

**Andreas Hohenstein/
Karl Wilbers (Hrsg.)**

Handbuch E-Learning

**Expertenwissen aus
Wissenschaft und Praxis**

**Stand:
9. Erg.-Lfg. Juli 2004**

Mit freundlicher Genehmigung des Verlags »Deutscher Wirtschaftsdienst«

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Handbuch E-Learning : Expertenwissen aus Wissenschaft und Praxis /
Hrsg.: Andreas Hohenstein ; Karl Wilbers. – Köln : Dt. Wirtschaftsdienst

Grundwerk. . – 2002
ISBN 3-87156-298-X

Grundwerk einschließlich Ergänzungslieferung Juli 2004

Wolters Kluwer Deutschland GmbH

✉ Marienburger Straße 22, D-50968 Köln (Marienburg)

Redaktion E-Learning: Erwin Stickling

☎ Telefon: +49 (0) 221/9 37 63-18, Telefax: (02 21) 9 37 63-94

@ Internet: <http://www.dwd-verlag.de>, eMail: erwin.stickling@dwd-verlag.de

© 2004 Wolters Kluwer Deutschland GmbH, Neuwied, Köln, München

Deutscher Wirtschaftsdienst ist ein Imprint der Wolters Kluwer Deutschland GmbH

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzung, Mikroverfilmung und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung und Layout: Ute Weber GrafikDesign, Geretsried

Satz: Satz-Offizin Hümmer GmbH, Waldbüttelbrunn

Druck: Druckerei Plump, Rheinbreitbach

Open-Source-Software – E-Learning zum Nulltarif?

von Joachim von Kiedrowski (Qualitus GmbH)

Open-Source-Software · Learning-Management-Systeme · Quellcode · Schlagworte
E-Learning-Kosten · CampusSource · OKI · Sakai-Projekt

Open-Source-Software wird zunehmend auch im E-Learning-Bereich als Alternative zu kommerziellen Produkten wahrgenommen. Die Diskussion um den Einsatz von Open-Source-Software wird häufig auf den Preisvorteil durch die kostenlose Verfügbarkeit verkürzt. Dadurch entsteht der falsche Eindruck, dass mit Open-Source-Software E-Learning »zum Nulltarif« möglich wird. Der vorliegende Beitrag untersucht, mit welchen Implikationen der Open-Source-Einsatz im E-Learning-Bereich über den Preisvorteil hinaus verbunden ist. Es wird auf die Merkmale der Lizenzform Open Source sowie deren Vor- und Nachteile eingegangen. Als Beispiel für den Einsatz im E-Learning-Bereich werden Open-Source-Learning-Management-Systeme angeführt. Zum Abschluss werden wichtige Erfolgsfaktoren für den Einsatz von Open-Source-Software erörtert.

Überblick

1	Open-Source-Software als Alternative zu kommerzieller Software	2
2	Was bedeutet Open Source?	3
3	Einsatz Open-Source-Software im Bildungsbereich	5
4	Beispiele für Open-Source-Tools im E-Learning-Bereich	6
5	Vor- und Nachteile von Open-Source-Software	8
6	Potenziale von Open-Source-Software im E-Learning-Bereich	11
7	Erfolgsfaktoren für den Einsatz von Open-Source-Produkten	12
8	Fazit	13
	Literaturhinweise	13

1 Open-Source-Software als Alternative zu kommerzieller Software

Betriebssysteme Open-Source-Software, die den Anwendern unter bestimmten Lizenzbedingungen i. d. R. kostenlos zur Verfügung gestellt wird, erfreut sich in unterschiedlichsten professionellen Anwendungsbereichen der Wirtschaft, Industrie und Verwaltung sowie im akademischen Bereich steigender Beliebtheit. Zur zunehmenden Akzeptanz hat insbesondere der Erfolg des Open-Source-Betriebssystems LINUX beigetragen. Die Entscheidungen prominenter Kommunen wie z. B. der Stadt München, den gesamten EDV-Betrieb der öffentlichen Verwaltung mit über 14.000 Rechnern auf Open-Source-Software umzustellen, haben im vergangenen Jahr bundesweit für Aufsehen gesorgt (KERBUSK 2003). In den Nachbarländern der Bundesrepublik ist Open-Source-Software ebenfalls zu einer ernsthaften Alternative für kommerzielle Software geworden. So hat beispielsweise der Informatikrat der Schweizer Bundesverwaltung (IRB) eine bereits im Jahr 2002 angestoßene Strategie verabschiedet, nach der bei jeder Softwarebeschaffung Open-Source-Tools gleichberechtigt mit den bisherigen Produkten wie Windows, MS-Office, Unix und SAP geprüft werden (HEISE ONLINE 2004). Nach Auffassung des IRB trägt die Berücksichtigung von Open-Source-Software im Rahmen von Auswahlprozessen (in denen Open-Source-Software und kommerzielle Produkte anhand gleicher Kriterien evaluiert werden) zur Belebung des Wettbewerbs im Bereich der Software bei, was sich positiv auf die Qualität und Wirtschaftlichkeit der Systeme auswirkt (KLEMMER/ARPAGAU/ROSENTHAL 2004, S. 6).

Lernplattformen Auch im E-Learning-Bereich haben Open-Source-Produkte einige Aufmerksamkeit auf sich gezogen. So erreichte z. B. das Open-Source-Learning-Management-System ILIAS in einer von Baumgartner/Häfele/Maier-Häfele im Auftrag des österreichischen Bildungsministeriums durchgeführten Studie (BAUMGARTNER/HÄFELE/MAIER-HÄFELE 2002 bzw. Beitrag 5.4 in diesem Handbuch), in der mehr als 90 Produkte untersucht wurden, einen Rang unter den TOP 5 der untersuchten Plattformen. Aufgrund der guten Evaluationsergebnisse und unter Berücksichtigung des zusätzlichen Kriteriums »Preis/Leistung« wurde ILIAS im April 2003 in eine Empfehlung des Ministeriums zur Nutzung von Lernplattformen im Bildungsbereich aufgenommen (KRISTÖFL 2003, 2004).

Diese Entwicklung wurde sowohl von Anwendern als auch von den Anbietern kommerzieller E-Learning-Produkte aufmerksam verfolgt und diskutiert. So war das Thema »Standard-Software versus Open Source« Gegenstand einer Podiumsdiskussion auf dem 4. Learning Management Congress 2003 der imc AG (Imc 2003). Der amerikanische LMS-Hersteller WebCT legte seine Position zu Open-Source-Software auf der 3. WebCT European Conference 2004 dar (SALARI 2004). Im Online-Angebot des Handbuchs E-Learning auf dem Portal Global Learning (www.global-learning.de) wurden als Topic des Monats die Standpunkte verschiedener Experten diskutiert (GLOBAL LEARNING 2004).

In Zeiten schrumpfender IT-Budgets erscheint Open-Source-Software besonders attraktiv, weil sie kostenlos zu erwerben ist. Es können dadurch zwar erhebliche Einsparungen erzielt werden, es wäre allerdings falsch, die Diskussion über den Einsatz von Open-Source-Software auf diesen Einzelaspekt zu reduzieren. Vielmehr sollten die Merkmale sowie die Vor- und Nachteile von Open-Source-Software differenziert betrachtet werden.

2 Was bedeutet Open Source?

Die Grundidee von Open Source ist, dass diese Software »freie« Software ist. Dabei bezieht sich das Merkmal »frei« nicht in erster Linie auf die kostenlose Verfügbarkeit. Es bezeichnet vielmehr die Freiheit im Umgang mit der Software und ist zu verstehen wie »frei« in »freier Rede«, und nicht wie in »Freibier« (FREE SOFTWARE FOUNDATION 2002). Jeder Benutzer sollte die Software nicht nur verwenden dürfen, sondern auch die Freiheit haben, sie zu kopieren, zu vertreiben, zu studieren, zu verändern und zu verbessern. Diese Idee geht auf Richard Stallmann zurück, der 1983 das Betriebssystem GNU (GNU's Not Unix) als Derivat der Software UNIX unter der so genannten *GNU General Public License (GPL)* veröffentlichte (KHARITONIOUK/STEWIN 2004, S. 5). Die »Freiheit« der Software wird gemäß der GPL bestimmt durch die Freiheit,

Freie Software

- das Programm für jeden Zweck zu benutzen.
- zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann,
- Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann,
- das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann.

Der freie Zugang zum Quellcode der Software ist die Voraussetzung für die Ermöglichung der genannten Freiheiten (FREE SOFTWARE FOUNDATION 2002).

In der GPL wird explizit von »Freier Software« und nicht von Open-Source-Software gesprochen. Der Begriff »Open-Source-Software« geht auf die Ende der 90er Jahre von Bruce Perens und Eric S. Raymond gegründete *Open Source Initiative (OSI)* zurück. Anders als die Free Software Foundation, die eher einen idealistischen Ansatz vertritt, verfolgt OSI das Ziel, die Akzeptanz von Open-Source-Software in der Wirtschaft zu steigern, die bis dahin durch die restriktiven Bedingungen der GPL hinsichtlich der Freigabe von eigenen Modifikationen eingeschränkt wurde (KHARITONIOUK/STEWIN 2004, S. 6 f.). Die OSI entwarf aus diesem Grund eine eigene Open-Source-Definition, die die Grundlage für alle existierenden Lizenzen bildet (OPEN SOURCE INITIATIVE 2004).

Gemäß dieser Definition (Version 1.9) erfüllt Open-Source-Software folgende zehn Merkmale:

Merkmale

1. Freie Weiterverbreitung

Die Software kann uneingeschränkt weiterverbreitet werden. Sie darf auch als Komponente einer aggregierten Softwaresammlung, die aus verschiedenen Quellen stammen darf, verkauft oder weitergegeben werden. Das Verkaufen bzw. Weitergeben ist dann allerdings als Dienstleistung zu verstehen, da in der Lizenz für einen solchen Vertrieb keine Gebühren verlangt werden dürfen.

2. Quellcode

Das Programm enthält den Quellcode. Die Lizenz muss die Verteilung von Quell- und Binärcode erlauben. Der Quellcode muss in einer verständlichen Form verbreitet werden. Zwischenformen, wie die Ausgabe eines Präprozessors oder eines Übersetzers und willentliche »Umgestaltungen« des Quellcodes mit dem Ziel, ihn schwerer verständlich zu machen, sind nicht erlaubt.

3. Abgeleitete Arbeiten

Die Lizenz muss Modifikationen und Weiterentwicklungen und ihre Distribution zu denselben Lizenzbedingungen erlauben. Das heißt aber auch, dass eine Weiterverbreitung von abgeleiteten Arbeiten unter anderen Lizenzbedingungen nicht ausgeschlossen wird, sofern die Bedingungen der ursprünglichen Lizenz dies zulassen. Es ist demnach auch möglich, dass ein auf einem Open-Source-Programm nach diesen Regeln basierendes Werk nicht frei sein muss. Es gibt Open-Source-Lizenzen (wie z. B. die *Berkeley Software Distribution License*), die entsprechend ausformuliert wurden (vgl. KHARITONIOUK/STEWIN 2004, S. 8 f).

4. Unversehrtheit des Quellcodes von Autoren

Die Lizenz darf die Distribution des Quellcodes in modifizierter Form nur dann einschränken, wenn die Lizenz die Distribution von Patches zusammen mit dem Quellcode erlaubt, um das Programm zur »build time« zu modifizieren. Die Lizenz muss explizit die Distribution von Software, der der modifizierte Quellcode zugrunde liegt, erlauben. Die Lizenz kann fordern, dass Weiterentwicklungen einen anderen Namen oder eine andere Versionsnummer tragen. Dadurch sollen Originalcode und veränderter Code unterscheidbar bleiben und die Nutzer sollen erkennen können, welcher Autor für welchen Code verantwortlich ist.

5. Keine Diskriminierung von Personen oder Gruppen

Die Lizenz darf Personen oder Gruppen nicht diskriminieren. Niemand darf von der Weiterentwicklung ausgeschlossen werden.

6. Keine Einschränkungen von Einsatzmöglichkeiten der Software

Die Nutzung der Software darf durch die Lizenz nicht für bestimmte Anwendungsgebiete eingeschränkt werden. Dies gilt auch für die kommerzielle Nutzung oder die Anwendung in umstrittenen Einsatzgebieten, wie z. B. der Gen-Forschung.

7. Verbreitung der Lizenz

Die zu dem Programm gehörenden Rechte gelten für alle, an die es verbreitet wurde, ohne dass zusätzliche Vereinbarungen getroffen werden müssen.

8. Die Lizenz darf nicht für ein bestimmtes Produkt gelten

Die zu dem Programm gehörenden Rechte dürfen nicht davon abhängig sein, dass sie Teil einer speziellen Distribution sind. Wenn das Programm aus einer solchen Distribution herausgelöst wird, gelten weiterhin für alle dieselben Rechte am Programm, als wenn sie mit der originalen Distribution vergeben worden wären.

9. Die Lizenz darf andere Software nicht einschränken

Aufgrund dieses Gebots darf eine Open-Source-Lizenz beispielsweise nicht verlangen, dass alle Programme, die auf demselben Medium oder System genutzt werden, auch Open-Source-Software sein müssen.

10. Die Lizenz muss technologisch neutral sein

Es dürfen keine bestimmten Technologien oder Arten von Schnittstellen vorgeschrieben und andere damit ausgeschlossen werden.

Die obigen Ausführungen zeigen, dass Open-Source-Lizenzen einerseits zwar strengen Richtlinien unterliegen, andererseits aber auch über einen gewissen Gestaltungsspielraum z. B. hinsichtlich der weiteren Verbreitung von Modifikationen und Weiterentwicklungen verfügen. Das hängt von der jeweiligen Variante der Open-Source-Lizenz ab. Die ursprüngliche und radikalste Lizenzvariante, die General Public License, schließt solche Regelungen jedoch

aus. Vor der Auswahl einer Open-Source-Software sind die Lizenzbedingungen im Einzelfall zu prüfen und zu beurteilen.

3 Einsatz Open-Source-Software im Bildungsbereich

Open-Source-Software ist im Bildungsbereich international vor allem in den Hochschulen verbreitet. Dort werden Open-Source-E-Learning-Tools nicht nur eingesetzt, sondern auch entwickelt, modifiziert und weiterentwickelt. Dies ist wahrscheinlich darauf zurückzuführen, dass einerseits die Ursprünge der Open-Source-Bewegung im akademischen Bereich liegen und dort auch heute noch die Grundidee des Teilens und Verifizierens von Wissen in einer Community auf fruchtbaren Boden fällt. Es ist universitäre Tradition, Forschungsergebnisse zum Zwecke der Verifizierung zu veröffentlichen. Ähnlich verhält es sich auch mit der Entwicklung von Software. Die CampusSource-Initiative des Landes NRW, durch die die Verbreitung von Open-Source-Software unterstützt werden soll, hat eine Analogie zur Veröffentlichung der Ergebnisse von naturwissenschaftlichen Experimenten formuliert:

Hochschulen

»Die genaue Beschreibung eines Versuchsaufbaus entspricht dem Quellcode und der Installationsanweisung, analog zur Experimentdurchführung ist die Programmausführung bzw. die Software-Nutzung zu sehen und die Verifizierung der Versuchs- bzw. Experimentsergebnisse entspricht dem Austesten der Softwarefunktionalität. So wie im physikalischen, chemischen oder psychologischen Bereich Forschung und Weiterentwicklung bestehend auf Versuchen, die Dritte durchgeführt haben, erlaubt und gewünscht sind, z. B. durch Änderungen am Versuchsaufbau – werden Modifikationen des Quellcodes für Open-Source-Software erlaubt und gewünscht.« (CAMPUSOURCE 2000)

CampusSource

Ein weiterer, etwas unromantischerer Grund für die starke Verbreitung von Open-Source-Software im Hochschulbereich liegt in den knappen Budgets. Vielen Universitäten und Fachhochschulen fehlen schlichtweg die nicht unerheblichen finanziellen Mittel für die Anschaffung kommerzieller Software.

Knappe Budgets

Außerdem wurde von Bund und Ländern in den vergangenen Jahren die Entwicklung von Software im Rahmen von unzähligen E-Learning-Projekten massiv gefördert. Der Open-Source-Ansatz wird dabei von den Fördermittelgebern als Nachhaltigkeitsstrategie eingesetzt, wie die Zielsetzung der CampusSource-Initiative verdeutlicht:

Fördermittel

»Indem anderen Bildungseinrichtungen die Software zur Verfügung gestellt wird, können Dritte auf vorhandener Software aufbauen und neue Mittel fließen in die Weiterentwicklung existierender Systeme statt in eine Parallelentwicklung. Das Überleben der Software ist so auch in finanziell angespannten Zeiten gesichert. Weiterhin können die finanziellen Ressourcen des Landes und des Bundes sinnvoll und zielgerichtet eingesetzt werden. Nur indem die einzelnen Kräfte gebündelt werden, kann ausreichend Stärke erzielt werden, um im weltweiten Wettbewerb zu bestehen.« (CAMPUSOURCE 2000 a)

Aus Sicht der Mittelgeber ist dies eine sinnvolle Strategie. In Forschung und Entwicklung wird ein für den Standort wichtiger Know-how-Aufbau im Bereich des Software-Engineering unterstützt und gleichzeitig können Mittel für den Erwerb kommerzieller Software eingespart werden.

EU-Projekte Auch auf europäischer Ebene wird die Verbreitung von Open-Source-Software im Bildungsbereich gefördert. Die beiden EU-Projekte SIGOSSEE und JOIN entwickeln ein Portal zur Information und zum Austausch über den Einsatz von Open Source im Bildungsbereich (SIGOSSEE 2004).

Im Bereich der beruflichen Bildung ist die Verbreitung von Open-Source-Software unterschiedlich. Während Institutionen, die als Bildungsträger Maßnahmen beruflicher Bildung anbieten, bereits gute Erfahrungen gemacht haben, steigt die Akzeptanz von Open-Source-Software für E-Learning in Unternehmen nur langsam.

4 Beispiele für Open-Source-Tools im E-Learning-Bereich

Intransparenter Markt Wie auch bei kommerzieller Software ist der Markt für Open-Source-Software (den es im eigentlichen Sinne ja gar nicht gibt, weil die Produkte i. d. R. kostenlos abgegeben werden) sehr intransparent. Open-Source-Produkte sind für die gesamte Palette von E-Learning-Anwendungen verfügbar. Hierzu zählen Learning-Management-Systeme, Content-Management-Systeme, Autoren- und Assessmenttools sowie Kommunikations- und Kollaborationstools.

Als Beispiel sei hier die Auswahl von 15 Produkten genannt, die die Initiative CampusSource auf ihrer Homepage (www.campussource.de) vorstellt. Eine Suche über Open Source »Directories« wie z. B. Freshmeat (<http://freshmeat.net>) oder Sourceforge (<http://sourceforge.net>) hilft dagegen kaum weiter, da hier auch viele Entwicklungsprojekte aufgeführt sind, deren Software noch nicht über einen verlässlichen Produktstatus verfügt. Der Entwicklungsstand der Projekte ist dort jedoch nicht direkt erkennbar.

Produkt/ aktuelle Version	Ursprung	als OS verfügbar seit	Skalierbarkeit (*)	Standards	Url
ATutor 1.3.3	Adaptive Technology Resource Centre, University of Toronto	Dez. 2002	k. A.	IMS SCORM WCAG(**)	http://www.atutor.ca/
Claroline 1.4.2	Université catholique de Louvain (Institut de Pédagogie universitaire et des multimédias)	März 2002	k. A.	SCORM (enthalten im 1.5. Release Candidate)	http:// www.claroline.net
ILIAS 3.0.0	Universität zu Köln	Sept. 2000	5.000 User (Erfahrungswerte für Version 2)	LOM SCORM	http://www.ilias.de
Moodle 1.2.1	Curtin University of Technology	Aug. 2002	k. A.	keine	http://moodle.org
.LRN 2.0.3	MIT Massachusetts Institute of Technology	April 2003	21.000 User	IMS	http://dotlrn.org/

(*) als Orientierung für die Skalierbarkeit werden Angaben über Benutzerzahlen von produktiven Installationen herangezogen

(**) WCAG = Web Content Accessibility Guidelines. Richtlinien, die erläutern, wie Web-Inhalte für Behinderte zugänglich gemacht werden können (WCAG 1999)

Abb. 1: Beispiele von Open-Source-Learning-Management-Systemen

Nach Einschätzung von Schulmeister gibt es allein für den Bereich Learning-Management-Systeme (LMS) etwa 50 Open-Source-Produkte (SCHULMEISTER 2004). Im Rahmen der bereits angesprochenen Studie für das österreichische Bildungsministerium wurden insgesamt 82 Learning-Management-Systeme und Content-Management-Systeme recherchiert. Eine Evaluation des Commonwealth of Learning listet 35 Produkte (COMMONWEALTH OF LEARNING 2003). In der folgenden Tabelle werden beispielhaft einige in der aktuellen Diskussion häufig genannte Open-Source-Tools aus dem LMS-Bereich aufgeführt.

Auf die Funktionalitäten der aufgeführten LMS soll hier nicht weiter eingegangen werden. Es sei neben der Evaluation des Commonwealth of Learning noch auf ein Evaluationstool verwiesen, das vom Edutools-Projekt der Western Cooperative for Educational Telecommunications zur Verfügung gestellt wird (EDUTOOLS 2004).

In den USA werden von einigen renommierten Universitäten seit einiger Zeit bemerkenswerte Aktivitäten entwickelt, die möglicherweise großen Einfluss auf die zukünftige Entwicklung des Marktes für Learning-Management-Systeme haben werden. Hierzu gehören die Open-Knowledge-Initiative (OKI) und das Sakai-Projekt. OKI wurde im Juli 2001 gemeinsam vom MIT, dem Advanced Distributed Learning Network (ADL) und dem IMS Global Learning Consortium (IMS) gegründet (OKI 2001), mit dem Ziel, eine offene und erweiterbare Architektur für Learning-Management-Systeme zu entwickeln. Durch die Open-Source-basierte OKI-Architektur wird festgelegt, wie die Komponenten eines LMS miteinander kommunizieren und wie sie in andere LMS (auch in solche kommerzieller Hersteller) integriert werden können (OKI 2003). Bis jetzt wurden bereits zwölf so genannte Open-Service-Interface-Definitionen (OSID) veröffentlicht.

OKI

Die Open-Knowledge-Initiative ist darüber hinaus eines der Gründungsmitglieder des Sakai-Projektes. Das Gemeinschaftsprojekt wurde Ende 2003 mit den weiteren Partnern University of Michigan, Indiana University, MIT, Stanford und dem uPortal Consortium gegründet, um ein »Collaboration and Learning Environment (CLE)« unter Open-Source-Lizenz zu entwickeln, das die Anforderungen der Partner berücksichtigt und die Stärken der bereits existierenden Tools integriert. Mit dem Budget von 6,8 Millionen US\$, das dem Projekt für die Entwicklung zur Verfügung steht, sollen richtungsweisende Ergebnisse erzielt werden: »This effort will demonstrate the compelling economics of »software code mobility« for higher education, and it will provide a clear roadmap for others to become part of an open source community« (SAKAI 2004, 2004 a). Eine erste Version der auf den OSIDs der OKI basierenden Sakai-Tools wurde für Mitte dieses Jahres angekündigt. Die Version 1.0 soll ein komplettes Coursemanagement-System und ein Assessment-Tool enthalten. In der Version 2.0, die für Mitte 2005 geplant ist, sollen weitere Tools für die Bereiche Workflow Management, Research und Authoring hinzukommen (SAKAI 2004 a).

Sakai-Projekt

Die Projekte OKI und Sakai setzen beide auf eine Open-Source-Variante, durch die eine kommerzielle Nutzung der Entwicklungen unter bestimmten Bedingungen nicht ausgeschlossen werden soll. Sakai spricht von einer »open-open licensing« die für »open source, open for commercialization« (SAKAI 2004 a) steht. Obwohl auch eine Offenheit für Systeme kommerzieller Anbieter angestrebt wird, zielt die Strategie eindeutig auf eine breitenwirksame Implementierung der Open-Source-Lösung im amerikanischen Hochschulbereich und damit faktisch auf eine Verdrängung der kommerziellen

Produkte aus diesem Marktsegment. Diese Absicht wird durch den Plan des Sakai-Projektes gestützt, Tools für den Wechsel von proprietären Systemen wie WebCT oder Blackboard zu entwickeln und als Open-Source-Software zur Verfügung zu stellen. Allerdings betont Sakai, dass dies nicht zu den Kernaufgaben des Projektes gehört und dass keine Mittel zur Verfügung stehen, um Rechtsstreitigkeiten darüber zu finanzieren, ob Anwender das Recht haben, mit Hilfe dieser Tools ihre Daten aus proprietären Systemen herauszuziehen (SAKAI 2004 b). Es bleibt abzuwarten, welchen Erfolg diese ambitionierten Projekte erzielen werden und welche Auswirkungen sich auf die zukünftige Entwicklung des LMS-Marktes ergeben.

5 Vor- und Nachteile von Open-Source-Software

Unabhängig von der Frage, ob Open-Source-Tools im E-Learning-Bereich die spezifischen Anforderungen von Organisationen und Unternehmen treffen (was im Einzelfall zu analysieren ist), bestehen teilweise noch grundlegendere Bedenken gegen den Einsatz von Open-Source-Software. Im Folgenden werden die Vor- und Nachteile sowie die Potenziale von Open-Source-Software für E-Learning im Allgemeinen beleuchtet. Da die zur Diskussion stehenden Software-Produkte sehr unterschiedliche Entwicklungsstände aufweisen, treffen die Aussagen nicht auf alle Produkte in gleichem Maße zu. Auch hier gilt: Es ist im Einzelfall zu untersuchen.

Vorteile Ein wichtiges Merkmal von Open-Source-Software ist, wie oben bereits ausgeführt, dass sich prinzipiell jeder Anwender auch an der Weiterentwicklung der Software beteiligen kann, sofern er über die notwendigen technischen Kompetenzen verfügt. Es bilden sich dezentrale, zum Teil global verteilte Entwicklergemeinschaften, welche die Weiterentwicklung der Software betreiben, ohne an die Weisungen einer zentralen Institution gebunden zu sein. Die Entwickler entscheiden entsprechend ihrer individuellen Interessenlage selbst, welche Bereiche einer Software sie weiterentwickeln bzw. modifizieren möchten. Die Open-Source-Software-Produkte, die im Rahmen eines solchen offenen und partizipativen Entwicklungsprozesses entstehen, haben gegenüber kommerziellen und oft auch proprietären Closed-Source-Software-Produkten folgende potenzielle Vorteile (HANG/HOHENSOHN 2003, S. 34 f., WIELAND 2004, S. 109 f.):

- **Höhere Produktqualität**, z. B. höhere Sicherheit, da durch das Prinzip des »Peer-Review« innerhalb der Entwickler-Community eine intensivere Qualitätskontrolle erfolgt (unter der Voraussetzung, dass kompetente Entwickler beteiligt sind).
- **Wiederverwendbarkeit**, da die Quellcodes von Open-Source-Software offen liegen. D. h. diese können von den Entwicklern eingesehen und gegebenenfalls vollständig oder in modifizierter Form in einem neuen Produkt wieder verwendet werden. Auf diese Weise können Effizienzvorteile im Entwicklungsprozess realisiert werden, da »das Rad nicht jedes Mal neu erfunden« werden muss.
- **Höhere Reife der Software**, da kein Marktdruck besteht. Die Software wird i. d. R. erst dann veröffentlicht, wenn die Entwicklung vollständig abgeschlossen und Fehler beseitigt wurden.
- **Höhere Performance**. Open-Source-Software ist im Serverbereich nach Auffassung vieler Analysten häufig schneller als vergleichbare proprietäre Software.

- **Offenheit/Flexibilität**, da die Software beliebig an die speziellen Bedürfnisse des Kunden angepasst und auch die Interoperabilität zwischen unterschiedlichen Systemen hergestellt werden kann.
- **Stabilität**, da Open-Source-Software u. a. aufgrund des fehlenden Marktdrucks sorgfältiger getestet und kommentiert ist.

Als weiterer wichtiger Aspekt von Open-Source-Software wird der Preisvorteil durch die kostenlose Abgabe der Lizenz genannt. Dieser Vorteil wirkt sich in langfristigen Wirtschaftlichkeitsbetrachtungen positiv aus (GROB/BENSBERG 2002, S. 273 f.; HANG/HOHENSOHN 2003, S. 39 f.; WIELAND 2004, S. 112 f.). Allerdings ist zu beachten, dass die Lizenzkosten nur einen Teil der softwarebezogenen Gesamtkosten, die so genannten Total Cost of Ownership (TCO) ausmachen. Die TCO setzen sich aus den Kosten für die Beschaffung und den Betrieb sowie die Außerstandsetzung am Ende des Betriebs zusammen. Zu den Beschaffungskosten zählen neben den Planungskosten auch die Anschaffungskosten für Hard- und Software. Die Betriebskosten beinhalten die Inbetriebnahme, Schulung, technischen Support und Disposition. Bei der Außerstandsetzung müssen schließlich noch Kosten für die Entsorgung der Hardware und die Überspielung von Daten auf Nachfolgesysteme zugerechnet werden (HANG/HOHENSOHN 2003, S. 41). Das bedeutet, dass E-Learning auch mit Open-Source-Software leider nicht zum Nulltarif zu haben ist.

Kosten

Neben dem Argument der Kosteneinsparung beim Erwerb einer Open-Source-Lizenz ist aus strategischer Sicht noch die Unabhängigkeit von der Lizenzpolitik der kommerziellen Anbieter anzuführen. Die Anwender müssen bei größeren Änderungen der Funktionalität einer kommerziellen Software Updates häufig nachvollziehen, um ihre bisherigen Investitionen zu schützen. Dies kann z. B. der Fall sein, wenn alte und neue Version inkompatibel sind oder wenn der Hersteller den Support für ältere Versionen einstellt. Dadurch ist der Anwender gezwungen, ein Update einzukaufen, obwohl er möglicherweise an den enthaltenen Funktionalen Neuerungen keinen Bedarf hat (WIELAND 2004, S. 114). Außerdem kann die Abhängigkeit zu einem Anbieter zu Problemen führen, wenn dieser aufgrund einer Insolvenz den Markt verlässt oder von einem Konkurrenten aufgekauft wird. In diesem Fall kann die Weiterexistenz der Software gefährdet sein.

Unabhängigkeit

Dieses Problem einer unsicheren zukünftigen Existenz der Software wird häufig auch als Argument gegen Open-Source-Software vorgebracht, weil deren Entwicklung i. d. R. nicht oder nur schwach institutionell abgesichert ist. Dass das mit einer kommerziellen Software eingekaufte Gefühl einer sicheren Softwareexistenz aber auch trügerisch sein kann, belegt das jüngste Beispiel aus dem LMS-Bereich. Nach dem Merger von click2 learn und Docent wird das neue Unternehmen SumTotal seine Produktpalette auf der Architektur von click2 learn aufbauen, was faktisch bedeutet, dass das LMS von Docent vom Markt verschwindet (WANG 2003, 2004). Beim Einsatz eines Open-Source-Produktes könnte die Software vom Anwender prinzipiell selbst weitergepflegt werden, sofern der Stellenwert der Anwendung und die bisher getätigten Investitionen die hierzu notwendigen Aufwendungen rechtfertigen.

Produktpflege

Der als Vorteil genannte fehlende Marktdruck bei Open-Source-Entwicklungen kann sich für den professionellen Anwender auch als Nachteil erweisen. Dies ist insbesondere dann der Fall, wenn ein Nutzer auf die Realisierung bestimmter Funktionalitäten durch die Community »wartet«. Dieser Prozess lässt sich nur dann beschleunigen, wenn der Anwender die Entwicklung der gewünschten Funktionalität selbst betreibt. In der Regel kommt es jedoch

Release-Zyklen

dennoch zu regelmäßigen Release-Zyklen, die allerdings vorwiegend von einer Produkt- und Community-immanenten Dynamik geprägt sind.

Funktionsumfang Open-Source-Produkte verfügen nicht immer über einen Funktionsumfang, der mit kommerziellen Produkten vergleichbar ist. Das muss nicht unbedingt ein Nachteil sein, denn für bestimmte Szenarien reichen auch Tools mit geringerer funktionaler Komplexität aus. Dies kann nur vor dem Hintergrund des individuellen Einsatzszenarios beurteilt werden.

Bei der Frage, ob mit Open-Source-Software lediglich die Funktionalitäten von kommerzieller Software »nachgebaut« werden oder ob auch Innovationen hervorgebracht werden können, scheiden sich die Geister. Ein Vorteil von Open-Source-Software ist, dass der Quellcode offen liegt und jeder Entwickler auf die vorhandenen Lösungen aufbauen kann, um diese zu verbessern oder um komplexere Lösungsansätze zu entwickeln (WIELAND 2004, S. 110). Er muss »das Rad nicht neu erfinden«, sondern er kann sich vollständig auf die Verbesserung der Problemlösung konzentrieren. Kommerzielle Anbieter halten dem entgegen, dass Open-Source-Entwicklungen insbesondere im Business-Process-Kontext auf den beständigen Wandel allenfalls reagieren können, während kommerzielle Anbieter dagegen technologische Entwicklungen und Markttrends aufnehmen und aktiv gestalten (KRAEMER/KUECHLER 2004).

Produktreife Hinsichtlich der Produktreife, z. B. der Stabilität oder der Fehlerfreiheit, haben manche Open-Source-Produkte Defizite. Heute setzen aber auch bereits viele Open-Source-Projekte moderne Methoden des Software-Engineerings ein. Die Entwicklung erfolgt dann gemäß festgelegter Richtlinien für die Anforderungsspezifikation, die Implementierung und die Dokumentation sowie das Testen der Software. Ein Beispiel für einen solchen strukturierten Entwicklungsprozess sind die Richtlinien des Open-Source-Projektes ILLIAS (ILLIAS 2004).

Support Zum Qualitätsmanagement im Entwicklungsprozess gehört auch ein professionelles Release-Management. Software-Versionen, die sich nach einer Bewährungszeit als »Release Candidate« als fehlerfrei erwiesen haben, werden als »Stable Release« gekennzeichnet. Vom Einsatz von »Beta Releases« im Produktionsbetrieb ist dagegen abzuraten, da solche Versionen noch Änderungen unterliegen und auch Fehler enthalten können. Probleme mit fehlerhaften Produkten lassen sich am besten vermeiden, indem ein Support-Vertrag mit einem Dienstleister abgeschlossen wird, der den störungsfreien Betrieb sicherstellt.

6 Potenziale von Open-Source-Software im E-Learning-Bereich

Auch wenn die Liste der möglichen Funktionalitäten länger wird, kann kein kommerzielles Produkt alle denkbaren Anforderungen erfüllen. Um die Produkte nicht zu komplex werden zu lassen, müssen sich die Hersteller auf jene Features beschränken, die von einem möglichst großen Anwenderkreis genutzt werden können und dadurch ein maximales Marktpotenzial entfalten. Nach Auffassung vieler Experten haben beispielsweise die derzeit verfügbaren Produkte im LMS-Bereich noch Entwicklungsdefizite und sind ins-

besondere aus didaktischer Perspektive noch verbesserungsbedürftig (RINN/BETT 2003; TERGAN/ZENTEL 2003).

Die Erfüllung spezifischer Kundenanforderungen, wie etwa die Realisierung von didaktisch außergewöhnlichen Lösungen, ist im Rahmen von Standardprodukten sehr schwierig. Das Feedback und die Anregungen der Anwender sind von den Entwicklern zwar erwünscht und für eine marktorientierte Weiterentwicklung der Software erforderlich, aber nicht jeder Anwenderwunsch kann in die Release-Planung der Hersteller aufgenommen werden. Die Release-Planung orientiert sich verständlicherweise am technisch Machbaren und ökonomisch Erfolgversprechenden. Sonderwünsche können zwar von den Herstellern auch außerhalb der Release-Planung erfüllt werden, jedoch müssen die Entwicklungskosten dann i. d. R. zusätzlich zu den ohnehin anfallenden Lizenzgebühren in vollem Umfang vom Kunden getragen werden.

Die Bestimmungen der Open-Source-Lizenz räumen dem Anwender explizit die Möglichkeit ein, die Software eigenständig an seine Anforderungen anzupassen und weiterzuentwickeln. Eine solche Lizenzvereinbarung ist bei kommerzieller Closed-Source-Software nicht möglich, da die Hersteller aus verständlichen Gründen nicht die Kontrolle über den Source Code abgeben.

Anforderungen für Anpassungen

Die Voraussetzungen für die Durchführung von individuellen Anpassungen oder Weiterentwicklungen von Open-Source-Software sind jedoch relativ anspruchsvoll. Es werden sowohl für die Konzeption, als auch die Planung und softwaretechnische Realisierung spezielle Kompetenzen benötigt. Zur Konkretisierung der geplanten didaktischen Lehr-/Lernszenarien, die mit einem Open-Source-Tool unterstützt werden sollen, hat sich die Zusammenarbeit mit Personen aus dem pädagogischen Bereich bewährt, die auch über die notwendigen mediendidaktischen Kompetenzen verfügen. In der weiteren Vorbereitung der technischen Umsetzung ist es erforderlich, die didaktischen Anforderungen in technische Spezifikationen zu übersetzen, die dann softwaretechnisch realisiert werden. Für die Erstellung der technischen Spezifikation ist eine Schnittstellenkompetenz zwischen Didaktik und Informatik erforderlich, eben eine Person, die sowohl die Sprache der Pädagogen als auch der Informatiker versteht, die diese Spezifikationen letztlich programmieren müssen.

An die Programmierer stellt die Entwicklung von Open-Source-Software ebenfalls besondere Anforderungen. Sie müssen über die Kompetenz zur kollaborativen Entwicklung von Software verfügen bzw. in der Lage sein, sich diese anzueignen. So erfordert die kollaborative Software-Entwicklung technische Kompetenzen im Umgang mit speziellen Entwicklungstools (z. B. Concurrent Version Systems) sowie sozial-kommunikative Kompetenzen für den notwendigen Erfahrungs- und Wissensaustausch in der virtuellen Entwickler-Community.

Anders als in Hochschulen sind bei Bildungsträgern und in Unternehmen diese speziellen Kompetenzen häufig nicht vorhanden und können bzw. sollen personell auch nicht aufgebaut werden. In diesem Fall ist es möglich, spezielle Aufgaben im Zusammenhang mit Customizing und Weiterentwicklung an entsprechende professionelle Dienstleister im Umfeld des jeweiligen Open-Source-Produkts zu vergeben.

7 Erfolgsfaktoren für den Einsatz von Open-Source-Produkten

Die Erfolgsfaktoren für den Einsatz von Open-Source-Software stimmen in vielen Punkten mit denen von kommerzieller Software überein. Warum sollte das auch anders sein? Nur weil die Lizenz kostenlos abgegeben wird? Dennoch müssen beim Einsatz von Open-Source-Produkten noch einige zusätzliche Aspekte beachtet werden (Abb. 2).



1. Anforderungsanalyse und Leistungsspezifikation

Wie bei jeder Softwareanschaffung sollte zunächst das Einsatzszenario geplant und die daraus resultierenden Anforderungen definiert und gewichtet werden. Hieraus lassen sich notwendige Funktionalitäten und ihre Wichtigkeit ableiten.

2. Evaluation der Produkte

Die in Frage kommende Open-Source-Software muss anhand der Anforderungsspezifikation intensiv evaluiert werden. Dabei lassen sich auch etwaige Schwächen hinsichtlich der technischen Stabilität und Produktreife aufdecken.

3. Einsatz von »stable Releases«

Bei der Auswahl der in Frage kommenden Produkte sollten nur »stable Releases« berücksichtigt werden. Beta-Versionen sind zu fehleranfällig.

4. Aktivität der Entwickler-Community

Die Aktivität der Entwickler-Community ist ein Gradmesser für die zukünftige Entwicklung des Produkts. Es sollten in regelmäßigen Abständen neue Versionen mit neuen Releases veröffentlicht werden. Nur wenn ein kontinuierlicher Ausbau der Software erkennbar ist, verfügt die Community über eine ausreichende Dynamik, um das Tool an die sich wandelnden Anforderungen anzupassen. Als Orientierung über die mittelfristig geplante Weiterentwicklung kann auch die so genannte Roadmap dienen, die i. d. R. im Internet veröffentlicht wird.

5. Verbreitung des Produkts

Eine große Anzahl von Anwendern spricht für ein stabiles und leistungsfähiges Open-Source-Produkt und stellt sicher, dass neue Versionen von vielen Anwendern getestet und evtl. Fehler schnell gefunden und beseitigt werden können. Auskunft über die Nutzer-Community sollte eine Liste von Referenzinstallationen geben, die im Internet verfügbar ist.

6. Aktivität der Nutzer-Community

Die Aktivität der Nutzer-Community ist ebenfalls ein wichtiger Aspekt für die zukünftige Entwicklung der Software. In Supportforen, die i. d. R. über die Homepages der Produkte zugänglich sind, tauschen sich die Anwender über ihre Erfahrungen und Probleme aus und helfen sich gegenseitig bei der Lösung von Problemen. Außerdem werden in solchen Nutzerforen auch gewünschte Änderungen und Weiterentwicklungen der Tools diskutiert.

7. Berücksichtigung von Standards

Wie auch bei der Anschaffung einer kommerziellen Software sollten zum Schutz der eigenen Investitionen in Content nur Produkte ausgewählt werden, die zukunftsorientierte Standards (wie z. B. LOM und SCORM) unterstützen. Dadurch wird eine Investitionsfalle vermieden, die auftreten kann, wenn ein Produktwechsel erforderlich wird, z. B. wenn durch das Produkt die eigenen Anforderungen nicht mehr abgedeckt werden.

8. Professioneller Support

Ein besonders wichtiger Aspekt für den erfolgreichen Einsatz von Open-Source-Software ist die Existenz von professionellem Support. Es sollten spezialisierte Dienstleister vorhanden sein, mit denen Vereinbarungen (Service Level Agreements) zur vertraglichen Absicherung eines störungsfreien Betriebs sowie einer regelmäßigen Wartung und Pflege getroffen werden können.

9. Organisatorische Implementierung

Wie auch bei der Implementierung von kommerzieller Software erfordert die Einführung einer Open-Source-Software Maßnahmen zur Organisationsentwicklung, wie z. B. Schulungen der Mitarbeiter, für die finanzielle Mittel zur Verfügung stehen müssen. Auch für den nachhaltigen Betrieb und einen eventuellen späteren Ausbau müssen personelle und finanzielle Mittel eingeplant werden.

Abb. 2: Checkliste zum Einsatz von Open-Source-Produkten

8 Fazit

Es sind inzwischen vielfältige Open-Source-Produkte für unterschiedlichste Anwendungen verfügbar. Wie Evaluationen belegen, besitzen sie teilweise einen Funktionalitätsumfang und eine Produktreife, die mit kommerziellen Produkten durchaus mithalten kann. Dennoch muss im Einzelfall geprüft werden, ob für ein spezifisches Einsatzszenario ein Open-Source-Produkt zur Verfügung steht, das die jeweiligen Anforderungen erfüllt. Erfüllt das ausgewählte Produkt darüber hinaus noch weitere Erfolgsfaktoren, wie eine angemessene Verbreitung, eine aktive Entwickler- und Nutzer-Community sowie professionellen Support, kann es als Alternative zu einem kommerziellen Produkt in Erwägung gezogen werden. Der Preisvorteil durch die kostenlose Verfügbarkeit kann dann sinnvoll in spezifische Anpassungen oder Weiterentwicklungen des Produkts, die Bereitstellung von Content oder in Maßnahmen zur organisatorischen Implementierung investiert werden.

Literaturhinweise

- BAUMGARTNER, P./HÄFELE, H./MAIER-HÄFELE, K.: E-Learning Praxishandbuch – Auswahl von Lernplattformen: Marktübersicht – Funktionen – Fachbegriffe, Innsbruck/Wien 2002.
- BAUMGARTNER, P./HÄFELE, H./MAIER-HÄFELE, K.: Evaluierung von Lernmanagement-Systemen: Theorie – Durchführung – Ergebnisse, in: Hohenstein, A./Wilbers, K. (Hrsg.): Handbuch E-Learning, Köln 2002 a, Beitrag 5.4.
- CAMPUSOURCE: Open Source und universitäre Tradition, online (2000): <http://www.campussource.de/opensource/#unitradition> (Abruf am 18. 5. 2004).
- CAMPUSOURCE: Ziele der Initiative, online (2000 a): <http://www.campussource.de/wir/#ziele> (Abruf am 18. 5. 2004).
- COMMONWEALTH OF LEARNING: COL LMS Open Source. Findings of a survey-style evaluation of Open Source Learning Management System Software commissioned by the Commonwealth of Learning from 3 waynet Inc., June 2003, online (June 2003): <http://www.col.org/Consultancies/03LMSOpenSource.htm> (Abruf am 18. 5. 2004).
- EDUTOOLS: Comparison of Course Management Systems. EduTools is based on the Landonline comparisons of course management systems developed by C2T2 (the Centre for Curriculum, Transfer & Technology) in British Columbia and Bruce Landon of Douglas College, also in British Columbia, online (2004): <http://www.edutools.info/course/compare/bygroup/selectproducts.jsp?group=2> (Abruf am 18. 5. 2004).
- FREE SOFTWARE FOUNDATION: Die Definition Freier Software, online (2002): <http://www.gnu.org/philosophy/free-sw.de.html> (Abruf am 18. 5. 2004).
- GLOBAL LEARNING: Topic des Monats: Learning Management Systeme als Open Source Software. Kostenlos = Kostenfrei? Online (1/2004): http://www.global-learning.de/g-learn/cgi-bin/gl_userpage.cgi?StructuredContent=m0701 (Abruf am 18. 5. 2004).

- GROB, H. L./BENSBERG, F.: Strategische Potenziale von Open Source-Software für die computergestützte Hochschullehre (cHL), dargestellt am Beispiel des cHL-Administrationssystems OpenUSS, in: BACHMANN, G./HAEFELI, O./KINDT, M. (Hrsg): Campus 2002. Die Virtuelle Hochschule in der Konsolidierungsphase, Münster u. a. 2002, S. 262–276.
- HANG, J./HOHENSOHN, H.: Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte. Eine Studie im Rahmen des Projektes NOW: Nutzung des Open Source Konzeptes in Wirtschaft und Industrie, C-LAB Report, Vol. 2 (2003), No. 2, S. 10 ff., online: <http://www.c-lab.de/home/de/reports/index.html#2003-02> (Abruf am 18. 5. 2004).
- HEISE ONLINE: Open Source in der Schweiz gleichberechtigt, online (2004): <http://www.heise.de/newsticker/meldung/45357> (Abruf am 18. 5. 2004).
- ILIAS: ILIAS Development, online (2004): <http://www.homer.ilias.uni-koeln.de/iliasdoc/doc/html/8.html> (Abruf am 18. 5. 2004).
- IMC: Programm des 4. Learning Management Congress 2003, online (2003): http://www.im-c.de/elearning/programm_mittwoch.htm (Abruf am 18. 5. 2004).
- KERBUSK, K.-P.: Mauerfall an der Isar, in: Spiegel 23/2003, online: <http://www.spiegel.de/spiegel/0,1518,251077,00.html> (Abruf am 18. 5. 2004).
- KHARITONIOUK, S./STEWIN, P.: Einleitung, in: GEHRING, R. A./LUTTERBECK, B. (Hrsg.): Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin 2004, online: <http://www.Think-Ahead.org/> (Abruf am 18. 5. 2004).
- KLEMME, D./ARPAGAU, J./ROSENTHAL, D.: OSS-Strategie der Bundesverwaltung, online (2004): http://www.isb.admin.ch/imperia/md/content/strategien/opensource/oss_strategie/oss_strategie_v_1-0_2004-02-23_genehmigt_d.pdf/ (Abruf am 18. 5. 2004).
- KRAEMER, W./KUECHLER, T. (2004): Statement zum Thema Open-Source-Software, in: Handbuch E-Learning online (1/2004): http://www.global-learning.de/g-learn/cgi-bin/gl_userpage.cgi?StructuredContent=m07031603 (Abruf am 18. 5. 2004).
- KRISTÖFL, R.: Evaluation von Learning Management Systemen, online (3/2003): <http://www.bildung.at/system/downloader.php/Evaluation-LMS.pdf/> (Abruf am 18. 5. 2004).
- KRISTÖFL, R.: Evaluation von Lernplattformen: Verfahren, Ergebnisse und Empfehlungen, online (3/2004): <http://www.bildung.at/system/downloader.php/LMS-Evaluation20040422.pdf/> (Abruf am 18. 5. 2004).
- OKI: MIT Open Knowledge Initiative, ADL Co-Laboratory, and IMS Cooperate to Advance Learning Technology, Presse-Erklärung vom 29. Juni 2001, online: http://web.mit.edu/oki/learn/press/oki_adl_ims.html (Abruf am 18. 5. 2004).
- OKI: Product Description. Learning management system = Enterprise system, online (2003): <http://web.mit.edu/oki/product/index.html> (Abruf am 18. 5. 2004).

- OPEN SOURCE INITIATIVE: The Open Source Definition. Version 1.9 online (2004): <http://www.opensource.org/docs/definition.php> (Abruf am 18. 5. 2004).
- RINN, U./BETT, K.: Lernplattformen zwischen Technik und Didaktik, in: BETT, K./WEDEKIND, J. (Hrsg.): Lernplattformen in der Praxis, Münster 2003, S. 193–209.
- SAKAI: Project Summary, online (2004): <http://www.sakaiproject.org/presentations/about.pdf> (Abruf am 18. 5. 2004).
- SAKAI: Project Overview, online (2004 a): <http://www.sakaiproject.org/presentations/Sakai10slideOverview.ppt> (Abruf am 18. 5. 2004).
- SAKAI: FAQ: What migration support is envisioned for moving from current primary proprietary systems currently in use on campus (for example, WebCT and Blackboard)? online (2004 b): http://www.sakaiproject.org/faqs.html#What_migration_support (Abruf am 18. 5. 2004).
- SALARI, S.: What's the Difference between Open Source and Open Systems? Präsentation auf der 3rd Annual WebCT European Conference, Amsterdam 2004, online: <http://www.webct.com/service/ViewContent?contentID=20665839> (Abruf am 18. 5. 2004).
- SCHULMEISTER, R.: Statement zum Thema Open-Source-Software, in: Handbuch E-Learning online (1/2004): http://www.global-learning.de/g-learn/cgi-bin/gl_userpage.cgi?StructuredContent=m07031604 (Abruf am 18. 5. 2004).
- SIGOSSEE (2004): SIG Open-Source Software for Education in Europe online (2004): <http://www.ossite.org/> (Abruf am 18. 5. 2004).
- TERGAN, S.-O./ZENDEL, P.: Lernplattformen und die Zukunft des E-Learning, in: BETT, K./WEDEKIND, J. (Hrsg.): Lernplattformen in der Praxis, Münster 2003, S. 223–240.
- WANG, E.: Saba: Massive PR-Angriffe auf Docent, in: Global Learning News vom 1. 12. 2003, online: http://www.global-learning.de/g-learn/cgi-bin/gl_userpage.cgi?StructuredContent=m060293 (Abruf am 18. 5. 2004).
- WANG, E.: SumTotal: »Wir konkurrieren jetzt gegen SAP und Peoplesoft«, in: Global Learning News vom 8. 4. 2004, online: http://www.global-learning.de/g-learn/cgi-bin/gl_userpage.cgi?StructuredContent=m06029930 (Abruf am 18. 05. 2004).
- WCAG: Web Content Accessibility Guidelines 1.0., W3C Recommendation 5-May-1999, online: <http://www.w3.org/TR/WAI-WEBCONTENT/> (Abruf am 18. 5. 2004).
- WIELAND, T.: Stärken und Schwächen freier und Open-Source-Software im Unternehmen, in: GEHRING, R. A./LUTTERBECK, B. (Hrsg.) Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin 2004, S. 121–134, online: <http://www.Think-Ahead.org/> (Abruf am 18. 5. 2004).