Studierenden beitragen - schließlich kann ein erfolgreich umgesetztes Projekt unmittelbar in die Community zurückgegeben werden. Doch auch in programmierfernen Studiengängen lassen sich auf diese Weise Motivationen erzeugen: Durch die Entwicklung einer (bisher nicht verfügbaren) spezifischen Dokumentation zu einem in einer Vorlesung untersuchten Aspekt einer Anwendung gewinnt die Beschäftigung mit dem Thema für viele Studierende erheblich an Attraktivität.

3.3 Organisatorische Vorteile

Andere konkrete Vorteile aus dem OSS-Lizenzmodell sind ganz praktischer Natur: Konventionell lizenzierte Software ist üblicherweise auf den hochschuleigenen Geräten in Hörsälen oder Labors installiert. Damit steht sie vorrangig im Unterricht und - je nach Betriebskonzept - auch für Lerngruppen zur Verfügung. Eine durchaus sinnvolle Nutzung auf den Laptops der Studierenden scheitert in den meisten Fällen an der Lizenzfrage, das Zeitfenster für die mögliche Nutzung im Rahmen des selbstorganisierten Lernens wird damit unnötig klein gehalten.

3.4 Strukturpolitische Vorteile

In bedeutenden Softwarebereichen liegen die zentralen Wertschöpfungsanteile in wenigen Lokationen - dass das nicht immer die USA sind, zeigt das Beispiel SAP sehr deutlich. Da im OSS-Bereich i.d.R. keine Lizenzkosten zu zahlen sind, führt ein größerer Anteil der Zahlungen, die auch ein OSS-Software einsetzendes Unternehmen leisten muß (Stichwort TCO) tendenziell

zu einer Erhöhung der regionalen Wertschöpfung. Insbesondere die Software-Anpassung und -Wartung sowie der Service durch unterschiedliche regionale Unternehmen wird positiv beeinflusst, da die zentrale Steuerung und Lizenzierung von Servicepartnern durch den Hersteller hier nicht erfolgt. Ob das auf Dauer so bleibt und nicht auch im Laufe der Zeit größere Unternehmen einen Konzentrationsprozess bewirken, ist natürlich nicht sicher. Gerade die Möglichkeit, durch einen Hochschul-Spin-Off die Gründung von in diesem Bereich tätigen Unternehmen zu fördern, stellt allerdings einen positiven Wert an sich dar.

Zusammengefaßt bleibt festzustellen, dass in der gegenwärtigen Marktsituation einige Punkte dafür sprechen, qualitativ vergleichbare OSS-Produkte in der Hochschullehre den kommerziellen Lösungen vorzuziehen. Der Unterricht kann bereichert werden, in einzelnen Fällen können sogar Unterrichtsergebnisse ins Produkt einfließen.

4 Unterschiedliche Projektstrukturen

Betrachtet man die Struktur von OSS-Projekten, so lässt sich die Motivation der Teilnehmer in unterschiedliche Gruppen unterteilen.⁵ In einer Gruppe findet man die häufig mit dem Begriff OSS verbundenen losen Verbünde von freien Entwicklern, die im wesentlichen in ihrer Freizeit am Projekt arbeiten. Eine zweite Gruppe unterscheidet sich in dieser Hinsicht von der ersten: Die Projekte werden schwerpunktmäßig von einem Unternehmen gesponsort, das selbst durchaus kommerzielle Interessen verfolgt.⁶ Dieses Konzept ist in der *harten* OSS-Szene nicht unumstritten, allerdings hat es sich als Erfolgsmodell gezeigt, wenn es um die schnelle und gezielte Entwicklung von produktiv einsetzbarer Software geht. Redhat und Canonical (Ubuntu) beispielsweise haben das im Bereich der Betriebssysteme mit ihren Linux-Distributionen gezeigt. Welche Gründe sind hier zu nennen?

Klare Zielsetzungen Ein kommerziell agierendes Unternehmen wird seinen steuernden Einfluß so einsetzen, dass das Produkt möglichst schnell den von ihm als bedeutsam erachteten Qualitätskriterien entsprechen wird. Wichtige Funktionalitäten werden implementiert, auch wenn kein Entwickler aus eigener Initiative in diesem Bereich tätig werden würde.

Dokumentation Das größte Problem vieler OSS-Projekte ist die mangelhafte Dokumentation - Systementwickler mit Interesse an der Erstellung und Pflege einer aktuellen und hilfreichen Dokumentation haben Seltenheitswert. Ein Unternehmen als Treiber eines OSS-Projektes hat die Möglichkeit, geeignete Mitarbeiterkapazität für die Erstellung der gewünschten Dokumentation einzuplanen. Dies ist im unmittelbaren Unternehmensinteresse, da dies die Einsetzbarkeit der Software bei neuen Anwendern fördert und damit Servicepotential schafft.

Entwicklungskapazität Viele Feierabendprojekte werden mit großem Enthusiasmus gestartet, versanden dann aber recht schnell, wenn die täglichen Mühen der Softwarepflege die Adrenalinsschübe der ersten Versionen gedämpft haben. Eine Sortierung zum Beispiel der Sourceforge-Projekte⁷ nach Datum des letzten Beitrags zeigt das sehr deutlich. Auch hier haben Unternehmen bessere Möglichkeiten: Eine gut kalkulierte Planung und die Auswahl der richtigen Mitarbeiter sind eine gute Grundlage einer längerfristigen Projektlebenszeit.

Spätestens an dieser Stelle stellt sich natürlich erneut die Frage nach der Kommerzialisierung der Lehre: Wenn oben die kostenfreien Lizenzen proprietärer Produkte für Ausbildungszwecke wegen der damit verbundenen *Prägung* der Studierenden abgelehnt werden, so werden nun dennoch

⁵Vgl. Luthiger (2004)

⁶Vgl. Knoblich (2006)

⁷Sourceforge.net betreibt eine Plattform für eine Vielzahl von freien Softwareprojekten.

die wesentlich von einem kommerziell agierenden Unternehmen vorangetriebenen OSS-Projekte wieder in die Hochschule geholt - ein Widerspruch? Tatsächlich entspricht dies nicht dem oben vorgezeichneten idealen Bild: Allerdings muß die Ausbildung aus Qualitätsgründen Applikationen nutzen, die zumindest als *Enterprise-Ready* angesehen werden können. Im Bereich der Wirtschaftswissenschaften sind das in vielen - wenn auch nicht allen - Fällen eben Softwareprodukte, die nach dem eben beschriebenen Modell entstanden und weiter entwickelt werden.

Dabei sollen diese Überlegungen nicht als Plädoyer für eine kommerziell gesponserte Projektentwicklung verstanden werden. Es gibt genügend Beispiele, dass auch in der konventionellen OSS-Organisationsstruktur erfolgreiche Produkte entwickelt werden können: Das Debian-Projekt etwa hat bewiesen, dass ein solides Betriebssystem in einer derartigen Struktur entstehen kann. Aber an diesem Beispiel wird ebenfalls deutlich, wie ein kommerziell unterstütztes Projekt schnell Ziele erreichen kann: Ubuntu als Debian-Derivat ist auf dem Desktop erheblich verbreiteter, weil es aus Sicht der Anwender offensichtlich besser dem dort bestehenden Bedarf entspricht.

5 Kriterien zur Auswahl geeigneter Systeme

An ein im Rahmen der Ausbildung sinnvoll einsetzbares System müssen eine Reihe von Anforderungen gestellt werden. Dabei ist es unerheblich, ob dieses System zum proprietären oder zum OSS-Bereich gehört.

5.1 Dokumentation

Ohne eine gute Dokumentation sind auch OSS-Produkte nicht sinnvoll in der Lehre einsetzbar. Da keine Dokumentation perfekt ist, gilt es in jedem Einzelfall zu prüfen, ob auf Basis der vorliegenden Dokumentation das geplante Lehrprojekt realisierbar erscheint. Durch eine geschickt gewählte Abfolge von Projekten lässt sich allerdings etwas tricksen: So kann beispielsweise in einem Semester ein eher systemtechnisches Projekt genutzt werden, um Dokumentationsdefizite zu reduzieren. In einem weiteren Projekt können dann Studierende anderer Fachrichtungen die erarbeitete Dokumentation nutzen.

5.2 Qualität

Eine Einsetzbarkeit von OSS im Unternehmen ist nur dann gegeben, wenn die Produkte zumindest eine zum proprietären Bereich vergleichbare Qualität haben. Dazu gehören zumindest folgende Punkte:

Systemstabilität: Die Applikation muß so stabil laufen, dass Abbrüche zumindest nicht häufiger vorkommen als in marktgängigen Produkten.

Funktionalität: Der Funktionsumfang muß in etwa dem entsprechen, der auch in marktgängigen Produkten abgedeckt wird. Zudem soll auch die Korrektheit der implementierten Algorithmen vergleichbar sein.

Die Implementierung von Standard-Verfahren im jeweiligen Bereich ist dafür zwingend erforderlich.

5.3 Verbreitung

Die Frage, ob eher ein sehr verbreitetes Produkt oder eine Alternative in der Lehre verwendet werden soll, wurde oben schon in einem anderen Aspekt diskutiert. Abschließend lässt sich dieser Punkt nicht klären, einige weitere Positionen für bzw. gegen den Einsatz proprietärer Produkte sollen hier noch kurz angerissen werden:

- Pro:**
- Der Absolvent kann in den Unternehmen, die dieses Produkt einsetzen, unmittelbar produktiv arbeiten, der notwendige Schulungsaufwand ist zumindest reduziert.
 - Eine bekannte Marke im Vorlesungstitel erhöht potentiell mehr das Interesse von Studierenden als ein eher unbekannter Produktname.
- Con:**
- Wenn man annimmt, dass einer der positiven Aspekte des Integrierens von Hochschulabsolventen in ein Unternehmen das Hinzukommen neuer Gedanken und Ideen ist, wäre es von Vorteil, wenn während des Studiums nicht nur die ohnehin schon in der Praxis eingesetzten Konzepte und Produkte vermittelt wurden.
 - Ein erfahrener Nutzer eines guten OSS-Systems sollte in der Lage sein, sich schnell in die entsprechende Anwendung eines anderen Herstellers einzuarbeiten. Dabei kann es ihm sogar helfen, wenn an dem OSS-System verschiedene Realisierungsmöglichkeiten einzelner Prozesse durchgespielt wurden. Zu klären wäre, wieviel Schulungsaufwand denn notwendig ist, um einem an einer generischen Lösung ausgebildeten Absolventen die fehlenden Kenntnisse zu vermitteln.

6 Konkrete Beispiele

Der folgende Überblick über eine Reihe von Softwarebereichen zeigt auf, welche OSS-Produkte nach Erfahrungen des Autors in der Lehre verwendbar sind. Diese Liste ist naturgemäß subjektiv und unterliegt zudem einer hohen Dynamik - sie kann demnach nur Anhaltspunkte für eigene Überlegungen liefern.

Eine vollständige Liste von für die Lehre geeigneten Projekten zusammenzustellen ist praktisch nicht möglich - zu groß ist einerseits die Zahl in Frage kommender Themen, andererseits die Zahl der verfügbaren Projekte - ein Blick auf Sourceforge und vergleichbare Seiten verweist zu praktisch jedem Thema auf passende Projekte. Für einen Unternehmenseinsatz sind jedoch die allermeisten nicht geeignet. Doch lassen sich relativ kompakt einige Beispiele zusammenstellen, die die Auswahl auch in anderen Themengebieten vereinfachen. Die jeweils genannten Projektnamen sind dabei nur als prominente Beispiele zu sehen, weitere Alternativen sind zumeist verfügbar, hier aber nicht genannt. Alle Beispiele beziehen sich auf den wirtschaftswissenschaftlichen Bereich - für andere Disziplinen mag die Aufstellung nicht unbedingt übertragbar sein.

6.1 Betriebssysteme

Das Reizthema verliert in letzter Zeit aus verschiedenen Gründen an Bedeutung: Einerseits lehnen sich die Desktop-Alternativen der weitverbreiteten Windows-Familie (im wesentlichen *Apple* und diverse Linux-Distributionen) sehr an die den Anwendern bekannte Oberfläche an (zumindest sind entsprechende Linux-Konfigurationen verfügbar), ein Lernaufwand beim Wechsel zwischen diesen Systemen ist häufig auch nicht größer als der beim Wechsel beispielsweise von *Windows-XP* zu *Windows-Vista*. Andererseits ist heute davon auszugehen, dass praktisch alle Studierenden im häuslichen Bereich Zugriff auf einen Rechner haben - dies ist zur Zeit meist ein Windows-System. Der Umgang mit diesem System muß also nicht mehr unbedingt geübt werden, zumal gerade in der Wirtschaftsinformatik das Betriebssystem ein *Mittel zum Zweck* ist. Wenn dann ein wenig verbreitetes, aber durchaus mit Vorteilen versehenes Betriebssystem wie Linux behandelt wird, dann kann das den Studierenden neue Erkenntnisse bringen und Denkanstöße liefern. Und schließlich steigt inzwischen auch in den Unternehmen und der öffentlichen Verwaltung der Anteil der Arbeitsplätze, für die ein Windows nicht mehr als technisch notwendig angesehen wird - sicherlich nicht zuletzt aufgrund von Ideen, die auch von Hochschulabsolventen in die Unternehmen und Behörden getragen wurden.

6.2 Enterprise Ressource Management

Das zur Zeit sicherlich bedeutsamste Segment der ERP-Systeme wird im kommerziellen Bereich von *SAP* angeführt. Doch gerade im Feld der kleinen und mittleren Unternehmen (KMU) wird vielfach der Einsatz von *SAP R3* aus unterschiedlichen Gründen kritisch gesehen. Neben den etablierten Systemen der kommerziellen Mitbewerber kommt in diesem Bereich auch der Einsatz von OSS-Systemen in Frage. Besonders wichtig ist in diesem Umfeld allerdings die Lokalisierung eines OSS-Projekts. Dabei geht es weniger um die Übersetzung von Dialogen und Handbüchern, vielmehr stellt die Anpassung an nationale Normen und Gesetzeslage das eigentliche Problem dar. Systeme wie *OpenBravo*⁸ oder *Compiere*⁹ sind von der Leistungsfähigkeit her für einen KMU-Einsatz durchaus geeignet. In manchen Fällen scheitert eine Evaluierung dann aber an unklarer Berücksichtigung zum Beispiel steuerlicher Belange. Auch die in manchen Fällen möglicherweise notwendige Zertifizierung kann eine - zu hohe - Hürde darstellen. Im Bereich der Lehre sind diese Systeme daher durchaus einsetzbar, allerdings scheint gerade bei Studierenden der Wirtschaftswissenschaften die Nachfrage nach Informationen speziell zum Produkt *SAP R3* besonders ausgeprägt zu sein.

6.3 Dokumentenmanagement

Ein Feld mit extremen Wachstumsraten ist das Dokumentenmanagement. Nachdem der jahrelang von der IT versprochene Umstieg auf das papierlose Büro eher ins Gegenteil verkehrt wurde, sind Systeme gefragt, die den Umgang mit allen Arten von Dokumenten effizient unterstützen. Für die Hochschullehre bedeutet das, dass in den Bereichen Organisation und Geschäftsprozessmanagement das Dokumentenmanagement einen wichtigen Platz einnehmen muß.

Ein relativ weit verbreitetes Dokumentenmanagementsystem (DMS) im OSS-Bereich ist sicherlich *ALFRESCO*¹⁰ - ein insofern erfolgreiches OSS-Projekt, da es zeigt, dass eine Verbindung von OSS und Kommerz durchaus möglich ist: Das Produkt ist in einer freien Version verfügbar und einsetzbar, *ALFRESCO-Software-Inc.* als Hauptentwickler bietet für zahlende Kunden allerdings eine mit Support erhältliche Version sowie zusätzliche Module und Services an. Gerade die angekündigte Zusammenarbeit zwischen *Canonical/Ubuntu* und *Alfresco* bietet die Chance, eine vergleichsweise einfach einzurichtende Alternative nutzen zu können.

Weniger verbreitet und auch weniger funktionsreich ist eine in Deutschland entwickelte Alternative: Mit *agorum core*¹¹ bietet das hinter dem OSS-Produkt stehende Unternehmen *Agorum* mit einer vergleichbaren Lizenzpolitik ein DMS an, das (bislang) im Unterschied zu *Alfresco* schnell und unkompliziert zu installieren und in Betrieb zu nehmen ist. Gerade für einen Einstieg in das Thema Dokumentenmanagement kann das in der Lehre von Vorteil sein.

6.4 Customer Relationship Management

Die mit dem Begriff Customer Relationship Management (CRM) benannten Systeme ermöglichen die effiziente Verwaltung von Kundenkontakten. Neben den marktbeherrschenden kommerziellen Produkten gibt es in diesem Bereich eine ganze Reihe von Systemen, die das Thema CRM mit unterschiedlichen Schwerpunkten, aber doch sehr professionell angehen. Vom Verbreitungsgrad her sind *SugarCRM*¹² und *vTiger*¹³ zu nennen, die gemeinsame Vorläufer haben. Beide Produkte sind weitgehend lokalisiert.

⁸Vgl. *OpenBravo*

⁹Vgl. *Compiere*

¹⁰Vgl. *Alfresco*

¹¹Vgl. *Agorum*

¹²Vgl. *SugarCRM*

¹³Vgl. *vTiger*

6.5 Mathematik

In einer Vielzahl von Studiengängen gehören zum Curriculum mathematische Grundlagen, die in den ersten Semestern vermittelt werden. Traditionell sind dies nicht die Fächer, die bei Studierenden Motivationsschübe auslösen. Es hat sich jedoch gezeigt, dass zumindest in Teilbereichen durch breiten Einsatz von Softwareunterstützung eine intensivere Beschäftigung mit den Problemstellungen bewirkt werden kann. Der Autor hat die Mathematiksoftware MAXIMA¹⁴ unterrichtsbegleitend eingesetzt, um den Studierenden einen praktischen Umgang mit mathematischen Zusammenhängen zu ermöglichen.

6.6 Freie Software - freie Daten: Ein Projekt mit OpenStreetmap

Ein weiteres Beispiel für die sinnvolle Verwendbarkeit von OSS auch ohne Programmierung stellen Projekte im Umfeld von OpenStreetmap¹⁵ dar. Zur Zeit gibt es noch genügend *weiße Flecken* in der OSM-Karte, so dass Studierende hochmotiviert Abschnitte erfassen und detailliert aufbereiten können. Derartige Projekte ermöglichen eine sehr anschauliche Erfahrung dessen, was OSS ausmacht. Die Auseinandersetzung mit Lizenzformen oder möglichen Geschäftsmodellen auf Basis dieser Daten bekommt dadurch eine eigene Dynamik, die bei einer rein theoretischen Betrachtung nicht realisierbar gewesen wäre.

7 Umsetzungsmöglichkeiten

Wie kann ein konkreter Einsatz von OSS-Produkten im beschriebenen Sinn ablaufen? Insbesondere der oben bereits angesprochene Punkt der Ausweitung von Support-Bedarfen erzwingt eine sorgfältige Planung. Der folgende Abschnitt soll einige alternative Vorgehensweisen aufzeigen - ohne Anspruch auf Vollständigkeit. Vorab scheint jedoch ein Blick auf die Ausgangslage angebracht, die nach Erfahrungen des Autors zumindest in den wirtschaftswissenschaftlichen Fächern gilt.

Hochschulausstattung: Die hochschuleigenen PC-Räume in den wirtschaftswissenschaftlichen Fachbereichen sind mehrheitlich mit Windows-Systemen ausgestattet. Solange die gewünschten Applikationen plattformneutral zur Verfügung stehen, ist eine Produkt-Installation für eine Vorlesungsperiode in aller Regel möglich. Allerdings ist damit die Übungszeit für die Studierenden meist weitgehend auf die Veranstaltungszeiten begrenzt: PC-Labore sind in vielen Fällen hoch ausgelastet. Hinzu kommt, dass es keine 1:1-Zuordnung zwischen Studierenden und PCs gibt. Damit muß in vielen Fällen die Applikationskonfiguration nach jedem neuen Login am PC auf den Ausgangszustand zurückgesetzt werden, ein Arbeiten über mehrere Sitzungen hinweg ist kaum möglich.

Serverbasierte Installationen: In vielen Fällen steht ein passender zentraler Server zur Verfügung, auf dem eine Installation der Applikation vorgenommen werden kann. Wenn alle Vorlesungsteilnehmer diese eine Installation nutzen müssen, können aus Stabilitätsgründen nur Benutzerkonten mit eingeschränkten Rechten an die Studierenden vergeben werden. Damit sind bestimmte Themen in praktischen Übungen in der Regel nicht mehr oder nur eingeschränkt möglich (beispielsweise Rechtekonzepte, Benutzerverwaltung). Im Idealfall steht eine so große Kapazität auf dem Zentralsystem zur Verfügung, dass jeder Studierende eine eigene Instanz der Applikation nutzen kann. Dies erfordert bei vielen Applikationen allerdings einen hohen Betreuungsaufwand, da bei komplexen Anwendungen keine einfachen Skripte zum Zurücksetzen auf beispielsweise den Vortagsstand möglich sind.

¹⁴Vgl. Maxima; eine passende grafische Oberfläche dazu: WxMaxima

¹⁵Vgl. OpenStreetmap

Private Geräte der Studierenden: Ein hoher Anteil der Studierenden verfügt heute über eigene Geräte (meist Notebooks). Etwa 85% bis 90% der Geräte laufen unter einer der Windows-Varianten, die zweite Gruppe stellen MAC OS-Systeme, die dritte verschiedene Linux-Varianten.

7.1 Download-Link

Die - für den Lehrenden - einfachste Variante ist natürlich die Angabe des Links zum ausgewählten Projekt. Verbunden mit Installationshinweisen wird das zu mehr oder weniger erfolgreichen, möglicherweise aber auch unterschiedlichen Installationsvarianten führen. Ein Teil der Unterrichtszeit wird für *Supportfragen* aufgewendet werden müssen. Vorteil dieser Variante ist jedoch, dass die Studierenden in ihrer gewohnten PC-Umgebung arbeiten können und keine Hochschulressourcen für Hardware aufgewendet werden müssen. Vorausgesetzt wird hier natürlich, dass es die gewünschte Software in einer auf Windows installierbaren Version gibt. Zumindest für Demo-Zwecke wird das jedoch häufig angeboten, auch wenn für den Produktivbetrieb eine Unix-Umgebung empfohlen wird.

7.2 Eigene Live-CD

Sollen mehrere Produkte verfügbar gemacht werden, kann es zweckmässig sein, diese in Form einer speziellen Live-CD vorzubereiten. In diesen Fällen hat der Lehrende dann größeren Einfluß auf die Installationsform: Die Live-CD wird in allen Fällen gleich reagieren, allerdings ist für ein vernünftiges Arbeiten eine CD häufig zu langsam. Da zudem keine Daten permanent gespeichert werden können, bietet sich eine leicht abgeänderte Vorgehensweise an: Die Live-CD wird als ISO-File auf den Laptop kopiert und dann mit Hilfe einer virtuellen Maschine in Betrieb genommen. Dann kann auch ein Filesystem für die Aufnahme von Datenänderungen mit eingebunden werden.

Die Erstellung einer individuellen CD ist je nach verwendeter Distribution vergleichsweise einfach: Unter Ubuntu beispielsweise ist das *Ubuntu Customization Kit* verfügbar, das den Prozess gut unterstützt. Beliebig aufwendig kann allerdings die Installationsvorbereitung der gewünschten Applikation werden.

7.3 Virtuelle Maschine

Steht auf den Notebooks der Studierenden ein Virtualisierungssystem zur Verfügung (mit *Virtual-Box*¹⁶ ist ein inzwischen zu Sun / Oracle gehörendes OSS-System erhältlich; Marktführer *VMWare*¹⁷ bietet ebenso wie *Microsoft* eine Freeware-Lösung an) und verfügt das physische Gerät über genügend RAM (mind. 1 GB hat sich als sinnvoll herausgestellt), so kann auch gleich eine virtuelle Maschine mit einer vom Lehrenden installierten Version der gewünschten Software bereitgestellt werden. Vorteil ist hier die identische Ausgangssituation für alle Studierenden. Je nach verwendetem Virtualisierungsprodukt ist allerdings die Inbetriebnahme einer heruntergeladenen virtuellen Instanz nicht immer unproblematisch - Virtual-Box muß dabei einen Vorsprung von VMWare noch aufarbeiten.

Die beiden Virtualisierungsvarianten bedeuten allerdings, dass als Betriebssystem der virtuellen Maschine Linux (oder ein anderes freies System) gewählt werden muß. *Reactos*¹⁸ als Plattform-Alternative zu Windows scheidet für diese Zwecke noch aus, damit müssen sich die Studierenden in vielen Fällen mit einem für sie neuen Betriebssystem auseinandersetzen. Wenn es jedoch eigentlich um die Applikationen gehen soll, muß entweder für den Einstieg mehr Zeit

¹⁶Vgl. VirtualBox

¹⁷Vgl. VMWare

¹⁸Reactos

eingepplant werden oder aber die Fähigkeit zur Nutzung von Linux als Serverbetriebssystem wird schon in den Vorseminestern erarbeitet.

7.4 USB-Stick

Eine vergleichsweise neue Kombination der in den Abschnitten 7.2 und 7.3 dargestellten Varianten erscheint zur Zeit am interessantesten: Auf einem 8GB-USB-Stick kann bequem ein komplettes Linux-System mit einem modernen Desktopmanager installiert werden, dazu dann die gewünschten Applikationen und eine etwa 1 bis 2 GB große beschreibbare Partition für Konfigurationsänderungen und Daten. Die Sticks können den Studierenden entweder für einzelne Vorlesungen auf Leihbasis zur Verfügung gestellt werden oder im Rahmen des Hochschulmarketings mit entsprechendem Logo versehen gleich an die Studenten verschenkt werden. Die Kosten für den Stick von zur Zeit etwa 10 bis 12 Euro lassen jedoch auch einen Kauf durch die Studierenden als zumutbar erscheinen.

Eine wesentliche Voraussetzung für das Funktionieren dieser Variante ist die Bootmöglichkeit vom USB-Stick, die bei den moderneren BIOS-Ausführungen allerdings die Regel ist.

Vorteile: Die Studierenden müssen auf ihren Notebooks keine eigene Software installieren, das Hauptbetriebssystem und dessen Anwendungen werden nicht berührt. Damit sinkt die (verständliche) Angst von Studierenden, die mühevoll eingerichtete Systemumgebung durch Installation eines *Semesterpaketes* zu verlieren.

Ein kleiner Test durch den Autor mit einer geringen Zahl von Studierenden und einem USB-Stick-Image ohne spezielle Applikationen im Sommersemester 2009 erreichte großes Interesse. Der USB-Stick enthielt nur ein Desktop-Linux¹⁹ mit den üblichen Anwendungen, vorlesungsspezifische Applikationen fehlten noch. Dennoch war die zu überwindende Schwelle zum Ausprobieren der Alternative offensichtlich weitaus niedriger als bei den in der Vergangenheit schon mehrfach angebotenen Live-CDs.

Ähnlich wie bei der Live-CD ist auch bei dieser Variante ein vergleichsweise einfacher Support möglich, da die Grundkonfiguration standardisiert ist und - im Extremfall - jederzeit wiederhergestellt werden kann.

Nachteile: Doch auch die USB-Stick-Variante hat (noch) Nachteile: Das Kopieren eines 8GB-Sticks dauert auf einem üblichen PC etwa 30 Minuten. Wenn Sticks für einen ganzen Jahrgang benötigt werden, wird für den Kopiervorgang spezielle Hardware erforderlich, die das Kopieren in höherem Maße parallelisiert ermöglicht. Dies ist jedoch ein eher technisch-organisatorisches Problem. USB-Hubs mit jeweils 7 Ports sind vergleichsweise günstig zu bekommen und können mit geeigneten Skripten eine USB-Ladestation mit Selbstbedienung ergeben.

Ein weiterer Nachteil ist die vergleichsweise geringe Zuverlässigkeit von USB-Sticks. Es erscheint sinnvoll, innerhalb des Linux-Systems eine einfache, lokal passende Backup-Komponente vorzubereiten, die via WLAN ein (differentielles) Backup auf den individuellen Speicherbereich der Studierenden auf dem Hochschulserver macht. So kann nach einem Austausch eines defekten Sticks schnell eine gesicherte Konfiguration samt Daten zurückgespielt werden.

Tendenziell wird eine Softwareverteilung auf Geräte bzw. Medien von Studierenden allerdings eine Steigerung des Supportbedarfs initiieren. Während ein Support für die individuellen Systeme von Privat-PCs noch nachvollziehbar abgelehnt werden kann, trifft dies

¹⁹Das Image wurde auf Basis der Knoppix-Distribution (Knoppix) erstellt.

auf eine durch die vom Dozenten ausgegebene Installation nicht mehr zu. Den zusätzlichen Supportbedarf kann der Hochschulbetrieb nicht ohne weiteres befriedigen.

Ein weiterer Nachteil betrifft die Dauerbaustelle Hardware-Treiber: Wenn Studierende eine besonders neue oder eine exotische Hardware nutzen, so ist der verwendete WLAN-Adapter häufig nicht Out-Of-The-Box verwendbar. In diesem Fall sind beliebig hohe Supportaufwände mit ungewissen Erfolgsaussichten möglich.

8 Fazit

Ziel dieser Übersicht soll nicht die Vertreibung aller kommerziellen Softwareprodukte aus den Hochschulen sein. In manchen Fällen jedoch scheint es bezüglich des hier beschriebenen Sachverhalts noch zu wenig Problembewusstsein zu geben: Eine komplette Ausbildung auf Produkten weniger oder gar eines Anbieters aufzubauen, erscheint vor diesem Hintergrund zumindest mittelfristig wenig durchdacht. Die Auseinandersetzung mit alternativen Bedienkonzepten, mit anderen Algorithmen und mit unterschiedlich interpretierten Abläufen kann helfen, die durchzuführende Arbeit besser zu verstehen und erfolgreicher zu hinterfragen. Und auch für Unternehmen kann es durchaus sinnvoll sein, Absolventen mit Erfahrungen abseits des aktuellen Mainstreams zu bekommen - ein Blick über den Tellerrand schadet nicht.

Literatur

- [Agorum] : *Agorum - Open Source DMS.* – URL <http://www.agorum.de>. – Zugriffsdatum: 23.4.2009
- [Alfresco] : *Alfresco - Enterprise Content Management.* – URL <http://www.alfresco.org>. – Zugriffsdatum: 23.4.2009
- [Compiere] : *Compiere - Open Source ERP.* – URL <http://www.compiere.com>. – Zugriffsdatum: 14.4.2009
- [Knoppix] : *Knoppix - Live-System.* – URL <http://www.knopper.net>. – Zugriffsdatum: 23.4.2009
- [Maxima] : *Maxima - Computer-Algebra-System.* – URL <http://maxima.sourceforge.net>. – Zugriffsdatum: 23.4.2009
- [Netcraft] : *Netcraft Webserver Survey Archives.* – URL http://news.netcraft.com/archives/web_server_survey.html. – Zugriffsdatum: 15.4.2009
- [OpenBravo] : *OpenBravo - Open Source ERP.* – URL <http://www.openbravo.org>. – Zugriffsdatum: 14.4.2009
- [OpenStreetmap] : *OpenStreetmap - Freie geografische Daten.* – URL <http://www.openstreetmap.org>. – Zugriffsdatum: 23.4.2009
- [Reactos] : *Reactos-Projekt (Betriebssystem).* – URL <http://www.reactos.org/de/index.html>. – Zugriffsdatum: 16.4.2009
- [SugarCRM] : *SugarCRM - Customer Relationship Management.* – URL <http://www.sugarforge.org/content/open-source>. – Zugriffsdatum: 23.4.2009

- [VirtualBox] : *VirtualBox - freie Virtualisierungslösung.* – URL <http://www.virtualbox.org>. – Zugriffsdatum: 23.4.2009
- [VMWare] : *VMWare - Virtualisierung.* – URL <http://www.vmware.com>. – Zugriffsdatum: 23.4.2009
- [vTiger] : *vTiger - Customer Relationship Management.* – URL <http://www.vtiger.org>. – Zugriffsdatum: 23.4.2009
- [WxMaxima] : *WxMaxima - GUI für Maxima.* – URL <http://wxmaxima.sourceforge.net>. – Zugriffsdatum: 23.4.2009
- [Knoblich 2006] KNOBLICH, Werner: *Erfolgreich mit Open Source - Das Red-Hat-Open-Source-Geschäftsmodell.* In: *Open Source Jahrbuch 2006*, Bernd Lutterbeck, Matthias Bärwolff, Robert A. Gehring (Hrsg.), 2006
- [Laudon u. a. 2006] LAUDON, Kenneth C. ; LAUDON, Jane P. ; SCHODER, Detlef: *Wirtschaftsinformatik.* Pearson Studium, 2006
- [Luthiger 2004] LUTHIGER, Benno: *Alles aus Spaß? Zur Motivation von Open-Source-Entwicklern.* In: *Open Source Jahrbuch 2004*, Robert A. Gehring, Bernd Lutterbeck (Hrsg.), 2004
- [Widmer und Bähler 2006] WIDMER, Ursula ; BÄHLER, Konrad: *Open-Source-Lizenzen - Wesentliche Punkte für Nutzer, Entwickler und Vertreiber.* In: *Open Source Jahrbuch 2006*, Bernd Lutterbeck, Matthias Bärwolff, Robert A. Gehring (Hrsg.), 2006