Open Source Projekte im Informatikunterricht

Tim Tenbusch
Institut für Informatik,
Freie Universität Berlin
tenbusch@inf.fu-berlin.de

Kurzfassung

Open Source Entwicklung erlebt einen rasanten Aufschwung. Mittlerweile gibt es mehrere tausend Projekte und einige haben bereits den Sprung zu einer fertigen und voll funktionsfähigen Anwendung geschafft. Für die Schulen und den Unterricht bedeutet dies, dass in vielen Bereichen nicht mehr auf ein kommerzielles Produkt zurückgegriffen werden muss und der Fachbereich dadurch Geld spart. Aber auch im Unterricht sollte Open Source ein festes Gebiet zugestanden werden, spiegelt es doch einen der erfolgreichsten Bereiche der Informatik wieder. Neben der Anwendung von Open Source Produkten sollte daher auch auf Open Source selbst eingegangen werden. Die Arbeit handelt deshalb davon, wie SchülerInnen der Sekundarstufe II selbst Open Source EntwicklerInnen werden und anhand dieser Erfahrung mehr zu dieser Art von Softwareentwicklung lernen.

Inhalt

1. Einführung

Open Source Projekte erleben in den letzten Jahren einen rasanten Aufschwung. Mittlerweile gibt es viele Programme, die sich als ernst zu nehmende Konkurrenten gegenüber kommerzieller Software entwickeln. Open Source ist heute nicht nur ein Geschäftsmodell und Entwicklungsstil, sondern ist zu einer Gemeinschaft geworden, die Erkenntnisse und Arbeitsergebnisse für allen Menschen zur Verfügung stellen. Wikipedia beispielsweise ist ein Wissensprojekt, welches aus der Open Source – Bewegung entstand.[7]

In der Schule wird der Begriff Open Source meist gar nicht thematisiert. Die SchülerInnen lernen Textverarbeitung, Betriebssystem und Programmierumgebung als Standard kennen, unabhängig davon, ob es sich um Open Source oder proprietäre Software handelt.[8] Ein Konflikt ergibt sich meistens nur dann, wenn ein anderes System zu Hause steht. Ansonsten lernen die SchülerInnen in der Schule die Software als etwas gegebenes kennen. Ob nun Open Source oder proprietäre Software ist den SchülerInnen dabei egal.[8] Die SchülerInnen arbeiten sich in die Software ein und binden sich an diese. Aber die Schule ist kein Ort, an dem die Benutzung einer Software trainiert wird, sondern zentraler Bestandteil des schulischen Unterrichts ist die Vermittlung von Kompetenzen in Textverarbeitung, Tabellenkalkulation und Programmierung, unabhängig von den eingesetzten Programmen.[3]

Ein Vorteil bei der Nutzung von Open Source liegt in den wegfallenden Kosten für die Software, dies sollte aber nicht der alleinige Grund sein, sondern der Unterricht kann diesen Unterschied gezielt thematisieren. Zum einen werden sich die SchülerInnen bewusst, dass es zumeist für kommerzielle Anwendungssoftware eine kostenlose Alternative gibt und dass sie selbst diese mit ihren im Informatikunterricht erworbenen Fähigkeiten weiterentwickeln könnten.

Der Informatikunterricht hat schon wegen seiner allgemeinbildenden Aufgabe die Pflicht Open Source im Unterricht zu behandeln und zu diskutieren.[3] Denn Open Source ist mittlerweile eine bedeutende Softwareentwicklungsform, hat eine große Community und hat damit einen festen Platz in der Informatik. Das Thema könnte auch im Rahmen des s.g. Projektsemesters in der Sekundarstufe II eingebunden werden. Hier müssen die SchülerInnen nämlich ein eigenes Softwareprojekt entwickeln und können dieses auch als Open Source Projekt starten. Dadurch setzen sich die SchülerInnen aktiv mit der Mentalität und Philosophie von Open Source auseinander. Thema dieser Arbeit ist es, die theoretischen Bedingungen für die Durchführung eines Open Source Projekts zu analysieren und mit den herkömmlichen Methoden des Informatikunterrichts zu vergleichen.

Dazu gehe ich kurz auf die Definition von Open Source ein und beschreibe die schulischen Rahmenbedingungen und die didaktischen Probleme von Open Source Entwicklung im Unterricht. Diese werden dann mit dem herkömmlichen Ansatz der schulischen Projektarbeit vergleichen. Zum Schluss stelle ich die beiden Ansätze gegenüber und erörtere welchen Ansatz unter welchen Bedingungen der bessere ist.

2. Open Source

Mit dem Begriff Open Source wird jener Teil von Software bezeichnet, der kostenlos, quelloffen und damit für jeden zugänglich ist. Die Definition ist fast deckungsgleich mit der Freien Software. Im Gegensatz zu dieser muss die Software aber unter einer Lizenz der Open Source Initiative stehen und die Bedingungen dafür erfüllen[1]. Erst dann darf diese Software Open Source genannt werden. Damit ist die Open Source Initiative die Dachorganisation für alle Software, die als Open Source lizenziert wird. Außerdem beschäftigt sie sich mit der Förderung von Projekten und Vermarktung des Begriffs. Sie wurde im Februar 1998 von Bruce Perens und Eric Raymond gegründet.

Die zentralen Bedingungen für Open Source Software sind, dass die Software beliebig kopiert, genutzt und verbreitet werden darf, ohne dass dafür eine Gebühr gezahlt oder Lizenzen

erworben werden müssen. Weiterhin muss die Software in allen Teilen als Quellcode, d.h. in einer Form vorliegen, die für Menschen lesbar ist. Dieser Quellcode muss ebenfalls kostenlos zur Verfügung gestellt werden. Ebenso darf der Quellcode verändert oder angepasst werden und in der neuen Form auch weitergegeben oder vertrieben werden. Auch dürfen Teile der Software in gänzlich anderen Projekten verwendet werden. Zentral ist auch, dass keinerlei Einschränkungen gegenüber Einsatzgebieten der Software oder benutzenden Personen gemacht werden. [1]

Der Begriff Open Source wurde aus Marketinggründen gewählt und soll von der Freien Software abgrenzen, die als kostenlos und damit als unwirtschaftlich eingestuft wird. Dabei muss Open Source Software nicht unkommerziell sein. Die Software darf im Gegensatz zur Freien Software verändert oder sogar unverändert verkauft werden. Trotz der Freiheit des Quellcodes hängt der Open Source Software der gewollte Ruf der finanziellen Nutzbarkeit an, also dass sich mit dieser Software Geld erwirtschaften lässt.[9]

Im Gegensatz zu proprietärer Software ist niemand der Eigentümer der Software, sondern die Software ist Allgemeingut und steht unter einer dementsprechenden Lizenz. Damit steht Open Source Software auch oftmals in direktem Konkurrenzkampf zu den kostenpflichtigen Produkten. Aber oftmals ist trotz einer kostenlosen und quelloffenen Alternative das kommerzielle Produkt bei den Benutzern erfolgreicher. Dies hängt insbesondere mit der sozialen Trägheit der Menschen zusammen[2]. Daraus folgt auch, dass in absehbarer Zeit der Erfolg proprietärer Software nicht gefährdet ist.

Die Zahl der Open Source Projekte ist groß. Die erfolgreichsten und meist genutzten Open Source Projekte sind Linux und der Browser von Mozilla. Zu den meisten proprietären Softwaren gibt es mittlerweile ein Open Source Projekt, insbesondere wenn die Software von sehr vielen Menschen genutzt wird oder die Entwickler ein besonders eigenes Bedürfnis in diesem Bereich haben. Gleichwohl gibt es für zahlreiche Anwendungsbereiche keine Open Source Projekte und die Benutzer müssen weiterhin für die Nutzung der Software bezahlen. Dies sind meist Randanwendungen, die nur für eine sehr eingeschränkte Nutzergruppe entwickelt werden.

3. Unterrichtseinheit

Ein Open Source Projekt in der Schule durchzuführen ist mit wenigen Änderungen an der herkömmlichen Projektmethode verbunden und daher leicht umsetzbar. Zentral ist aber, dass die SchülerInnen die Open Source Philosophie kennen und sich selbst tendenziell dazu bekennen. Denn den SchülerInnen stellt sich nämlich die Entscheidung, ob sie ihr Softwareprojekt mit Quelltext kostenlos veröffentlicht wollen. Dieses Bekenntnis eine Software für alle zu veröffentlichen ist nicht für jeden ein einfacher Schritt.

Ein Open Source Projekt durchzuführen ist mit einem Mehraufwand verbunden, der aber meist nicht so groß ist, wie von der Lehrkraft befürchtet. Zum einen muss die Unterrichtsreihe geändert werden, um beispielsweise mit einem CVS-System zu arbeiten und zum anderen müssen die Kommunikationswege auf eine computergestütztes Medium verlagert werden. Ebenfalls entsteht Mehraufwand, wenn die SchülerInnen ein vorhandenes Projekt erweitern wollen oder ein Projekt in Kooperation mit anderen Klassen oder Schulen entsteht. Hier muss die Lehrkraft innerhalb kurzer Zeit sich selbst in das Projekt einarbeiten.

Neben diesen Anforderungen an die LehrerIn gibt es aber auch einige institutionelle Rahmenbedingungen.

3.1. Rahmenbedingungen

Rahmenlehrplan

Im Folgenden beziehe ich mich bei der Analyse des Rahmenlehrplanes auf den der Länder Berlin, Brandenburg und Mecklenburg-Vorpommern. [3] In den anderen deutschen Bundes-

ländern ist aber in ähnlicher Form ein solches Projekt vorgesehen, wenn Informatik in der Sekundarstufe II unterrichtet wird.

Der Rahmenlehrplan schreibt sowohl für den Grund- als auch den Leistungskurs im vierten Semester ein Softwareprojekt vor. Dieses findet also im letzten Schulhalbjahr statt. Die schriftlichen Abiturprüfungen sind zu diesem Zeitpunkt bereits geschrieben und die SchülerInnen bereiten sich auf das mündliche Abitur und ihre Präsentationsprüfung vor. Daher eignet sich dieses Semester auch besonders, um sich im Rahmen eines Projekts auf die Prüfungen vorzubereiten. Auch geht der Rahmenlehrplan zu diesen Zeitpunkt davon aus, dass alle wichtigen Kompetenzen und Methoden der Informatik in den vergangenen Semestern gelehrt wurden, so dass dies ein reines Anwendungssemester ist, mit nur wenig neuer Theorie.[3]

Im vierten Semester müssen daher nur die Grundlagen der systematischen Softwareentwicklung, den s.g. Software-Life-Cycle und die der Ergonomie wiederaufgegriffen werden. Dazu soll ein neues Projektthema begonnen oder ein altes fortgeführt werden. Das Projektthema soll mindestens so komplex sein, dass der Software-Life-Cycle mit allen seinen implizierten Methoden als notwendig erlebt wird. Auf der anderen Seite darf das Projekt auch nicht so komplex sein und die SchülerInnen sollen mindestens einen Prototypen fertig stellen und diesen testen. Die Projektarbeit darf sich nicht nur auf eine Phase der Softwareentwicklung und aus reiner Programmiertätigkeit beschränken. Eine der Phasen muss auch die Anwendersicht beinhalten. Das Projekt soll von den SchülerInnen selbstständig lösbar sein und dies möglichst in Eigenorganisation. Zur Wahl des Themas lässt der Rahmenlehrplan viel Spielraum und gibt nur den Hinweis, dass das Thema aus dem Erfahrungskreis der SchülerInnen kommen sollte, um zeitintensive Sachanalysen bei völlig fremden Themen zu vermeiden. Außerdem sollte auf bereits bekanntes Wissen und den Interessen der SchülerInnen aufgebaut werden.

Neben fachlichen Inhalten stehen auch die Methodenkompetenzen der Informatik auf dem Lehrplan. Diese sind allerdings nicht einem bestimmten Unterrichtshalbjahr oder Thema zugeordnet, sondern stellen die Standards dar, die am Ende der Schulzeit von jeder SchülerIn beherrscht werden sollten. Daher ist bei dem Thema Projektarbeit, neben den gesamten Methoden die zum Problemlösen, Modellieren und Programmieren benötigt werden auch Soziale Kompetenzen nötig. Diese werden im Rahmenlehrplan unter ‚Kommunizieren und Kooperieren – Teamarbeit organisieren und koordinieren' zusammengefasst. Für das Projektsemester hat die Lehrkraft besonders auf eine angemessene Fachsprache der SchülerInnen im Unterricht und im Gespräch untereinander zu achten. Außerdem sollen netzbasierte Kommunikations- und Kooperationssysteme für die Gruppenarbeit eingesetzt werden und die Netiquette soll beachtet werden. Weiterhin sollen GrundkursschülerInnen selbstständig ihre Projektarbeit organisieren, während der Leistungskurs zusätzlich diese selbständig planen und anleiten soll.

Zeitplanung

Für die im Rahmenlehrplan vorgesehene Projektarbeit muss ein Projekt gewählt werden, welches die SchülerInnen aber auch beenden können. Dazu muss abgeschätzt werden, wie viele Unterrichtsstunden zur Verfügung stehen und wie viele Zeitstunden die SchülerInnen an dem Projekt arbeiten könnten. Das Projektsemester ist von dem Rahmenlehrplan für das zweite Halbjahr geplant und dieses ist zeitlich kürzer als das erste. Insgesamt ist das zweite Schulhalbjahr 20 Wochen lang und Informatikunterricht wird in Grundkurs mit 3 und im Leitungskurs mit 5 Unterrichtsstunden in der Woche gelehrt. Dies ergibt 60 bzw. 100 Unterrichtsstunden. Bei der Langzeitplanung gilt die Faustregel, dass im normalen Schulbetrieb ca. 20% der Fachstunden ausfallen.[10] Die Gründe dafür sind vielfältig, so müssen Krankheit, Wander-, Feier- und Studientage, Klausuren sowie der Zensurenstopp in der Planung berücksichtigt werden. Auch ist eine Unterrichtsstunde nie mit 45 Minuten Arbeitszeit zu planen, da für Organisationsaufgaben selten weniger als fünf Minuten benötigt werden. Auch ist die reine Arbeitszeit durch Störungen oder Ablenkungen geringer, als die noch übrigen 40 Minuten. In

der Unterrichtspädagogik werden als reine Arbeitszeit für eine Schulstunde ca. 10 bis maximal 35 Minuten angegeben.[10] Informatikunterricht hat hierbei den Vorteil, dass die Arbeitszeit durch strukturelle Gründe meist im oberen Bereich liegt. Für die konkrete Planung gehe ich von 40 Minuten Unterricht pro Stunde aus und rechne 30 Minuten pro Woche für Hausarbeiten hinzu. Dies ist auch der empfohlenen Werte für die wöchentliche Dauer von Hausarbeiten pro Fach. Im Leitungskurs liegt dieser Wert bei ca. 2 Stunden. Zusammengerechnet ergibt dies für den Grundkurs 2460 bzw. 5240 Minuten für den Leitungskurs (41 bzw. 87 Zeitstunden). Der höhere Wert für den Leitungskurs kommt insbesondere durch die längere Hausaufgabenzeit zusammen.

Die Zeit die für einen Grundkurs bleibt ein Open Source Projekt durchzuführen ist also beschränkt und gerade einmal eine normale Arbeitswoche mit 8 Stunden am Tag. Die laut Rahmenlehrplan zu durchlaufenden Softwareentwicklungsschritte sowie die Einarbeitung in die für ein Open Source Projekt nötigen Techniken und Methoden schränken die Themenwahl des Projekts stark ein.

Im Leistungskurs stehen hingegen komfortable zweieinhalb Wochen zur Verfügung. Damit lassen die Rahmenbedingungen es noch besser zu ein Open Source Projekt durchzuführen, welches auch am Ende des Entwicklungszeitraums ein akzeptables Ergebnis vorweisen kann. Die Themenwahl kann hier etwas erweitert werden.

Projektthema

Die SchülerInnen sollen an der Planung und Durchführung von Projekten aktiv beteiligt sein. Damit folgt auch der Rahmenlehrplan der ‚neuen' pädagogischen Linie der Schülerpartizipation. Für den Unterricht heißt dies, dass nicht mehr die LehrerIn ein geeignetes Thema vorgibt, sondern die SchülerInnen sollen ein Thema aus ihrem Interessensbereich auswählen. Die Möglichkeiten sind vielfältig und unmöglich abzuschätzen. Daher gelten bei dem Auswahlprozess nur wenige Kriterien.

Das Projekt muss so gestaltet sein, dass alle SchülerInnen, gerade auch die schwächeren, sich aktiv beteiligen können. Daher müssen alle Sonderwünsche zu bestimmten Techniken oder Sprachen nur im Sinne der Allgemeinheit entschieden werden. Es nützt nichts wenn 2-3 SchülerInnen den Großteil der Arbeit erledigen und die restlichen nur für die Dokumentation, Entwurf oder Hilfe zuständig sind. Grundsätzlich gilt, dass alle SchülerInnen in der Lage sein müssen an jeder Stelle des Projekts z.B. eine kranke SchülerIn ersetzen können. Sonderwünschen ist nur nachzugeben, wenn das allgemeine Projekt nicht beeinträchtigt wird. Beispielsweise wenn zwei SchülerInnen ein besonderes Feature entwerfen darf dies meiner Meinung nach nicht länger als 2 Schulstunden dauern, bis sie zur eigentlichen Projektarbeit zurückkehren.

Dem Interesse der meisten SchülerInnen ist bei der Projektwahl nachzugehen. Das Projekt ist für die SchülerInnen da, um ihre Kompetenzen zu entwickeln. Und damit sie besonders motiviert sind, muss das Projekt aus ihren Erfahrungsraum stammen. Keine SchülerIn wird bei dem Projekt Bibliothekssoftware überdurchschnittlich motiviert sein, wohl aber bei einer Chat-Software oder einem Browsergame.

Hier ist dann aber auch die nächste Schwierigkeit bei der Wahl des Projektthemas. Meistens wollen die SchülerInnen Spiele entwickeln und gleich in die 3D-Programmierung einsteigen. Solche Projekte scheiden aber sofort aus, da sie zu komplex sind und in der Zeit nicht zu entwickeln sind. Der Lehrkraft kommt hier die schwierige Aufgabe zu, den Entwicklungsaufwand abschätzen zu können und den SchülerInnen bei der Komplexität zu bremsen. Gleichzeitig dürfen aber nicht zu viele Schülerwünsche über Bord geworfen werden, sonst verlieren sie das Interesse an dem Projekt. Ein didaktischer Trick dafür ist, diese Wünsche zwar im Pflichtenheft als optional festzuhalten, aber zuerst darauf einigen, dass ein lauffähigen Prototypen entwickelt wird.

Das Projekt sollte aus didaktischer und pädagogischer Sicht am besten aus einem oder mehreren anderen Unterrichtsfächern stammen, um ein fächerübergreifendes Projekt durchführen zu können. Denn das Wissen, welches bei einer solchen Arbeit erworben wird, ist vernetztes Wissen und daher präsentes, abrufbares Wissen.

3.2. Didaktische Analyse

Die Rahmenbedingungen ermöglichen es also durchaus ein Softwareprojekt an der Schule als Open Source Projekt durchzuführen. Aber gerade im Grundkurs ist mit der wenigen Zeit die zur Verfügung steht ein enger Zeitrahmen gesetzt.

Open Source Entwicklungen sollten in einer solchen Projektumgebung durchgeführt werden, für den Informatikunterricht ist dies aber nicht zwingend. Vielmehr hängt die Art der Durchführung damit zusammen, wie sich die Klasse zusammensetzt und welche Vorkenntnisse die SchülerInnen haben. Der Einsatz eines CVS-Systems ist für jedes größere Projekt vom Vorteil, erfordert aber auch, dass die SchülerInnen dieses erst kennen und benutzen lernen. Den Einsatz müssen die SchülerInnen auch erst üben, damit nicht große Teile eines Projekts durch fehlerhafte Bedienung verloren gehen. Auch die Einarbeitung in die neuen Kommunikationswege müssen erst trainiert werden, gerade wenn das Projekt mit anderen Fächern, Kursen oder Schulen durchgeführt wird.

Bei der Planung und Durchführung des Projekts muss die Schülerzusammensetzung berücksichtigt werden. Welche Vorkenntnisse haben die SchülerInnen und die Frage, welche SchülerInnen besonderen Förderungsbedarf haben bestimmen zum großen Teil die Art der Projektarbeit. Der Grundsatz, dass jeder eine beliebige Arbeit durchführen kann, muss hier berücksichtigt werden. Bei Open Source Entwicklungen besteht die Gefahr, dass wenige SchülerInnen den Großteil der Arbeit machen, während die meisten nur in Nebenrollen am Projekt beitragen. Daher sollten während des Projekts die Entwicklungsteams rotieren, so dass jede SchülerIn an jedem Projektteil mitgearbeitet hat.

Bei der Schülerzusammensetzung spielt leider auch die Anzahl der SchülerInnen eine entscheidende Rolle. In Berlin kann es Grundkurse mit 3-32 SchülerInnen geben. Während mit wenigen SchülerInnen gute Projektarbeit durchgeführt werden kann, ist die Arbeit an einem Projekt mit mehr als 20 SchülerInnen im Schulalltag nicht leicht zu bewältigen. Hier müssten dann mehrere verschiedene Projekte durchgeführt werden oder verschiedene Teams arbeiten am gleichen Projekt mit unterschiedlichen CVS-Systemen.

Weiterhin müssen die SchülerInnen auch im gesamten Projektsemester motiviert werden, gerade wenn das Projekt stockt oder die Entwurfsphase den SchülerInnen zu lange dauert. Dieser Teil darf im Unterricht nicht vernachlässigt werden, da gerade eine Open Source Arbeit von der Motivation der Beteiligten abhängt und der Bereitschaft über die normale Hausaufgabenzeit an dem Projekt zu arbeiten.

Eine Vernetzung mit anderen Schulen ist zwar für die beteiligten Lehrer schwer zu organisieren und mit Mehraufwand bei der Planung verbunden, für die SchülerInnen aber dafür umso interessanter. Wenn diese nämlich die vorhandenen Kommunikationsmedien nutzen müssen, um ihre Arbeit durchzuführen, erhalten sie einen direkten Einblick in die reale Open Source Entwicklung.

Im Grundkurs ist die Durchführung des Projekts in der Hand der SchülerInnen. Die LehrerIn ist nur helfende BegleiterIn. Dieser selbstorganisierte Unterricht schreibt der Rahmenlehrplan vor. In der Regel sind aber die SchülerInnen nicht an eine solche Unterrichtsführung gewohnt, da die meiste Zeit eine lehrerzentrierter Frontalunterricht durchgeführt wird. Daher ist die Eigenorganisation der SchülerInnen immer zu überwachen und bei Bedarf sofort einzugreifen. Im Leistungskurs sind sie Anforderungen an die Selbstorganisation der SchülerInnen noch stärker und die Lehrkraft muss hier noch mehr die Arbeiten der SchülerInnen kontrollieren.

Als letztes stellt sich die Frage, ob ein neuen Projekt erstellt oder an einem bestehenden Open Source Projekt mitgearbeitet werden sollte. Das neue Projekt ist zweifelsfrei einfacher als ein bestehendes. Denn in ein bestehendes müssen sich die Betreffenden erst einarbeiten und es kann immer Überraschungen geben, wie z.B. eine neue Technik die unbekannt ist oder dass die Struktur oder Kommentierung des Quelltext unverständlich ist. Zum anderen ist es auch informatische Sichtweise besser ein bestehendes Projekt beispielsweise nur um ein Modul zu erweitern. Dies würde ein Projekt im kleinen Bedeuten und keine größere Einarbeitung erfordern. Die Entscheidung was durchgeführt wird, ist daher prinzipiell von dem Thema abhängig, welches die SchülerInnen bearbeiten wollen.

Wenn die SchülerInnen Open Source als Projektarbeit erstellen, sollten sie sich auch der Gründe dafür gewusst sein und ihr eigenes Handeln reflektieren können. Dazu sollen sie auch ein Wissen über Urheberrechte sowie den Unterschied zwischen Freier, kommerzieller und Open Source Software kennen. Auch muss die Mentalität und die Philosophie von Open Source Thema des Unterrichts sein. Ansonsten ist die Arbeit an dem Projekt nur eine vom Lehrer vorgeschriebene Tätigkeit und die Ideen von Open Source werden für die SchülerInnen nur eine leere Hülle sein.

4. Vergleich herkömmlicher Unterrichtsmethoden

Der neue Ansatz für die Projektarbeit verspricht neben der Einbindung von dem Thema Open Source in den Unterricht auch für die SchülerInnen interessanter zu sein und ihnen mehr Kompetenzen zu vermitteln als der bisherige Ansatz. Aber wie wurde bisher das Projektsemester durchgeführt und welche Aspekte in der Projekttheorie sind interessant für ein Open Source Projekt? Die Frage soll dieses Kapitel klären.

4.1. Projektarbeit

Die Softwareentwicklung des vierten Semesters soll laut Rahmenlehrplan in Projektarbeit geführt werden. Dabei bearbeitet eine Gruppe von Lernenden ein Gebiet. Sie plant ihre Arbeit selbst und führt diese auch aus. In der Regel steht am Ende ein sichtbares Produkt. Das lernen in Projektarbeit erstreckt sich über mehrere Stunden, die möglichst zusammenhängen sollten. Am Anfang steht eine Projektinitiative, die aus dem Lehrplan, vom Lehrer oder den SchülerInnen selbst kommen kann. Bedürfnisse, Neigungen und Interessen der Lernenden können und sollen mit in das Projekt einfließen, genauso wie andere Fachgebiete mit einbezogen werden sollten. Am Ende eines Projekts entsteht ein präsentierbarer Gegenstand.[4] Die in Projekten erreichten Lerneffekte sind aufgrund der Vernetzung mit anderen Wissensgebieten, der Einbindung eigener Interessen und der Praxisorientierung vielschichtiger, tiefergehend und resistenter gegen das Vergessen. Auch sind durch die Anwendungspraxis die Erkenntnisse leichter in andere Wissensgebiete übertragbar und bilden das besonders wertvolle Transferwissen.

Der Ablauf von einem Informatikprojekt läuft mit einigen Ausnahmen wie ein allgemeines Projekt ab.[5] Am Anfang steht eine Projektinitiative. In dieser diskutieren die SchülerInnen die Projektvorschläge und nehmen eine erste Projektplanung vor. Die verschiedenen Initiativen müssen so präsentiert werden, dass das Interesse der MitschülerInnen geweckt wird. Zu einer Eingrenzung und um die Komplexität der Aufgabe in der Informatik abzuschätzen, wird bei ausgewählten Themen eine Projektskizze erstellt. Diese besteht aus zwei Teilen, einer kurzen inhaltlichen Auseinandersetzung und einer Diskussion über die Durchführbarkeit und das Interesse der Beteiligten. Einigen sich die SchülerInnen auf ein Projekt, wird die Projektskizze vervollständigt.

Als nächsten Schritt wird der Projektplan erstellt. Im Informatikunterricht ist die inhaltliche Projektplanung das Pflichtenheft. Der zweite Teil ist die Arbeitsorganisation. Hier müssen die SchülerInnen entscheiden, wie sie die Arbeit im Projekt durchführen wollen. Dazu müssen auch noch einige organisatorische Punkte geklärt werden, wie z.B. Projektleiter und Zeitpla-

nung. Des Weiteren sind Termine für Zwischenpräsentationen, Klausuren oder Deadlines für bestimmte Projektstadien festzusetzen. Beispielsweise muss die Deadline für die Fertigstellung des Prototyps mindestens 3 Wochen vor Ende des Schuljahres sein, damit die Arbeit noch bewertet werden kann. Des Weiteren muss geplant werden, welche Techniken und welche Software eingesetzt werden soll und welche Voraussetzungen dafür nötig sind.

Nachdem diese Planung abgeschlossen ist beginnt die Durchführung eines Projekts. Bereits hier sollte möglichst arbeitsteilig gearbeitet werden, um ein reales Projekt zu simulieren. Als erstes wird das Problem modelliert. Dies wird zuerst mit Grobentwürfen realisiert. Diese Entwürfe werde dann miteinander verglichen und aus dem besten ein genaues Modell des Problems erstellt. Diese soll möglichst so gestaltet werden, dass eine Modularisierung möglich ist. Der Entwurf wird dokumentiert und mit graphischen Mitteln visualisiert. Ist der globale Entwurf fertig, wird die Bearbeitung der Teilgebiete arbeitsteilig durchgeführt. Dazu müssen sich die Projektteilnehmer selbstständig einem Teilgebiet zuordnen und die Arbeit in den neuen Teilgruppen mit organisieren. In den Teilgruppen wird die globale Modellierung für das eigene Modul nochmals verfeinert und bei Bedarf wird die Gruppe nochmals geteilt. Ziel ist es, dass es eine nachvollziehbare Teilung der Arbeit der SchülerInnen gibt.

Die vollständigen Teilentwürfe werden nun präsentiert und einer Bewertung durch die Gruppe unterzogen, um konstruktive Kritik zuzulassen. Anschließend wird der Entwurf implementiert. Die genaue Vorgehensweise bei der Implementierung ist Organisationssache der SchülerInnen.

Nach Fertigstellung des ersten Prototyps wird dieser einem Funktionstest unterzogen. Dieses Testen wird sowohl aus der Entwickler- als auch Anwendersicht durchgeführt. Ist die Software möglichst fehlerfrei wird das Ergebnis mit dem Modell und dem Entwurf verglichen. Das Pflichtenheft, als Teil der Aufgabenstellung, wird ebenfalls mit der Software verglichen. Anschließend muss die Funktionsweise mit der Realität verglichen werden, um die Funktionsweise zu validieren. Gerade die letzen drei Schritte eines jeden Projekts werden häufig nicht beachtet. Diese sind aber gerade die wichtigsten Schritte, um sich zu überzeugen, dass die entwickelte Software genau die Aufgaben löst für die sie entwickelt wurde. Der letzte Schritt eines jeden Projekts ist die Präsentation. Zu diesem Zeitpunkt muss jeder Projektschritt durchlaufen worden sein. Anschließend findet die Bewertung des Projektes statt.[5],[6]

Die Bewertung des Projekts muss nicht nur durch den Lehrer erfolgen. Durchaus kann auch eine Projektbewertung durch die SchülerInnen selbst erfolgen. Dies sollte natürlich möglichst durch die LehrerIn begleitet werden und die LehrerIn sollte das Recht auf die endgültige Festlegung der Note bei sich behalten.

4.2. geleitetes Projekt

Der Rahmenlehrplan schreibt zwar ein selbstorganisiertes Projekt vor, dieses ist aber in der Definition so weit gefasst, dass es auch ein geleitetes Projekt einschließt. Dieses Projekt ist dann aus Sicht der Projekttheoretiker kein Projekt mehr, sondern ein lehrergelenkte und geplante Unterrichtsreihe. Dabei wird das Projekt inklusive Thema von Lehrer vorgegeben. Bei der Darstellung dieser Projektform beschreibe ich meine eigenen Beobachtungen diverser Schulstunden, die ich im Rahmen meines Studiums hospitiert habe.

Die SchülerInnen erhalten am Anfang der Stunde konkrete Aufgabenstellungen und werden so durch das Projekt geführt. Diskussionen über den Entwurf oder die Modellierung werden als lehrergelenktes Unterrichtsgespräch durchgeführt, indem die SchülerInnen keine Möglichkeit haben auf das Projektthema, den Entwurf oder die Modellierung Einfluss zu nehmen. Auch die Durchführung wird vom Lehrer geplant. Die SchülerInnen erhalten zwar jeder ein eigenen Teilbereich, den sie Programmieren müssen, haben aber durch eine genaue Spezifikation keinerlei Gestaltungsmöglichkeiten.

Die Methode ist bei allen Nachteilen und entgegen jeder neuen pädagogischen Unterrichtsforschung in ausgewählten Fällen sinnvoller, als der eigenständige Projektunterricht. Bei-

spielsweise ist es in Klassen mit extremen Leistungsgefällen, sozialen, ethnischen und Disziplinproblemen keine gute Idee, den SchülerInnen zu viel Eigenverantwortung zu geben. Dies wird dann meist ausgenutzt und das Projekt wird nie die Planungsphase verlassen. Auch sind viele SchülerInnen nicht an die Selbstorganisation ihres Lernens gewöhnt. Am Ende ihrer Schulzeit, ohne jegliche Erfahrung mit Selbstorganisation ein komplettes Schulhalbjahr zu planen ist dann der falsche Ansatz. Daher ist es besonders wichtig im Informatikunterricht den Projektansatz von Anfang an zu üben und selbstorganisiertes Lernen in jeder Beziehung zu fördern.

5. Fazit

Das Projektsemester verfolgt von der Intention her zwei Ziele. Zum einen sollen die SchülerInnen einen Einblick in die professionelle Softwareentwicklung gewinnen und alle dafür nötigen Schritte selbst durchlaufen. Zum anderen sollen sie ihre soziale Kompetenz unter Beweis stellen und vertiefen, dass sie in einer Gruppe an einem gemeinsamen Projekt arbeiten können. Diese vom Rahmenlehrplan verfolgten Ziele lassen sich unabhängig vom Thema Open Source erreichen.

In einem herkömmlichen Projekt, dass mit richtiger Projektarbeit durchgeführt wird, entwickeln die SchülerInnen selbstständig aus einer Idee eine Software, die sie im Idealfall später einmal selbst regelmäßig nutzen werden. Genau hier liegt auch der Ansatzpunkt das Projekt Open Source zu entwickeln. Warum sollen nur die SchülerInnen an dem Projektergebnis teilhaben? Vielmehr könnten die SchülerInnen eine interessante Entwicklung freigeben und für alle zugänglich machen, damit viele Menschen ihre Software nutzen können. In der Informatikdidaktik wird oft bemängelt, dass die im Unterricht geschriebene Programme ihren Nutzen bereits am Ende der Unterrichtsstunde verlieren. Hier kann den SchülerInnen gezeigt werden, dass ihre Software mehr Wert hat als sie vermuten. An einem Berliner Gymnasium wurde z.B. in Rahmen des Projekthalbjahres eine Vertretungsplansoftware entworden, die über das Schulnetzwerk an jedem Rechner und im Internet angezeigt werden konnte. Über Monitore wird in dieser Schule nun in der Eingangshalle der aktuellste Vertretungsplan elektronisch angezeigt. Andere Berliner Schulen interessierten sich hingegen für ein kommerzielles System. Dieses sollte zusammen mit einem elektronischen Schwarzen Brett und eben dieser Vertretungsplansoftware mehrere tausend Euro kosten. Wäre die Vertretungsplansoftware von Anfang an als Open Source geplant worden, hätten sich einige Schulen die Anschaffung sparen können und auf das wesentlich kostengünstigere Modell zurückgreifen können.

Aber nicht immer haben die im Informatikunterricht entwickelten Projekte nur einen Nutzen für die Schulen. Auch könnten die SchülerInnen an einem nützlichen Plug-In für eine bestehende Open Source Software mitgearbeitet haben, die nach der Veröffentlichung von vielen genutzt werden kann. Die Motivation für solch ein Projekt ist bei den SchülerInnen meist schon dann erreicht, wenn sie sich vorstellen können, dass ihre Software tatsächlich einmal benutzt wird und dass sie mit ihrem Projekt etwas sinnvolles und produktives tun.

Ein weiterer wünschenswerter Nebeneffekt von einem Open Source Projekt besteht darin, dass die SchülerInnen gezwungen sind, ihren Quelltext übersichtlich zu gestalten, vernünftige Variablenbezeichner zu verwenden und auf eine gute Kommentierung ihres Codes zu achten. Auch erkennen sie durch den obligatorischen Einsatz eines CVS-Systems, dass ein Modularisierung sehr wichtig ist und es vermieden werden sollte, dass zwei Personen an einer Datei programmieren. Auch lernen die SchülerInnen die Mentalität in einem Open Source Projekt kennen und nehmen sogar Kontakt zu einigen Entwicklern auf. Weiterhin ist es sogar möglich, dass externe Entwickler zu dem Projekt hinzustoßen und in die Organisation eingebunden werden müssen. Diese Fälle bergen zwar auch das Risiko, dass die SchülerInnen von der eigentliche Arbeit abgelenkt werden, ansonsten sind sie für die Motivation und für die Lerneffekte nur förderlich. Bei jedem nicht freien Projekt haben die SchülerInnen keine Möglichkeit

einem breiten Publikum ihr noch unfertiges Projekt vorab zu präsentieren. In der Open Source Community ist es hingegen jederzeit möglich und wird auch bei geeigneter Onlinepräsentation angenommen. Auf diese Art lernen die SchülerInnen auch andere Projekte kennen und können sich für ihr eigenes Projekt Anregungen hohlen.

Insgesamt ist das Aufgreifen der Open Source Thematik im Projektsemester bei geeigneten Vorraussetzung sehr zu empfehlen, da hier ein größerer potentielle Lerneffekt zu erwarten ist.

References

[1] K. Coar, “The Open Source Definition”, www.opensource.org/docs/osd, 2006

[2] R. Stallmann, “Die vier Freiheiten eines Software-Benutzers – Ein Interview mit Richard Stallman”, Log In, Nr. 144 (2007), S.27ff

[3] Senatsverwaltung für Bildung, Jugend und Sport Berlin, “Rahmenlehrplan für die gymnasiale Oberstufe”, Berlin: Oktoberdruck AG (2006)

[4] K. Frey, “Die Projektmethode”, Weinheim: Beltz (1990)

[5] C.A. Zehnder, “Informatikprojektentwicklung” Stuttgart: Teubner (1986)

[6] M. Ludwig (Hrsg.), “Projekte im methematisch- naturwissenschafltichen Unterricht”, Würzburg: Franzbecker (2001)

[7] vgl. http://de.wikipedia.org/wiki/Wikipedia bzw. http://de.wikipedia.org/wiki/Open_Source

[8] L. Humbert, “Didaktik der Informatik”, 2., überarb. u. erw. Aufl. Stuttgart: Teubner (2006)

[9] T. Renner, u.a., “Open Source Software”, Fraunhofer-Institut für Arbeitswirtschaft und Organisation, http://www.e-business.iao.fraunhofer.de/docs/fhg_oss-studie.pdf

[10] Hilbert Meier, “Schulpädagogik II: Für Fortgeschrittene”, Berlin (1997)