

Kim Neunert

Pädagogische Aspekte freier Software

Zum Einsatz freier Software in Schulen

Der Einsatz von Software im Unterricht kann sehr unterschiedlich betrachtet werden. Die grundlegendste Entscheidung beim Einsatz von Computern an der Schule ist die Wahl des Betriebssystems. Allerdings scheinen hier pädagogische Aspekte so gut wie gar nicht vertreten zu sein. Die Wahl des Betriebssystems wird wohl derzeit noch am ehesten von dem Pflegeaufwand, der Bedienbarkeit oder der Anzahl verfügbarer Anwendungen beeinflusst.

Pädagogische Aspekte werden hauptsächlich auf die Anwendungssoftware angewandt. Gerade im Multimediabereich gibt es mittlerweile unzählige pädagogische Bewertungen von Software. Sicherlich ist es wesentlich einfacher, eine spezielle Anwendungssoftware nach pädagogischen Aspekten zu betrachten, als eine ganze Sparte von Software, die sich noch dazu auf den ersten Blick nur durch ein äußerst abstraktes Merkmal unterscheidet. Freie Software ist zunächst, ohne Lizenzgebühren entrichten zu müssen, frei kopierbar. Das bedeutet nicht, dass freie Software keiner Lizenz unterliegt.

Diverse Lizenzen auch bei freier Software

Die bekannteste und wohl am häufigsten eingesetzte Lizenz ist die GPL (GNU Public License). Diese Lizenz ermöglicht ein freies Kopieren und auch Verändern der Software, da die Quelltexte, die man zum Verändern der Software braucht, immer mitgeliefert werden müssen. Die veränderte Software unterliegt wiederum der GPL, darf also ebenfalls frei kopiert und verändert werden. Man könnte also von Gemeinschaftseigentum sprechen.

Das Gegenteil wäre proprietäre Software: also Software, die einen Eigentümer besitzt, der die Verwendungsmöglichkeiten einschränkt.

Freie Software heißt nun nicht, dass sie umsonst zugänglich sein muss. Freie Software wird auch verkauft. Aber, wer würde sie schon kaufen, wenn man sie umsonst aus dem Internet laden kann?

Dafür kann es gute Gründe geben: Wenn beispielsweise die Zeit bzw. die Kosten beim Download aus dem Internet zu groß werden, dann kauft der oder die Interessentin sich das Produkt lieber, zum Beispiel auf CD-ROM. Ein weiterer Grund wäre die Vorauswahl, welche die Verkäufer für die Kundin oder den Kunden treffen. Sie oder er kann im Geschäft sicherer sein, dass die Software einen gewissen Qualitätsstandard besitzt. Außerdem ist die Installation von der CD-ROM erheblich einfacher.

Freie Software kann sowohl Betriebssysteme wie auch Anwendungssoftware oder jede beliebige andere Software (Programme zum Betrieb des Internets etc.) einschließen. Freie Software ist also unglaublich breit gefächert. Kann man denn dann überhaupt generelle pädagogische Aussagen über freie Software im Unterrichtseinsatz treffen? Ja, man kann.

Freie Verfügbarkeit

Seit den 60er Jahren dieses Jahrhunderts ist die Bildung vor allem sozial benachteiligter Schichten in unserer Gesellschaft ein wichtiges Thema. Trotz hoher Ansprüche scheint sich hier in der Praxis nicht viel Wesentliches getan zu haben. Im Gegenteil: Laut der internationalen Schulstudie PISA gehört Deutschland zu den Ländern, in denen die Bildungswohlstandschere besonders stark auseinander klafft. Besonders die informationstechnische Bildung soll, sofern man unseren Bildungsstrategen glaubt, eine der wichtigsten Schlüsselqualifikationen der Zukunft sein. Freie Software kann ohne lizenzrechtliche Bedenken kostenlos an die Schüler verteilt werden. Wenn man bedenkt, dass damit die Schüler völlig umsonst die gleiche Software zuhause einsetzen können wie im Unterricht, ist das ein enormer Vorteil. Das betrifft nun nicht nur einzelne Programme, sondern auch ganze Betriebssysteme (damit ist vor allem Linux gemeint).

Dies ist vor allem für sozioökonomisch benachteiligte Schüler wichtig, denn auch wenn auf heutigen Computern bereits ein Betriebssystem (nämlich Windows) vorinstalliert ist, kostet dies doch versteckt Lizenzgebühren. Dieser Effekt äußert sich aber nicht nur schlicht in der Geldfrage: So ist es an unserer Schule möglich, die Desktops dank KDE in verschiedenen Sprachen laufen zu lassen. So können selbst Schüler, die des Deutschen nur wenig mächtig sind, an die Benutzung einer grafischen Oberfläche herangeführt werden. Dies ist ein entscheidender Vorteil. Eine Möglichkeit, die besonders dort genutzt werden kann, wo sozial und kulturell integrativ gearbeitet werden muss.

(Von Freier Software und Bildung, LinuxMagazin 3/2000, S. 74)

Freie Verfügbarkeit als größtes Plus

Die freie Verfügbarkeit ist aber auch für die Schule bzw. die Lehrerinnen und Lehrer sehr vorteilhaft. Mit proprietärer Software kann – wenn die Schule entsprechende Sammellizenzen geordert hat – meist nur eine geringe Bandbreite an Software Lehrern und Schülern zur Verfügung gestellt werden. Bei freier Software jedoch kann die Lehrkraft nach eigenen didaktischen Gesichtspunkten eine Auswahl von einem oder mehreren Programmen treffen. Für eine bestimmte Funktionalität können also mehrere Programme eingesetzt werden. Man ist nicht darauf beschränkt, ein einzelnes Programm zum Unterrichtsgegenstand zu machen, sondern die Funktionalität der Programme kann in den Mittelpunkt gestellt werden.

Informationstechnische Medienkompetenz wird sich in der Zukunft nicht auf die Bedienung einzelner Programme reduzieren lassen. In der momentanen Diskussion hat man den Eindruck, ein Schüler würde in der Berufswelt eine Bruchlandung erleben, wenn er nicht schon in der Schule die Bedienung der Office-Suite von Microsoft gelernt hätte.

Erkennen von Konzepten statt Erlernen von Produkten

Der Markt für Softwareprodukte lebt von ständiger Innovation. Eine Version jagt die Nächste. Würde ein Schüler heute die Bedienung einer Anwendung in der Schule lernen, so ist fast mit Sicherheit davon auszugehen, dass diese Anwendung überholt ist, wenn der Schüler ins Berufsleben eintritt. Die Erlernung der Bedienung von Anwendungen lässt sich nachholen, sonst würde unsere Wirtschaft nicht länger funktionieren. Warum sollen nicht die Firmen, welche die Software verkaufen, deren Bedienung im Berufsleben (sozusagen just-in-time) schulen? Welche Voraussetzungen oder Kenntnisse muss nun der später Berufstätige haben bzw. in der Schule erlernen haben, um später die Bedienung einer Anwendung möglichst schnell zu beherrschen?

Wissen um unterschiedliche Lösungsmöglichkeiten

Was in der Schule gebraucht wird, ist vor allem die Kenntnis von übergreifenden Konzepten.

Schaut man sich beispielsweise Textverarbeitungsprogramme an, so arbeiten diese (fast) alle mit den gleichen Konzepten: Formatvorlagen, Überschriften, automatisch erstellte Inhaltsverzeichnisse, eingebettete Grafiken und vieles mehr. Natürlich ist die Bedienung jeweils – bei Textverarbeitungen meist nur marginal – verschieden. Natürlich gibt es auch Unterschiede, die über die Bedienung hinausgehen. Ein Programm verwendet vielleicht ein völlig anderes Lösungskonzept für ein Problem (z.B. die Textverarbeitung LyX). So wird das Wissen über die unterschiedlichen Lösungsmöglichkeiten und deren Vor- und Nachteile zu einer Schlüsselqualifikation. Dies ist nur möglich, wenn den Schülern die unterschiedlichsten Implementierungen dieser Lösungskonzepte, also die unterschiedlichen Anwendungen, zur Verfügung stehen.

Freie Software und selbstgesteuertes Lernen

Proprietäre Software vermittelt oft den Eindruck von intuitiver Bedienbarkeit: plug and play auch für den Benutzer. Ob dies so immer funktioniert, sei dahingestellt. In einer schulischen Bildung, die Wert auf informationstechnische Kompetenz legt, kann ein solches Konzept nicht unbedingt überzeugen. Differenzieren sollte man hier zwischen den Bildungsinhalten der unterschiedlichen Fächer. Bei Benutzung eines multimedialen Weltatlas in Erdkunde sollte ein produktives Arbeiten schnell möglich sein. Gleichwohl: Wer bereits mit drei unterschiedlichen Textverarbeitungsprogrammen vertraut gemacht wurde, der wird die Bedienung des Vierten umso schneller erlernen.

Wenn es Probleme gibt, wird ein Lernprozess bei Schülern nur dann einsetzen, wenn sie selbst das Handbuch lesen. Das Hantieren mit technischen Dokumentationen und das Aufstöbern der gesuchten Informationen gehört mit Sicherheit genauso zu den informationstechnischen Schlüsselqualifikationen (die übrigens vor allem bei der Studie PISA geprüft wurden), wie der selbstbewußte Umgang mit Anwendungen, ohne das Handbuch durchgelesen zu haben.

Dokumentation freier Software durch Mailinglisten

Die Dokumentation freier Software hat mittlerweile einen erheblich höheren Stand erreicht. Dies betrifft sowohl die Lesbarkeit für Laien als auch die Einbettung in die Anwendung (wie

Handbücher in HTML). Aber auch bei mangelnder Dokumentation oder darüber hinausgehenden Fragen wird dem Schüler/Lehrer mit dem Internet ein Medium bereitgestellt, mit dem weitere Fragen beantwortet werden können – dort im speziellen News-/Mailinglisten bzw. News-/Mailinglistenarchive. Damit kann in der Schule bereits das eingeübt werden, was der Schüler spätestens in der Arbeitswelt können sollte: Zum einen die selbständige beziehungsweise selbst gesteuerte Suche und Auswahl relevanter Informationen, zum anderen die Fähigkeit, in elektronischen Kommunikationsmedien angemessen agieren zu können.

Transparenz

Schüler sind in zunehmendem Maße von elektronischen Geräten aller Art umgeben. Nicht nur der Computer, auch das Handy, die Spielkonsole, der videotextfähige Fernseher usw. gehören zum Lebensalltag der Kinder und Jugendlichen. Diese Geräte werden zwar in der Bedienung immer einfacher. Andererseits bauen sie eine unsichtbare Schranke auf, hinter die das Kind nicht mehr blicken kann. Das Tamagotchi als Black-Box. Wie viel Kinder haben bitterlich geweint, als das kleine Geschöpf das Zeitliche segnete?

Das Prinzip lässt sich aber auch auf ältere Alterstufen und ebenso auf Software übertragen. Zu den Zeiten des Commodore 64 hat wohl fast jeder Jugendlicher mindestens eine Endlosschleife mit Ausdruck auf dem Bildschirm programmiert. Dies konnte jeder als Beispiel aus dem Handbuch abschreiben. Dies war nur möglich, weil früher die Homecomputer eine Art eingebaute Programmiersprache enthielten.

Das ist beim Kauf eines heutigen PCs – bei denen mittlerweile nicht mal mehr der früher noch enthaltene Basicinterpreter mitgeliefert wird (GWBasic) – ganz anders. Was fehlt, ist Transparenz

Computer für Kinder verständlich machen

Der vor allem auch in der Pädagogik wichtige Begriff der Transparenz findet sich in besonderem Maße bei der freien Software wieder. Das bezieht sich im schulischen Kontext nur in zweiter Hinsicht auf die Tatsache, dass zu jedem Open Source Softwareprodukt die Quellcodes offen liegen.

Das Design des Betriebssystems Linux kann dem Schüler vor Augen führen, was ein

Betriebssystem leisten muss, wie es funktioniert und wie es den eigenen Bedürfnissen angepasst werden kann: Dass ein Softwareprodukt nicht vom Himmel fällt, dass es grundsätzlich Fehler enthält, in verschiedenen Versionen vorliegt, ursprünglich mit Hilfe von Quelltext und einem Compiler erzeugt wurde (dies lässt sich auch am Rechner nachvollziehen) sind Tatsachen, die im Rahmen freier Software wesentlich besser erkannt und vermittelt werden können.

Linus Torvald, der Erfinder von Linux schreibt: „Tatsächlich waren Computer damals besser für Kinder geeignet: Sie waren weniger ausgereift als heute, und Grünschnäbel wie ich konnten unter der Haube daran herumbasteln. Heutzutage ist es mit Computern wie mit Autos: Mit zunehmender Komplexität wurde es schwieriger, sie auseinander zu nehmen und wieder zusammenzubauen und auf diese Weise zu lernen, was eigentlich dahinter steckt.“ (Torvald, L.: Just For Fun, 2001, S.24)

Verschiedene Anwendungsbeispiele aus der Praxis

Ein gutes Beispiel für eine praktische Anwendung unter Linux ist das Programm Gimp. Gimp ist ein frei erhältliches Grafikprogramm, dass in der Funktionalität sich durchaus mit professionellen Grafikprogrammen wie beispielsweise Photoshop messen kann. Das Arbeitsergebnis von Gimp ergibt sich aus der freien Verwendung unzähliger kleiner Funktionen, die nacheinander ausgeführt, zum letztendlichen manipuliertem Bild führen. Bei dieser Vorgehensweise stehen die Einzelteile, die zusammengesetzt das Gesamtergebnis liefern, im Vordergrund.

Bei proprietärer Software kommt es meistens darauf an, dem Benutzer zu demonstrieren, dass man mit wenigen Mausklicks tolle Ergebnisse „zaubern“ kann. Zu diesem Zweck werden meist sogenannte „Wizards“ eingesetzt, die den Benutzer durch einen komplizierten Erstellungsprozess leiten sollen. Mehrere Fragemasken werden vom Benutzer, in freudiger Erwartung des Ergebnisses meistens mithilfe des Buttons "Weiter" angeklickt. Am Ende steht z.B. ein aus zwölf eigenen Fotos erstellter Jahreskalender, der dem Benutzer eindrucksvoll die Leistung des Programms vor Augen führen soll. Auch wenn ein solcher Wizard einfach zu bedienen ist, stellt sich die Frage, welcher der zwei Ansätze den größeren Lerneffekt verspricht.

Die Bedienung von Gimp ist zuerst ein wenig eigentümlich. Für den Benutzer von Windows ist zunächst ungewohnt, dass für alle Bedienelemente (Werkzeugkasten, Pinselauswahl, Arbeitsfläche etc.) ein eigenes Fenster zur Verfügung gestellt wird. Des weiteren erfolgen große Teile der Bedienung über das Kontextmenü. Hier lässt sich die Datei speichern, Filter anwenden und vieles mehr.

Beispiel LyX: LyX ist ein Textverarbeitungsprogramm, das ganz anders funktioniert als die WYSIWYG-Textverarbeitungsprogramme, die beispielsweise unter Windows laufen. Die Formatierungsmöglichkeiten scheinen eingeschränkt zu sein. Die Bildschirmansicht entspricht nicht dem, was man später auf Papier zu Gesicht bekommt. Es steht ein völlig unterschiedliches Konzept dahinter. Übrigens ein Konzept, mit dem der spätere Student der Naturwissenschaften sich spätestens zur Erstellung seiner Diplomarbeit auseinandersetzen muss.

Vielfalt statt Einfalt der Programme

Wenn man sich die verschiedenen Programme anschaut, so sind diese auf den ersten Blick bunt durcheinander gewürfelt. Einige Programme haben ähnliche gestylte Buttons, andere gar keine. Wieder andere lassen sich ohnehin nur über die Konsole bedienen. Was zunächst wie ein Manko aussieht, kann auch völlig anders betrachtet werden: Man hat die volle Auswahl.

Und weiter: Der Schüler wird schnell merken, dass das Aussehen nur ein sekundäres Merkmal bei der Auswahl darstellt. Wichtig wird nur die Funktionalität: Können die Programme zusammen verwendet werden? Dies hängt nicht vom Aussehen ab, sondern von den verwendeten Datenformaten. Das ist eine für die Schüler wichtige und grundlegende Erkenntnis. Programme arbeiten nicht deswegen Hand in Hand zusammen, weil sie die gleiche Oberfläche besitzen, sondern weil sie austauschbare oder sich ergänzende Datenformate benutzen. Diese Datenformate müssen nicht, wie dies bei proprietärer Software oft implizit mitvermittelt wird, an eine Anwendung gekoppelt sein. Es existieren auch produktunabhängige Datenformate, die sogar für den Menschen lesbar sein können. Eine Webseite in HTML lässt sich zur Not auch mit einem einfachen Texteditor erstellen.

Im Internet kann sie dann mit unterschiedlichen Programmen (den Browsern) von unterschiedlichen Herstellern betrachtet werden. Wenn ein Schüler nur mit Produkten eines Herstellers arbeitet, wird es für ihn schwer sein, sich über die Bedeutung von offenen

Standards im klaren zu werden. Im Gegensatz dazu scheint für Schüler die einheitliche Oberfläche einen wesentlich bedeutenderen Stellenwert zu besitzen: Auffallend bei einigen wenigen Schülern war, wie sehr sie schon auf das bloße Aussehen von Windows fixiert waren. Als einer von ihnen durch Zufall fvwm95 (ein Windowmanager unter Linux, der wie Win95 aussieht) sah, fragte er entrüstet: „Warum haben Sie uns das vorenthalten?“
(Von Freier Software und Bildung, LinuxMagazin 3/2000, S. 74)

Stabilität

Unbestritten bringt der einmal etablierte Einsatz freier Software eine höhere Betriebs- und Zugangssicherheit mit sich. Dies ist mit Sicherheit nicht nur ein technisches Argument. In jeder Bildungssituation muss man sich als Schulungsleiter beziehungsweise Lehrer auf seine Werkzeuge verlassen können. Ausfallzeiten wegen Instabilität oder Datenverluste der Schüler erzeugen nicht nur schlechte Lernergebnisse, sondern mindern auch auf die Motivation der Schüler und die Lust am Lernen. Natürlich ist keine Software ohne Fehler und Systemabstürze können nie ausgeschlossen werden. Im Umfeld freier Software wird mit dieser Tatsache aber wesentlich offener, ehrlicher und lösungsorientierter umgegangen. An beruflichen Schulen können Instabilitäten in freier Software beispielsweise für den Unterricht fruchtbar gemacht werden. Während Systemabstürze bei proprietärer Software in der Berufsschulpraxis häufig nur mit einem Achselzucken und einem Neustart abgetan werden, können Instabilitäten bei freier Software zu einem tieferen Verständnis der Probleme der Informatik führen. Im Anschluss daran kann sich der Berufsschüler die Frage stellen: Gibt es vielleicht für diesen Fehler eine rationale Erklärung und eine Lösungsmöglichkeit? Ist der Fehler bekannt? Habe ich die neueste Version installiert?

Bildung auf Basis ehrenamtlicher Arbeit

Freie Software wird weitgehend ohne monetäres Entgelt in Zusammenarbeit von vielen Freiwilligen mithilfe des Internets erstellt. Das Arbeitsergebnis wird der Allgemeinheit mit Hilfe der GNU Public License (GPL) der Allgemeinheit zur Verfügung gestellt. Es ist Gemeinschaftseigentum. Wird andere Art von Software in der Bildung eingesetzt, so ist diese notwendigerweise proprietäre Software, die unter wirtschaftlichen Aspekten vermarktet wird. In unserer heutigen Zeit, in der wirtschaftliche Aspekte immer weitere Teile unseres Lebens

beeinflussen, stellt also freie Software eine Möglichkeit dar, Bildung – zumindest in einem Teilbereich – ohne Einwirkungen aus der Welt der Wirtschaft zu ermöglichen.

Das Jahr 2001 war bundesweit das Jahr des Ehrenamts. Die Bedeutung ehrenamtlicher Arbeit sollte in der Gesellschaft wieder mehr Beachtung finden. Vielleicht kann auch die ehrenamtliche Arbeit in der freien Softwareproduktion speziell für die schulische Bildung über das Jahr 2001 hinaus mehr an Bedeutung gewinnen. An der Schnittstelle von freier Software und Bildung haben sich mehrere untereinander kooperierende Initiativen auf Basis ehrenamtlicher Arbeit gebildet. Der Verein Freie Software und Bildung (FSuB e.V.) wurde formell 1999 gegründet (<http://fsub.schule.de>). Er ging aus einem Zusammenschluss des Arbeitskreis Schule der GUUG (German Unix User Group) und dem informellen Kreis „Linux für Schulen“ hervor. Im Moment hat der FSuB ungefähr 100 Mitglieder. Die Aktivitäten reichen von Schulungen und Lobbyarbeit über die (Vor-)Auswahl geeigneter bildungsrelevanter Software bis hin zur Unterstützung und/oder Initiierung von Projekten, in denen für die Bildung relevante Software geschrieben wird.

Die Linux-Gemeinschaft entwickelt sich weiter

Ein zweites Projekt, das 1999 aus einem Diskussionsstrang in einer Newsgruppe im Internet entstanden ist, ist PingoS (Ping on Schools). Ping ist ein Linux/Unix-Befehl, mit dem man feststellen kann, ob ein entfernter Computer sich im Netzwerk befindet. Analog dazu versteht sich PingoS als Zusammenschluss von Linux-Kennern, die Schulen im praktischen Umgang mit Linux im Rahmen ehrenamtlicher Arbeit unter die Arme greifen. Zudem sieht PingoS seine Aufgabe in der Dokumentation von Linux (<http://www.selflinux.de>) und in der Erstellung bildungsnaher Software (Tipptrainer, Zeugnistool etc.). PingoS hat deutschlandweit mittlerweile über 100 eingetragene Helfer (<http://www.pingos.org>).