

---

# Freie Software am Gymnasium

---

Diplomarbeit GYMINF

von  
ADRIAN HÄFLIGER

eingereicht bei  
PROF. RONNY STANDTKE  
BERNER FACHHOCHSCHULE

Luzern, 14. Juli 2023

Wenn nicht ich für mich bin, wer ist dann für mich?  
Solange ich aber nur für mich selber bin, was bin ich?  
Und wenn nicht jetzt, wann sonst?  
Hillel

Diese Arbeit ist veröffentlicht unter einer Creative Commons Attribution-ShareAlike 4.0 International License.



# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einleitung</b>                          | <b>1</b>  |
| <b>2</b> | <b>Freie Software</b>                      | <b>6</b>  |
| 2.1      | Definition freier Software . . . . .       | 7         |
| 2.2      | Kultur freier Software . . . . .           | 16        |
| 2.3      | Einfluss freier Software . . . . .         | 29        |
| <b>3</b> | <b>Freie Software am Gymnasium</b>         | <b>37</b> |
| 3.1      | Doppelte Zielsetzung . . . . .             | 38        |
| 3.2      | Nicht für die Schule lernt man ... . . . . | 43        |
| 3.3      | Wissenschaftspropädeutik . . . . .         | 45        |
| 3.4      | Gesellschaftsreife . . . . .               | 47        |
| 3.5      | Reife Gesellschaft . . . . .               | 49        |
| 3.6      | Digital Commons . . . . .                  | 51        |
| 3.7      | Nachhaltigkeit . . . . .                   | 54        |
| 3.8      | Sapere Aude! . . . . .                     | 57        |
| <b>4</b> | <b>Fazit</b>                               | <b>60</b> |

# Kapitel 1

## Einleitung

Zur Einleitung möchte ich darstellen, wie ich zum Schreiben dieser Arbeit gekommen bin. Ich hoffe, dass dies eine gute Einführung ins Thema der Arbeit bildet.

Ich unterrichte seit dem Jahr 2008 Philosophie und Mathematik an der Kantonsschule Alpenquai. In dieser Zeit wurden schrittweise Computer für Schüler\*innen eingeführt. Zuerst handelte es sich um Pilotversuche mit wenigen Klassen im Obergymnasium, dies wurde dann auf das ganze Obergymnasium ausgeweitet und in den letzten Jahren wurden Geräte im Besitz des Kantons auch an Schüler\*innen der unteren Klassenstufen abgegeben. Ab dem kommenden Schuljahr werden alle 2000 Schüler\*innen mit einem Computer im Unterricht sein. In den Klassen des Obergymnasiums wird fast ausschliesslich digitale Heftführung gemacht mit OneNote, es handelt sich dabei um die weitaus häufigste Verwendung der Geräte im Unterricht. Im Untergymnasium wird keine digitale Heftführung gemacht und die Geräte werden nur punktuell eingesetzt, bisher insgesamt eher wenig.

Das Kollegium hat gespalten auf die Einführung reagiert. Es gab Early-Adapter, welche früh die Chancen für ihren Unterricht erkannten und mit Freude Einsatzszenarien entwickelten. Es gab einige, die der Entwicklung sehr kritisch gegenüberstanden und sich dagegen zur Wehr setzten. Sie fürchteten die Ablenkung der Schüler\*innen und erkannten keinen didaktischen oder pädagogischen Mehrwert im Einsatz von Geräten im Unterricht. Die meisten Lehrpersonen jedoch waren weder begeistert noch vollständig dagegen und haben sich mit der neuen Situation arrangiert.

Und zwischen diesen Lagern stand all die Jahre ich. Ich war immer der Überzeugung, dass es unsere Aufgabe als Schule ist, die Schüler\*innen auf

eine digitalisierte Gesellschaft vorzubereiten und deshalb Computer an der Schule zum Einsatz kommen müssen. Gleichzeitig war ich entsetzt darüber, wie fest kommerzielle Softwareanbieter, namentlich Microsoft, in die Schule vordrangen und wie wenig dagegen unternommen wurde. Es werden von der Schule konkrete Vorgaben darüber gemacht, mit welcher Software gearbeitet werden muss, was wiederum bedingt, mit welchem Betriebssystem gearbeitet wird. Damit bereiten wir die Schüler\*innen ausschliesslich auf die Verwendung von Produkten eines bestimmten Herstellers vor. Am weitesten verbreitet ist die Arbeit mit OneNote. OneNote-Notizbücher können nicht in einem unabhängigen Format exportiert werden. Schüler\*innen, welche auch nach der abgeschlossenen Matura auf ihre Unterlagen zugreifen wollen, sind gezwungen, weiterhin OneNote zu verwenden und werden dadurch über die Schulzeit hinaus an ein Produkt einer Firma gebunden.

Wir tragen damit direkt zur Festigung eines bereits bestehenden Monopols bei. Gleichzeitig begeben wir uns in die Abhängigkeit von kommerziellen Interessen. Wir haben keinen Einfluss auf die Weiterentwicklung der Produkte und können nicht kontrollieren, wie lange diese in dieser Form weiter unterstützt werden. Zudem sind die meisten der verwendeten Produkte nicht für den Einsatz in der Schule entwickelt und deshalb nur bedingt dafür geeignet.

Wir verwenden nicht für den pädagogischen Einsatz gemachte Bürosoftware, deren Entwicklung wir nicht kontrollieren können, schränken unseren pädagogischen Handlungsspielraum ein und bezahlen für all das beträchtliche Lizenzgebühren an einen Anbieter, welchem wir helfen, seine Monopolstellung zu festigen: Es schien mir intuitiv klar, dass dies für eine öffentliche Schule inakzeptabel ist. Mit zunehmender Verwunderung habe ich aber festgestellt, dass ich mit dieser Ansicht beinahe alleine war und meine geäusserten Bedenken fast kein Gehör fanden. Die weitaus häufigste Reaktion auf meine immer wieder formulierten Einwände von Seiten der Schulleitung und der Kolleg\*innen war: *Du hast schon recht, aber es gibt keine Alternativen*. Aber es musste doch Alternativen geben! So habe ich begonnen, mich eingehender mit der Thematik zu befassen.

Ich verfügte über keine vertieften Computerkenntnisse und auch nur wenig Erfahrung mit freier Software. Im Studium und dann für den Mathematikunterricht habe ich intensiv mit LaTeX gearbeitet. Ich wusste also, dass es freie Software gibt und hatte die Erfahrung gemacht, dass diese funktioniert. Ich habe dann begonnen mit Linux zu arbeiten und nach und nach so gut es ging möglichst viel freie Software verwendet. Was zuvor nur eine

Vermutung und Behauptung war, wurde zur gesicherten Erkenntnis: Es gibt funktionierende Alternativen zu proprietärer Software.

Parallel dazu schritt die Einführung von Computern an der Schule in grossen Schritten voran und die Anbindung an Microsoft-Produkte wurde weiter intensiviert. Weiterhin brachte ich meine Einwände vor und durch meine Erfahrungen mit freier Software konnte ich auch Alternativen vorschlagen. Aber die Reaktionen blieben identisch: *Du hast schon recht, aber es gibt keine Alternativen.* Mehr und mehr kam als Reaktion hinzu: *Dafür ist es nun zu spät.* Wir waren bereits so fest eingebunden in das Microsoft-System, dass ein Aus- und Umstieg nicht mehr vorstellbar war. Dieses Eingeständnis allein hätte nach meinem Verständnis den dringenden Anlass liefern müssen, so schnell wie möglich mit der Reduktion dieser Abhängigkeiten zu beginnen. Stattdessen wurde das Argument der Alternativlosigkeit und des verpassten Ausstiegstermins wie ein Mantra wiederholt. Da es für mich weiterhin offensichtlich schien, dass die Abhängigkeit von kommerziellen Interessen für eine Schule nicht zulässig ist, konnte ich diese Argumente nicht akzeptieren. Nur sah ich nicht, wie ich meinen Einwänden mehr Gehör verschaffen konnte.

In diese Zeit fiel die Ankündigung, dass das Obligatorische Fach Informatik auf das Schuljahr 21/22 eingeführt würde und dass Lehrpersonen die Möglichkeit hätten, den GYMINF-Studiengang zu absolvieren und sich zur Informatiklehrperson zu qualifizieren. Ich sah in diesem Angebot eine doppelte Chance für meinen Einsatz gegen proprietäre Software an der Schule: Als ausgebildete Informatiklehrperson hätten meine Einwände und Vorschläge mehr Glaubwürdigkeit und als Informatiklehrperson könnte ich direkteren Einfluss auf den Einsatz von Geräten an der Schule haben. Deshalb habe ich den GYMINF-Studiengang begonnen und mit viel Freude und grossem Gewinn absolviert.

In der Zwischenzeit unterrichtete ich seit drei Jahren Informatik, mittlerweile ausschliesslich. In meinem Unterricht kann ich sehr direkt dazu beitragen, dass die Schüler\*innen mehr Informatikkenntnisse haben und ich kann in den Bereichen, welche nur den Informatikunterricht betreffen, mit freier Software arbeiten. Und ich stelle auch fest, dass meine Argumente teilweise mehr Gehör finden. Aber gleichzeitig ist die Abhängigkeit von Microsoft ungebrochen sehr gross und Alternativen sind schwer vorstellbar. Um wirklich eine Änderung möglich zu machen, braucht es gezieltes und strategisches Vorgehen.

Die Bedenken gegen proprietäre Software zu formulieren, war offensichtlich nicht zielführend. Das Problem ist gleichzeitig zu gross und zu wenig

spürbar. Die Abhängigkeit von Softwaremonopolisten ist allgegenwärtig in unserem Leben, nicht nur an der Schule. Und ich bin überzeugt, dass dies von den meisten Lehrpersonen als Problem betrachtet wird. Es ist aber ein Problem einer Dimension, welches nicht konkret angegangen werden kann. Und gleichzeitig ist das Problem wenig spürbar. Die wenigsten haben bisher konkrete negative Erfahrungen gemacht im Zusammenhang mit der Abhängigkeit von Softwaremonopolisten. In einer solchen Situation kann eine Ohnmacht entstehen, welche zur Handlungsunfähigkeit führt.

Die Situation ist vergleichbar mit dem Klimawandel. Die allermeisten Menschen glauben, dass der Klimawandel eine Tatsache ist und dass die Auswirkungen davon schwerwiegend und schlimm sein werden. Und die meisten Menschen teilen auch die Einschätzung, dass etwas gegen den Klimawandel unternommen werden muss. Aber die Auswirkungen sind zu wenig konkret spürbar, gleichzeitig sind die Möglichkeiten für effektive individuelle Handlungen begrenzt. Angesichts des Klimawandels fühlen sich deshalb viele ohnmächtig und handlungsunfähig.

Beim Klimawandel zeigt sich, dass negative und warnende Stimmen wenig Gehör finden. Es ist einfach, diese zu verdrängen. Das Problem wird zwar anerkannt, aber dies führt nicht dazu, dass eine Notwendigkeit zum Handeln entsteht. Genauso verhält es sich angesichts der Macht von Big-Tech-Unternehmen, welche unser Leben stark beeinflussen und eine grosse Gefahr für die Freiheit darstellen. In einer solchen Situation braucht es eine andere Strategie, als auf die Probleme hinzuweisen und die negativen Folgen zu benennen.

Das Problem war: Das Aufmerksammachen auf Probleme bei gleichzeitig nicht sichtbaren Alternativen führt zu Ohnmacht und Abwehr. Ich bin zum Schluss gekommen, dass die Strategie stattdessen darin bestehen sollte, dass wir einerseits die positiven Aspekte freier Software, insbesondere im Zusammenhang mit Bildung, hervorheben und gleichzeitig positive Erfahrungen mit freier Software ermöglichen müssen.

Für den zweiten Pfeiler sehe ich eine konkrete Chance in der Einführung von Moodle. Im Kanton Luzern wird Moodle ab diesem Schuljahr schrittweise eingeführt, einerseits als Prüfungssoftware, mehr und mehr auch als LMS. Moodle ist eines der erfolgreichsten Projekte freier Software für den Bildungsbereich. Es bietet vielfältige Funktionalitäten für viele der im Bildungszusammenhang benötigten Aufgaben. Moodle bietet zudem die Möglichkeit, Ausgangspunkt verschiedener anderer Anwendungen zu sein. Da ich in Moodle eine Möglichkeit sehe, dass möglichst viele Lehrpersonen und Schüler\*innen

positive Erfahrungen mit freier Software machen können, bin ich auf kantonaler und schulischer Ebene dabei, die Einführung von Moodle mitzugestalten. In dieser Arbeit gehe ich aber auf Moodle und insgesamt diesen ersten Pfeiler nicht ein. Fokus der Arbeit ist der andere Pfeiler: Die positiven Aspekte freier Software für die Bildung zusammenzutragen ist Ziel dieser Arbeit.

Gerade in den durch das Aufkommen von Chatbots befeuerten breit und kontrovers geführten Diskussion um die Bedeutung von KI für die Bildung eröffnen sich möglicherweise neue Diskursräume, bei denen auch die Rolle von freier Software in den Fokus geraten könnte. Dass KI zukünftig eine wichtige Rolle spielen wird, kann heute nicht mehr bezweifelt werden. Genauso wenig scheint es wünschbar zu sein, dass die Schule die Schüler\*innen nicht auf den Umgang mit KI vorbereitet. Aber ein entscheidender Faktor wird sein, wer die Algorithmen und die Modelle, auf welchen sie arbeiten, kontrolliert. Diese Frage zu stellen, bietet die Chance, allgemeiner die Frage ins Zentrum zu rücken, wer die Technologie, welche unsere Gesellschaft prägt und welche wir auch an der Schule verwenden, kontrolliert. Und es bietet sich die Gelegenheit, das Argument der Alternativlosigkeit und des verpassten Zeitpunkts nicht gelten zu lassen und ins Handeln zu kommen. Frei nach einem jüdischen Sprichwort möchte ich deshalb fragen: Wenn nicht wir, wer? Wenn nicht jetzt, wann?



# Kapitel 2

## Freie Software

Der erste Teil der Arbeit ist eine Darstellung von freier Software in drei Kapiteln. Im ersten Kapitel wird definiert, was freie Software ist. Dazu werden die für das Verständnis der Definition notwendigen technischen und juristischen Aspekte und Begriffe erläutert. Diese Definition genügt aber nicht, um das Phänomen freier Software adäquat zu beschreiben. Im zweiten Kapitel wird die Kultur freier Software beschrieben. Es wird aufgezeigt, dass es um mehr als Technologie und Lizenzen geht. Diese Kultur hat sich nicht nur immerfort weiterentwickelt, sondern auch darüberhinaus weitere Bewegungen beeinflusst. Dies wird im dritten Kapitel dargestellt.

Für diesen ersten Teil wurde eine Fülle von Literatur über freie Software und die verschiedenen Aspekte freier Software bearbeitet. Sehr wichtig dafür waren dabei die Bücher von zwei Anthropolog\*innen: *Coding Freedom. The Ethics and Aesthetics of Hacking* von Gabriella Coleman sowie *Two Bits: The Cultural Significance of Free Software* von Michael Kelty. Diese beiden Bücher bieten einerseits eine umfassende Darstellung der Geschichte freier Software, sowie Ansätze für eine kulturwissenschaftliche Auseinandersetzung damit. Ebenfalls wichtige Quellen waren Bücher von zwei Juristen: *Code 2.0* von Lawrence Lessig und *The Wealth of Networks* von Yochai Benkler. Unter anderem diese Bücher haben dazu beigetragen, dass ich verstanden habe, dass es sich bei freier Software um mehr als nur eine technologische und juristische Angelegenheit handelt - und dass der Einsatz von freier Software gerade deswegen einen grossen Mehrwert für die Bildung bietet.

## 2.1 Definition freier Software

Freie Software ist Software, welche unter einer GNU General Public License<sup>1</sup> oder einer dazu analogen<sup>2</sup> Lizenz veröffentlicht wird. Freie Software enthält an prominenter Stelle die folgende Notiz:

```
This program is free software: you can redistri-
bute it and/or modify it under the terms of the
GNU General Public License as published by the
Free Software Foundation, either version 3 of the
License, or (at your option) any later version.
```

Eine freie Lizenz garantiert, dass die Software verbreitet und verändert werden kann und hält fest, dass Software, welche durch Veränderung daraus hervorgegangen ist, ebenfalls freie Software sein muss (und also ebenfalls unter einer solchen Lizenz veröffentlicht werden muss). Durch eine freie Lizenz werden dem\*der Besitzer\*in der Software die exakt gleichen Rechte zugesprochen, wie sie der\*die ursprüngliche Autor\*in hat.

Freie Software garantiert ihren Benutzer\*innen vier grundlegende Freiheiten: Die Freiheit, Software zu verwenden, zu verstehen, zu verändern und zu verbreiten. Die Motivation für diese Praxis entstammt der Überzeugung, dass Benutzer\*innen ein Recht auf die vollständige Kontrolle der von ihnen verwendeten Software haben und dass dies nur dann der Fall ist, wenn Sie über diese vier Freiheiten verfügen. Da Software in jedem Fall von jemandem kontrolliert wird, wird sie von jemand anderem als dem\*der Benutzer\*in kontrolliert, wenn diese Freiheiten nicht gegeben sind. In einer Umwelt, welche in zunehmendem Mass digitalisiert und somit durch Software bestimmt ist, bedeutet dies, dass Individuen in entscheidenden Bereichen ihres Lebens nicht frei sind. Sie haben keine Möglichkeit zu verstehen, wie das funktioniert, was sie verwenden und somit auch keine Kontrolle darüber. Richard Stallman, der ursprüngliche Verfasser der GNU General Public License, beschreibt dies so:

---

<sup>1</sup>In dieser Arbeit wird immer auf die aktuellste Version GNU General Public License Version 3 Bezug genommen: <https://www.gnu.org/licenses/gpl-3.0.html>.

<sup>2</sup>Ob und welche Lizenzen dazu wirklich analog sind, ist Gegenstand von Kontroversen und kann an dieser Stelle nicht entschieden werden. Da die GNU General Public License unbestrittenermassen die strengste Lizenz für freie Software ist, nehme ich konkret nur auf diese Bezug, die Aussagen dieser Arbeit können aber je nach Kontext auch für andere Lizenzen Gültigkeit haben, wenn sie in dieser Hinsicht zur GNU GPL analog sind.

Eigentlich geht es darum, den Benutzern Freiheit zu geben, indem ihnen freie Software zur Verfügung gestellt wird und die Grenzen dessen, was man mit freier Software tun kann, möglichst stark zu dehnen. Schliesslich besteht die Idee von GNU<sup>3</sup> darin, den Leuten die Arbeit mit ihren Computern zu ermöglichen, ohne dass sie die Vorherrschaft von jemand anderem anerkennen müssen. Ohne dass irgendein Softwareeigentümer sagen kann: “Ich lasse nicht zu, dass Sie verstehen, wie das funktioniert. Ich werde dafür sorgen, dass Sie hilflos und von mir abhängig bleiben. [...]”<sup>4</sup>

Computerprogramme werden in einer Programmiersprache geschrieben. Ein Computerprogramm in Textform dargestellt heisst Sourcecode. Um die Funktionsweise eines Programms zu verstehen, muss der Sourcecode studiert werden. Veränderungen am Programm können nur im Sourcecode vorgenommen werden. Damit ein Programm von einem Computer ausgeführt werden kann, muss es in eine von Maschinen lesbare Sprache umgewandelt werden. Dazu wird der Programmcode kompiliert. Je nach Programmiersprache ist es aus dieser kompilierten Version nicht (oder nur unter enorm grossem Aufwand) möglich, den Sourcecode zurückzugewinnen. Um ein Programm verwenden zu können, genügt es, die kompilierte Version zu besitzen, da dieses vom Computer gelesen und ausgeführt werden kann. Es kann dann aber weder die Funktionsweise studiert noch können Änderungen am Programm vorgenommen werden. Aus der Forderung, dass Software studiert und verändert werden kann, folgt die Forderung, dass ihr Sourcecode veröffentlicht werden muss.<sup>5</sup>

Das Gegenteil von freier Software ist proprietäre Software. Proprietäre Software wird unter einer exklusiven Copyright-Lizenz veröffentlicht, welche die Weitergabe und Veränderung des Codes verbietet. Bei Zuwiderhandlung gegen diese Lizenzbestimmungen drohen strafrechtliche Konsequenzen. Obwohl dies nicht strikt zur Definition von proprietärer Software gehört, wird

---

<sup>3</sup>GNU ist das ursprünglich von Stallman geschriebene Betriebssystem, für welches die GNU General Public License entwickelt wurde. In Kombination mit Linux ist GNU/Linux bis heute das am weitesten verbreitete freie Betriebssystem.

<sup>4</sup>Stallman wird hier zitiert von Moody: Moody, *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*, S. 45.

<sup>5</sup>Dass der Sourcecode zusammen mit dem Programm verbreitet wird ist der Normalfall, die GNU General Public License beschreibt unter Punkt 6 aber auch andere Verbreitungsformen, welche jedoch allesamt eine Möglichkeit vorsehen, den Sourcecode zu erhalten.

diese in den meisten Fällen ohne den Sourcecode vertrieben. Was folgerichtig ist, da für den Sourcecode keine Verwendung besteht, wenn diese weder studiert noch verändert werden können soll.

Software mit exklusiven Copyright-Lizenzen zu schützen ist ein relativ junges Verfahren, dessen Entstehung eng mit Entwicklungen in der Produktion von Computern verknüpft ist. Nachdem in der Frühphase der Entwicklung von Computern diese für jeweils eine spezifische Tätigkeit konzipiert waren und die Algorithmen zur Erfüllung dieser Aufgaben entweder direkt in der Hardware codiert waren oder die dazu geschriebenen Programme auf spezifische Hardwareeigenschaften Bezug nahmen, wurde ab den 1970-er Jahren Software zunehmend unabhängiger von der Hardware. Solange Software an spezifische Hardware geknüpft war, gab es für die Firmen, welche diese herstellten, keinen Grund, diese zu schützen, da sie für Anbieter von anderen Geräten von keinem Nutzen sein würden. Sobald die Software von der Hardware unabhängig ist, kann diese sehr einfach kopiert und für die Verwendung mit einer konkreten Hardware angepasst werden. Nachdem in einer ersten Phase der relativen Unabhängigkeit von Software deren Produzenten ihre Konkurrenz sogar dazu ermunterten, diese zu verwenden und bearbeiten, da sie sich davon erhofften, dass ihre Systeme auch auf anderen Geräten funktionieren würden,<sup>6</sup> gab es ab Mitte der 1970-er Jahre von Seiten der Softwareindustrie sehr starke Bemühungen, ihre Produkte zu schützen.

Dabei war es zu Beginn noch unklar, wie und mit welchen rechtlichen Mitteln Software geschützt werden konnte. Denn nicht nur im Zusammenhang mit Computern, sondern insgesamt für technische Entwicklungen war die entstandene Situation neuartig. Zwar bestand auch in der Zeit vor Computern das Bedürfnis, Entwicklungen zu schützen. Aber weil Software letztlich nur Text war und deshalb einfach kopiert und verändert werden konnte, stellte sich das Problem in einer akzentuierten Form. Zwar gab es technische Möglichkeiten, den Code zu verstecken, so dass beim Verkauf von Software lediglich kompilierte Programme weitergegeben wurden. Aber dies war nicht in jeder Hinsicht befriedigend, denn es war auch wichtig, offene Standards zu haben, so dass verschiedene Hersteller Produkte aufeinander abstimmen konnten, was wiederum eine Offenheit des Codes zumindest zu einem gewissen Grad erforderte.<sup>7</sup>

---

<sup>6</sup>Schoonmaker, *Free Software, The Internet, and Global Communities of Resistance*, S. 20.

<sup>7</sup>Für eine übersichtliche Geschichte im Umgang mit dieser Situation anhand der Betriebssysteme aus der Unix-Familie vgl. Kelty, *Two Bits: The Cultural Significance of Free*

Es gibt verschiedene juristische Verfahren, um Erfindungen zu schützen: Copyright-Lizenzen, Patente und der Schutz von Geschäftsgeheimnissen. Alle Verfahren können und werden für den Schutz von Software angewendet, wobei die weitaus häufigste Verwendung Copyright-Lizenzen sind.<sup>8</sup> Copyright schützt eine kreative Arbeit und gibt seinem\*seiner Benutzer\*in das exklusive Recht, diese zu kopieren, veröffentlichen und zu verändern. Im Gegensatz zu einem Patent, welches eine Idee schützt, wird durch eine Copyright-Lizenz lediglich der Ausdruck dieser Idee geschützt. Die Gültigkeit von Copyright-Lizenzen wird durch nationales Gesetz geregelt, in der Schweiz durch das Urheberrechtsgesetz.<sup>9</sup>

Die technologische Entwicklung und die daraus resultierenden neuartigen Bedürfnisse nach Schutz von Erfindungen widerspiegelt sich auch in der Geschichte der Copyright-Lizenzen, welche zunehmend restriktiver formuliert wurden. Seit dem 18. Jahrhundert gab es eine stetige Entwicklung hin zu restriktiveren Anwendungen von Copyright zum Schutz von geistigem Eigentum. Bis zum 18. Jahrhundert war lediglich die Veröffentlichung eines geschützten kommerziellen Werkes verboten, die Veränderung jedoch war erlaubt und für nicht-kommerzielle Werke gab es überhaupt keine Einschränkungen, so dass diese frei veröffentlicht und verändert werden durften. Bis zum Ende des 19. Jahrhunderts war auch die Veränderung von kommerziellen Werken durch Copyright-Lizenzen verboten. Und bis 1975 galt dies auch teilweise für die Veröffentlichung von nicht-kommerziellen Werken.<sup>10 11 12 13</sup> Die zunehmende Wichtigkeit von Software und die damit einhergehende Macht der Softwareindustrie mit dem Bedürfnis nach Schutz ihrer Entwicklungen förderte gegen Ende des 20. Jahrhundert Bestrebungen zur weiteren Verschärfung des Copyright-Gesetzes, welche in den USA 1998 im Digital Millennium Copyright Act (DMCA) gipfelten.<sup>14</sup> Der DMCA kriminalisierte jegliche andere Verwendung von proprietärer Software über ihre reine An-

---

*Software*, S. 119 - 142.

<sup>8</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 199 - 200.

<sup>9</sup>[https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1993/1798\\_1798\\_1798/20200401/de/pdf-a/fedlex-data-admin-ch-eli-cc-1993-1798\\_1798\\_1798-20200401-de-pdf-a.pdf](https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1993/1798_1798_1798/20200401/de/pdf-a/fedlex-data-admin-ch-eli-cc-1993-1798_1798_1798-20200401-de-pdf-a.pdf), aufgerufen am 11. Juli 2023.

<sup>10</sup>Lessig, *Free Culture*, S. 170 - 173.

<sup>11</sup>Vaidhyanathan, *Intellectual Property. A Very Short Introduction*.

<sup>12</sup>Thalheim, „Trusted Computing“.

<sup>13</sup>Benkler, *The Wealth of Networks*, S. 383 - 459.

<sup>14</sup>[https://web.archive.org/web/20000824110859/http://www.eff.org/IP/DMCA/hr2281\\_dmca\\_law\\_19981020\\_pl105-304.html](https://web.archive.org/web/20000824110859/http://www.eff.org/IP/DMCA/hr2281_dmca_law_19981020_pl105-304.html), aufgerufen am 11. Juli 2023.

wendung hinaus.<sup>15 16 17 18</sup>

Es mag auf den ersten Blick überraschen, dass sich eine Bewegung, welche sich der Freiheit verschrieben hat, in erster Linie über eine juristische Lizenz definiert. Um dies einordnen zu können, muss verstanden werden, dass das Gegenteil von Software, welche durch Copyright geschützt ist, nicht Software ist, für welche kein Copyright besteht. Denn Software ohne Copyright-Lizenz kann auf jede Art verwendet werden, unter anderem eben auch so, dass sie unter eine exklusive Copyright-Lizenz gestellt wird. Entwickler\*innen freier Software, deren Anliegen die freie Verwendung und Verfügbarkeit von Software ist, wollen dies verhindern. Innerhalb der Community von interessierten Softwareentwickler\*innen gab es zwar bereits eine gelebte Kultur vom Teilen von Code und kollaborativen Arbeiten an Software, diese war aber nicht rechtlich geschützt und führte entsprechend immer wieder zu Konflikten.<sup>19</sup> Gesucht war eine Möglichkeit, Sourcecode zu veröffentlichen und zu jeglicher Verwendung frei zu geben, diesen aber vor dem Schutz durch eine exklusive Copyright-Lizenz zu schützen. In einem längeren Prozess entwickelte sich ein Verfahren, bei welchem genau dies durch die Verwendung von Copyright-Lizenzen selbst gelöst werden konnte.<sup>20</sup>

Dabei wird in der Copyright-Lizenz explizit festgehalten, dass die Software ohne weitere Bestimmungen verwendet und weitergegeben und verändert werden darf, dass aber in jedem Fall die Software unter einer solchen Lizenz erhalten bleiben muss. Weil diese Copyright-Lizenzen dazu dienen, das Copyright abzugeben und zu garantieren, dass es auch in Derivaten abgegeben wird, werden diese umgangssprachlich auch als Copyleft-Lizenzen bezeichnet.<sup>21</sup> Diese Verwendung entspricht nicht dem ursprünglichen Zweck von

---

<sup>15</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 9 - 10, 64 - 88, 201.

<sup>16</sup>Schoonmaker, *Free Software, The Internet, and Global Communities of Resistance*, S. 22.

<sup>17</sup>Webb, *Coding Democracy*, S. 159.

<sup>18</sup>Vaidhyanathan, *Copyrights and Copywrongs: The Rise of Intellectual Property and How It Threatens Creativity*.

<sup>19</sup>Moody, *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*, S. 30.

<sup>20</sup>Vgl. Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 179 - 209.

<sup>21</sup>Dabei wird mit der Ambiguität des englischen Begriffs *left* gespielt, was sowohl *links* bedeutet und somit das Gegenteil von *rechts/right* ist, aber auch das Partizip von *leave/lassen* ist und somit auf den Akt des Loslassens hinweist, welches mit dieser Art von Lizenz vollzogen wird. Ein anderes Wortspiel mit Copyright ist *Copyright - Copywrong*, welches von Kritikern von der Verwendung von Copyright verwendet wird.

Copyright-Lizenzen, ist aber innerhalb ihres rechtlichen Rahmens möglich.

Die Möglichkeit von freier Software basiert somit auf der exakt gleichen gesetzlichen Grundlage, auf welche sich Produzent\*innen von proprietärer Software zum Schutz ihrer Entwicklungen verlassen und macht sich zunutze, dass die Copyright-Gesetze sehr rigide formuliert sind.<sup>22</sup> Kelty bezeichnet deshalb die Entwicklung von Copyleft-Lizenzen als einen Hack - also eine originelle und konstruktive Zweckentfremdung eines Mittels:

Stallman's GNU General Public License "hacks" the federal copyright-law right law [...]. It does this by taking advantage of the very strong rights granted by federal law to actually *loosen* the restrictions normally associated with ownership.<sup>23</sup>

Und Coleman beschreibt diese kreative Umnutzung der Copyright-Lizenzen als einen Kurzschluss:

The freedom to labor within a framework of their own making is enabled by licenses that cleverly reformat copyright law to prioritize access, distribution, and circulation. Thus, hackers short-circuit the traditional uses of copyright: the right to exclude and control.<sup>24</sup>

Ein prominentes Beispiel für den Erfolg des Copyleft-Ansatzes ist Linux. Das Betriebssystem GNU war das erste freie Betriebssystem, welches zwar von der Architektur her dem proprietären Unix-System ähnlich war, aber von Richard Stallman ohne Verwendung einer Zeile Unix-Code geschrieben wurde. Entsprechend lagen die Eigentumsrechte allein bei Richard Stallman und er entschied sich, dieses unter eine freie Lizenz zu stellen, welche er speziell zu diesem Zweck entwickelte (deshalb bis heute der Name: GNU General Public License). Was GNU fehlte war ein Kernel, jener Teil eines Betriebssystems, welcher direkt das Zusammenspiel zwischen Hard- und Software regelt. 1991 entwickelte der finnische Informatikstudent Linus Torvalds einen Kernel und nannte diesen Linux. Dieser war ebenfalls ohne Verwendung einer fremden Zeile Code verfasst und somit lagen die Eigentumsrechte und entsprechend auch die Wahl einer Lizenz bei Torvalds. Torvalds merkte, dass der Linux-Kernel erst im Zusammenspiel mit GNU seinen Nutzen voll entfalten konnte

---

<sup>22</sup>Lessig, *Free Culture*, S. 265.

<sup>23</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 182.

<sup>24</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 1.

und wollte ihn deshalb in GNU integrieren. Da GNU unter einer freien Lizenz veröffentlicht war, war dies erlaubt - allerdings nur unter der Bedingung, dass das daraus entstandene Konglomerat GNU/Linux, und damit auch Linux selbst, ebenfalls freie Software wurde. So entschied sich Torvalds, der keine eigentlichen Präferenzen bezüglich Lizenzen hatte und sich wenig um rechtliche und allgemein um über das technische hinausgehende Aspekte kümmerte, Linux unter der GNU General Public License zu veröffentlichen.<sup>25 26 27</sup>

Dass Software frei ist, bedeutet nicht, dass sie gratis sein muss. Das englische Adjektiv *free* ist ambig und kann *gratis* wie auch *frei* bedeuten. Freie Software ist frei, aber nicht zwingend gratis. Um die Unterscheidung zu verdeutlichen wird gesagt *free as in free speech, and not as in free beer*.<sup>28</sup> Software verkaufen zu können ist eine weitere Freiheit, welche mit freier Software einhergeht. Solange Software, welche unter einer freien Softwarelizenz erworben (für Geld oder gratis) und allenfalls verändert worden ist, wiederum unter einer freien Lizenz steht, darf diese zu einem beliebigen Preis verkauft werden oder für eine mit der Software zusammenhängende Dienstleistung Profit erzielt werden.<sup>29</sup>

Software kann frei sein, auch wenn sie nicht gratis ist. Umgekehrt ist es nicht der Fall, dass Software frei ist, *weil* sie gratis verfügbar ist. Immer mehr Software wird gratis vertrieben, dies gilt insbesondere für Smartphone-Apps.<sup>30</sup> Dabei ist es entscheidend zu verstehen, dass die Tatsache, dass etwas gratis ist, dies nicht zu etwas Freiem macht - eher das Gegenteil ist der Fall. Grundsätzlich haben wir Freiheit über Software nur, wenn die Möglichkeit besteht, diese zu kontrollieren. Ist dies nicht der Fall, so sind wir nicht frei, wenn wir die Software verwenden. Und konkret müssen wir uns fragen, wieso ein Produkt, welches Resultat von investierter Zeit und Ressourcen ist, gratis zur Verfügung gestellt wird. Wir müssen davon ausgehen, dass in den meisten

---

<sup>25</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 74.

<sup>26</sup>Moody, *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*, S. 42 - 44, 70, 110.

<sup>27</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 66, 207.

<sup>28</sup><https://www.gnu.org/philosophy/free-sw.html>, aufgerufen am 29. April 2023.

<sup>29</sup>Die Free Software Foundation ermutigt sogar ausdrücklich das Erzielen von Profit mit freier Software, weil sie darin eine Möglichkeit erkennt, Ressourcen für die Weiterentwicklung freier Software zu generieren: <https://www.gnu.org/philosophy/selling.html>, aufgerufen am 29. April 2023.

<sup>30</sup>Sowohl im App-Store Google-Play für Android wie auch im App-Store von Apple sind ungefähr 95 % aller angebotenen Apps gratis: <https://www.statista.com/statistics/263797/number-of-applications-for-mobile-phones/>, aufgerufen am 29. April 2023.



Fällen damit ein Gewinn erwirtschaftet wird und somit der\*die Benutzer\*in einen Preis dafür bezahlt. Und wenn dieser nicht direkt in monetärer Form bezahlt werden muss, muss es eine andere Form von Preis sein. In den meisten Fällen ist dieser Preis nicht explizit. Und weil es sich um nicht-freie Software handelt, haben wir keine Möglichkeit herauszufinden, worin dieser Preis besteht. Oftmals ist also gerade gratis verfügbare Software nicht frei und schränkt die Freiheit des\*der User\*in ein.

Neben dem Begriff *freie Software* wird oftmals der Begriff *Open Source* verwendet. Formal gesehen handelt es sich um Open-Source-Software, wenn diese unter einer Open-Source-Lizenz<sup>31</sup> veröffentlicht wird, wobei auch die freien Lizenzen darunter fallen. Formal gesehen ist also freie Software immer auch Open-Source-Software, aber umgekehrt ist nicht alle Open-Source-Software freie Software. Sowohl freie Software wie auch Open-Source-Software grenzen sich ab von proprietärer Software, insofern bei beiden der Sourcecode zugänglich sein muss. Während in der Praxis oftmals kein Unterschied zwischen freier Software und Open-Source-Software besteht, sowohl was deren Verwendung als auch deren Entwicklung betrifft,<sup>32</sup> so verrät doch die Wahl der einen oder der anderen Bezeichnung etwas über die Motive, mit welchen nicht-proprietäre Software gewählt wird. Während es bei freier Software in erster Linie um Freiheit geht, geht es bei Open-Source-Software in erster Linie um die Qualität der Software. Und so können die Differenzen letztlich doch beträchtlich sein.<sup>33 34</sup>

Die Differenzierung zwischen den beiden Bezeichnungen entstand in den 1990-er Jahren. Zuvor gab es lediglich den Begriff der freien Software, welcher im wesentlichen durch die Free Software Foundation geprägt und propagiert wurde. Mit dem wachsenden Erfolg von freier Software, insbesondere durch GNU/Linux/Debian und Apache, wurden immer breitere Kreise auf freie Software aufmerksam. Offenbar war es möglich, grosse Softwareprojekte ohne die hierarchische Struktur und ohne die finanziellen Ressourcen von grossen Unternehmen erfolgreich zu gestalten. Freie Software wurde von der proprietären Softwareindustrie zunehmend nicht nur als Konkurrenz und Gefahr wahrgenommen,<sup>35</sup> sondern als Quelle der Inspiration. Die proprietären

---

<sup>31</sup><https://opensource.org/licenses/>.

<sup>32</sup>Vgl. Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 112 - 115.

<sup>33</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 79 - 88.

<sup>34</sup>Webb, *Coding Democracy*, S. 25.

<sup>35</sup>Moody, *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*, S. 11 - 12.

Softwarehersteller begannen sich für die Arbeitsprozesse und Entwicklungsverfahren zu interessieren, welche zum Erfolg freier Software beigetragen hatten. Als erster fasste Eric S. Raymond 1999 in *The Cathedral and the Bazaar* die Erkenntnisse zusammen, welche aus der Entwicklung freier Software gewonnen wurde. Für Raymond waren die wichtigsten Punkte eine effiziente Organisation der Arbeit durch flache Hierarchien und die grössere Chance, Fehler zu finden, wenn viele Menschen den Code untersuchen können. Letzteres fasste er im berühmt gewordenen Slogan *Given enough eyeballs, all bugs are shallow*<sup>36</sup> zusammen. Diese offensichtlichen Vorteile von freier Software standen jedoch den idealistischen Ansätzen gegenüber, welche nicht mit der Welt der grossen Softwarefirmen kompatibel schienen. Um die Arbeitsweisen freier Software für die Industrie anschlussfähig zu machen, wurde deshalb nach einer alternativen Bezeichnung gesucht. So entstand der Begriff *Open Source*.<sup>37 38 39</sup>

Der Begriff *Open Source* wurde also als Gegenbegriff gegen *freie Software* eingeführt, mit dem Ziel, die möglicherweise abschreckende Wirkung der Konnotationen von *frei* auf kommerzielle Akteure zu vermeiden und stattdessen einen Begriff zu verwenden, der ins Zentrum rückte, was diese Art der Software auch für Unternehmen interessant machte.<sup>40</sup> Aus Sicht der Advokaten der freien Software handelt es sich um ein gefährliches Reframing, die für die kommerzielle Anwendung vorteilhaften Aspekte zu nehmen, ohne sich auf die diesen Interessen zuwiderlaufenden Ideale zu verpflichten. Es ist nicht überraschend, dass explizit vor allem Vertreter der Idee von freier Software diese von Open Source abgrenzen, da diese restriktiver ist und sie in nur Open-Source-Projekten den für sie zentralen Aspekt der Freiheit nicht verwirklicht sehen.<sup>41</sup> Demgegenüber verwenden die grossen proprietären Softwareproduzenten allesamt den Begriff *Open Source*, wenn sie an entsprechenden Projekten beteiligt sind.<sup>42 43 44 45</sup>

---

<sup>36</sup>Raymond, *The Cathedral and the Bazaar*, S. 30.

<sup>37</sup>Raymond, *The Cathedral and the Bazaar*, S. 11.

<sup>38</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 209.

<sup>39</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 178, 208.

<sup>40</sup>Vgl. Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 79.

<sup>41</sup><https://www.gnu.org/philosophy/open-source-misses-the-point.html>.

<sup>42</sup><https://www.microsoft.com/en-us/legal/intellectualproperty/open-source>, aufgerufen am 14. Juli 2023.

<sup>43</sup><https://opensource.apple.com/>, aufgerufen am 14. Juli 2023.

<sup>44</sup><https://opensource.google/>, aufgerufen am 14. Juli 2023.

<sup>45</sup><https://opensource.fb.com/>, aufgerufen am 14. Juli 2023.

## 2.2 Kultur freier Software

Freie Software ist per Definition Software, welche unter einer freien Lizenz veröffentlicht ist. Diese Definition ist jedoch zu einseitig und eng, wenn es darum geht, dem Phänomen freier Software auf die Spur zu kommen und dieses zu untersuchen. Es handelt sich um eine Kultur und muss deshalb auch so beschrieben werden. Dies gilt insbesondere für den Rahmen dieser Arbeit. Die Argumentation für die Verwendung freier Software an der Schule wird stärker, wenn dieser eine Charakterisierung von freier Software zugrundeliegt, welche neben juristischen und technologischen auch weitere Aspekte mitberücksichtigt.

Bei freier Software handelt es sich um eine Bewegung. Für Christopher Kelty wird dies dadurch deutlich, dass sich die Individuen, welche freie Software entwickeln und verwenden, als zugehörig zu einer Gruppe verstehen. Kelty verwendet als Bezeichnung für die Mitglieder dieser Gruppe den Namen “geeks”.<sup>46</sup> Und für die besondere Form der Bewegung prägt er den Begriff einer *rekursiven Öffentlichkeit* (recursive public).

Eine Öffentlichkeit<sup>47 48</sup> ist verstanden als ein Raum (im übertragenen wie auch im wörtlichen Sinn), in welchem Individuen, welche einen bestimmten (möglicherweise auch sehr minimalen) Katalog von Werten miteinander teilen, gemeinsam handeln. Der Katalog von geteilten Werten ist Bedingung der Möglichkeit gemeinsamen Handelns und gleichzeitig konstituierendes Element einer Öffentlichkeit. Über den geteilten Katalog von Werten hinaus bedarf eine Öffentlichkeit keiner weiteren strukturierenden Organisation. Es gehört zum Wesen einer Öffentlichkeit, dass sie wandelbar und flexibel ist. Der Begriff *Öffentlichkeit* grenzt sich dadurch ab von Begriffen wie *Staat*, *Institution* oder *Verein*. Hingegen können neben natürlichen Personen auch juristische Personen und staatliche oder nicht-staatliche Organisationen Teilnehmende einer Öffentlichkeit sein. Eine Öffentlichkeit kann innerhalb eines geographisch oder politisch begrenzten Raums bestehen, sie kann aber auch global existieren, insofern die Möglichkeit der Begegnung gegeben ist. Hingegen ist es nicht zwingend, dass alle Menschen, welche beispielsweise innerhalb eines Staatsgebiets leben, Teil einer gemeinsamen Öffentlichkeit sind. Verschiedene Öffentlichkeiten können nebeneinander bestehen und ein einzelnes

---

<sup>46</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 34 - 36.

<sup>47</sup>Dewey, *The Public and Its Problems*.

<sup>48</sup>Habermas, *Strukturwandel der Öffentlichkeit. Untersuchungen zu einer Kategorie der bürgerlichen Gesellschaft*.

Individuum kann Teil mehrerer Öffentlichkeiten sein.

Das spezifische Charakteristikum einer rekursiven Öffentlichkeit ist, dass ihr konstituierendes Element, also das, was die Mitglieder dieser Öffentlichkeit zusammenführt und was sie teilen, das Schaffen der Bedingungen dieser Öffentlichkeit selbst ist. Eine rekursive Öffentlichkeit definiert sich dadurch, dass sie die Bedingungen der Möglichkeit ebendieser schafft:

A recursive public is a public that is vitally concerned with the material and practical maintenance and modification of the technical, legal, practical, and conceptual means of its own existence as a public; it is a collective independent of other forms of constituted power and is capable of speaking to existing forms of power through the production of actually existing alternatives.<sup>49</sup>

Nach Kelty bilden die Geeks, welche an freier Software interessiert sind, diese schreiben und verbreiten, eine rekursive Öffentlichkeit. Geeks sind nicht durch einen geographischen oder politischen Raum verbunden, sondern über das Internet. Das Internet ist aber nicht nur der Raum, in welchem sie sich treffen und austauschen können, sondern auch das zentrale Beispiel für freie Technologie und somit Gegenstand ihres geteilten und für sie als Gruppe konstitutiven Interesses. Sie sind daran interessiert, dass das Internet als ein freier Ort für den Austausch existiert. Und dies ist nur möglich, wenn die Technologie, welche das Internet ausmacht, ebenfalls frei ist. Es ist dieses geteilte Interesse am Internet als einem Raum der offenen Begegnung und des freien Austauschs und der Technologie, welche dies ermöglicht, welche die Geeks zu einer Öffentlichkeit macht. Und da das Internet gleichzeitig der Raum ist, in welchem sie sich treffen, handelt es sich um eine rekursive Öffentlichkeit.

[G]eek's affinity with one another is structured by shared moral and technical understandings of order. They are a public, an independent public that has the ability to build, maintain, and modify itself, that is not restricted to the activities of speaking, writing, arguing, or protesting. Recursive publics form through their experience with the Internet precisely because the Internet is the kind of thing they can inhabit and transform.<sup>50</sup>

---

<sup>49</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 3.

<sup>50</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 4.

Die Gemeinschaft der an der Entwicklung freier Software Interessierten und Beteiligten bilden eine rekursive Öffentlichkeit, deren Treffpunkt und gleichzeitig verbindendes Element das Internet ist. Das Internet basiert weitestgehend auf freier Software und dezentraler Architektur. Somit ist es nicht nur Treffpunkt, sondern auch Beispiel. Und um die Errungenschaften des Internets wie Freiheit und Dezentralität zu bewahren, bedarf es wiederum freier Software und dezentraler Architektur.

Die Gemeinschaft der an der Entwicklung freier Software Interessierten und Beteiligten bilden eine rekursive Öffentlichkeit. Die Mitglieder dieser Öffentlichkeit teilen einen Kern von Interessen und Vorlieben, sind dabei aber nicht immer einer Meinung. Vielmehr das Gegenteil ist der Fall: Es wird viel und kontrovers diskutiert und teilweise heftig gestritten. Aber diese Auseinandersetzungen sind kein Grund, sie nicht als eine Gruppe zu betrachten, vielmehr zeugen sie von einem gemeinsamen Grundinteresse und geteilten Wertvorstellungen, welche erst den Raum für diese Kontroversen schaffen. Eine Kontroverse, welche im Kern der freien Software liegt, betrifft die im letzten Kapitel beschriebene Unterscheidung zwischen Open Source und freier Software und damit zusammenhängend die Frage nach dem Umgang mit proprietärer Software. Während einige auf ausschliesslicher Verwendung freier Software beharren, jegliche Verwendung von proprietärer Software ablehnen und die Integration von proprietärer Software in freie Lösungen bekämpfen,<sup>51</sup> so sind andere demgegenüber gleichgültig bis sehr offen. Trotzdem können so verschiedene Individuen der rekursiven Öffentlichkeit, welche sich um freie Software bildet, zugehören. Vertreter beider Ansichten können jedoch zur rekursiven Öffentlichkeit der freien Software gezählt werden, da sie einen minimalen Kern von gemeinsamen Werten teilen.

Nach der oben vorgeneommenen Charakterisierung der rekursiven Öffentlichkeit können neben Personen auch Institutionen, Behörden oder Firmen Teil davon sein. So kann zum Beispiel die Firma RedHat, welche zwar börsenkotiert ist und 2019 von IBM für 34 Milliarden US-Dollar gekauft worden ist<sup>52</sup> Teil dieser Öffentlichkeit sein. RedHat trägt bedeutend zur Wei-

---

<sup>51</sup>Exemplarisch zeigte sich dies bei der Entwicklung von Linux, bei welcher der nicht-freie Source Code Manager Bitlocker verwendet wurde, was zu heftigen Kontroversen und letztlich zur Entwicklung eines freien Source Code Managers (das von Linus Torvalds entwickelte git) führte, vgl. Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 233.

<sup>52</sup><https://www.redhat.com/en/about/press-releases/ibm-closes-landmark-acquisition-red-hat-34-billion-defines-open-hybrid-cloud-future>, aufgerufen am

terentwicklung freier Software bei und ihr zentrales Geschäftsmodell basiert auf freier Software. Hingegen können die Anbieter von proprietärer Software selbst nicht zu dieser Öffentlichkeit gezählt werden, obwohl sie gleichzeitig beträchtliche Ressourcen in die Entwicklung von Software unter Open Source Lizenzen investieren.

Microsoft bezeichnet sich selbst als einen der „top open source collaborators in the world“.<sup>53</sup> Apple sagt umgekehrt, dass verschiedene Open-Source-Anwendungen das Herz der Apple-Produkte ausmacht.<sup>54</sup> Und Google wirbt mit dem Slogan: „We bring all the value of open source to Google and all the resources of Google to open source.“<sup>55</sup> Beide Seiten dieser Behauptung treffen offensichtlich zu: Das von Google für ihre Smartphone-Anwendungen verwendete Betriebssystem Android ist eine direkte Adaption von Linux für mobile Geräte und wird unter einer Open-Source-Lizenz veröffentlicht.<sup>56</sup> Und die Entwicklung von Android, wie auch von Linux, wird massgeblich von Google unterstützt. Andererseits aber sind alle diese drei Techgiganten weit davon entfernt, die Ideale von freier Software zu teilen. Bei Microsoft sind sowohl das Betriebssystem Windows wie auch die Anwendungen aus dem Office-Paket proprietäre Software. Es handelt sich dabei um die nach wie vor kommerziell erfolgreichsten Anwendungen von Microsoft. Microsoft hat sowohl mit dem Betriebssystem Windows, wie auch mit den Anwendungen aus der Office-Palette Quasi-Monopole errichtet, und verlässt sich dabei massgeblich auf den Schutz ihrer Software durch exklusive Copyright-Lizenzen (sowie eine riesige Rechtsabteilung, welche diese Urheberrechte durchsetzt).<sup>57</sup> Ähnlich verhält es sich bei Apple, wo Betriebssysteme für Smartphones wie auch für Apple-Computer durch Copyright-Lizenzen geschützt sind. Bei Apple kommt hinzu, dass Apple nicht in erster Linie durch den Vertrieb von Software, sondern von Hardware verdient, und die Software und Hardware eng aufeinander abgestimmt sind. Diese Abhängigkeit von Hard- und Softwa-

---

7. März 2023.

<sup>53</sup><https://www.microsoft.com/en-us/legal/intellectualproperty/open-source>, aufgerufen am 12. Juli 2023.

<sup>54</sup><https://opensource.apple.com/projects/>, aufgerufen am 12. Juli 2023.

<sup>55</sup><https://opensource.google/>, aufgerufen am 12. Juli 2023.

<sup>56</sup>Android verwendet verschiedene Open-Source-Lizenzen, mehrheitlich die Apache License.

<sup>57</sup>Ironischerweise ist es gerade die Microsoft Tochterfirma GitHub und ihr AI-unterstützter Coding-Assistent Copilot gegen den wegen systematischem Verstoss gegen Copyright-Bestimmungen im November 2022 eine Sammelklage eingereicht wurde (<https://githubcopilotlitigation.com/>, aufgerufen am 12. Juli 2023).

re widerspricht grundsätzlich der Idee freier Software. Und das Zentrum von Google (bzw. des Mutterkonzerns Alphabet) ist weiterhin die Suchmaschine, welche als Werbeplattform verwendet wird und welche die Grundlage für das Geschäftsmodell von Google bildet. Der Algorithmus hinter der Google-Suche gilt als eines der bestgehüteten Geheimnisse der Welt. Dabei wird deutlich, dass die Verwendung und Unterstützung des Open-Source-Betriebssystems Android für Google nur Mittel zum Zweck ist. Eine Aussage des ehemaligen Finanzchefs von Google macht dies deutlich:

If we move forward the adoption of these mobile phones by lowering the cost because it is open source, think of how many searches [that will produce].<sup>58</sup>

Diese Firmen mögen zwar die Entwicklung freier Software vorantreiben, tun dies aber lediglich im Hinblick auf ihr Geschäftsmodell, welches bei allen drei genannten Beispielen auf einer restriktiven Auffassung von geistigem Eigentum basiert und diese voraussetzt und fördert. Diese Firmen können deshalb nicht zur rekursiven Öffentlichkeit freier Software gezählt werden, denn die Einschränkung der Freiheit von Software und dem Internet liegt im Kern ihrer Geschäftsmodelle. Es kann deshalb nicht behauptet werden, dass diese den minimalen Katalog an Werten teilen, welche diese Öffentlichkeit konstituiert.<sup>59</sup>

Weniger eindeutig ist die Zugehörigkeit zu dieser Öffentlichkeit bei IBM. IBM hat als eines der ersten führenden Techunternehmen das Potential freier Software erkannt und genutzt und schon relativ früh einen grossen Teil des Umsatzes mit Produkten erzielt, welche im Zusammenhang mit freier Software standen. Deshalb hat IBM auch immer massgeblich zur Weiterentwicklung freier Software beigetragen. Andererseits war IBM während all den Jahren die Firma mit den weltweit meisten Patentanmeldungen und hat grosse Anstrengungen zum Schutz ihres geistigen Eigentums unternommen und somit auch gegen die zentralen Ideen der Kultur freier Software gearbeitet.

In diesem Kapitel werden die geteilten Werte und Grundhaltungen beschrieben, welche die Bewegung der freien Software ausmachen. Einige dieser Aspekte lassen sich auf die von Steven Levy formulierte Hackerethik

---

<sup>58</sup>Zitiert nach Zuboff, *The Age of Surveillance Capitalism*, S. 134.

<sup>59</sup>Und zumindest Microsoft hat in der Vergangenheit direkte Schritte unternommen, um die Möglichkeit von freier und Open-Source-Software nicht weiter zu fördern, vgl. Lessig, *Free Culture*, S. 262 - 267.



zurückführen,<sup>60</sup> welches nicht nur die frühe Geschichte der Bewegung erzählt, sondern auch den weiteren Verlauf prägte und Bezugspunkt vieler Diskussionen blieb.<sup>61</sup>

Auch wenn in einem Rechtsstaat Rechte gelten und eingehalten werden, so sind technologische Entwicklungen oftmals derart disruptiv, dass sie vielmehr Gesetz schaffen, als durch Gesetz reguliert werden. Diese Erfahrung macht die Gesellschaft immer wieder, aktuellstes Beispiel ist die rasante Verbreitung künstlicher Intelligenz, welche neuartige Fragestellungen produziert, was wir unter anderem an Schulen und Universitäten hautnah miterleben. Die Entwicklung des Internets ist begleitet von solchen Disruptionen. Als ein Phänomen, welches sich über nationale Grenzen hinwegsetzt und sich somit weitgehend einer Regulierung entzieht und durch die Geschwindigkeit der Entwicklungen, welche solche Regulierungen behindern, ist das Internet oder allgemeiner der Cyberspace nicht durch herkömmliche Gesetze reguliert. Daraus folgt aber nicht, dass in diesem Raum keine Gesetze herrschen. Aber die Gesetze, welche im Cyberspace gelten und somit bestimmen, was wem möglich ist, werden durch die Technologie selbst gegeben. Lawrence Lessig hat sich als erster intensiv mit diesem Phänomen beschäftigt und dies mit dem Slogan *Code is Law* beschrieben.<sup>62</sup> Technologie ist nicht neutral. Was für ein Ort das Internet ist und welche Regeln im Internet gelten, wird zu einem grossen Teil durch die Technologie selbst bestimmt. Ist das Internet ein Ort der Freiheit, Kreativität und Partizipation oder der Überwachung und Kontrolle? Für die Öffentlichkeit der freien Software, für welche der Cyberspace der wichtige geteilte Raum bildet, ist es deshalb von Bedeutung, wie die Technologie ausgestaltet ist und welche Gesetze dadurch implementiert werden. Deshalb ist für die Bewegung der freien Software eine weitere von Lessig formulierte Einsicht zentral:

Code is never found; it is only ever made, and only ever made by us.<sup>63</sup>

Wenn die Gesetze in der Technologie festgeschrieben sind, so kann durch die Gestaltung der Technologie auf die Gestaltung der Gesetze Einfluss genommen werden. Aus dieser Einsicht folgt ein Gefühl der Verantwortung, aber

---

<sup>60</sup>Levy, *Hackers: Heroes of the Computer Revolution*.

<sup>61</sup>Webb, *Coding Democracy*, S. 6.

<sup>62</sup>Lessig, *Code Version 2.0*, S. 5.

<sup>63</sup>Lessig, *Code Version 2.0*, S. 6.



auch für Gestaltungsmöglichkeiten. Code wird letztlich darüber entscheiden, wie wir leben. Wir können diesen Code mitgestalten und somit mitentscheiden, wie wir leben wollen. Freie Software macht diese Mitgestaltung real und möglich. Die Einsicht, dass Code die Welt und die Bedingungen prägt, gepaart mit den Fähigkeiten, diese zu gestalten und Einfluss zu nehmen, führen zu einem Gefühl der Verantwortung, Dringlichkeit und Wichtigkeit.<sup>64</sup> Verbunden mit der Einsicht in die Wichtigkeit den Code zu gestalten ist das Verständnis dafür, dass die Technologie das Potential hat, die Welt zum besseren zu verändern. Wie es Levy als einen der Hackergrundsätze formulierte:

Computers can change your life for the better.<sup>65</sup>

Innerhalb der freien Softwarebewegung ist Technooptimismus eine verbreitete Einstellung. Diese Einstellung ist jedoch nicht naiv, sondern immer im Bewusstsein, dass die gleichen Aspekte und Eigenschaften, welche Anlass zu Optimismus geben, auch dystopisches Potential haben. Es spielt deshalb eine Rolle, wie sich die Technologie entwickelt und vor allem wer diese Entwicklung bestimmt und kontrolliert.

Nicht alle an dieser rekursiven Öffentlichkeit beteiligten vertreten die gleichen politischen Ansichten. Es ist sogar so, dass die an der Entwicklung freier Software beteiligten explizit weltanschauliche oder politische Parteinarbeit verweigern, insofern diese über Positionen im direkten Zusammenhang mit freier Software hinausgehen.<sup>66</sup> Sehr deutlich kommt dies im Debian Social Contract zum Ausdruck. Debian war die erste weit verbreitete Distribution auf der Basis von GNU/Linux. Im Debian Social Contract, welcher die Ziele von Debian und die daraus abgeleiteten Verpflichtungen gegenüber der Gemeinschaft freier Software definiert, lautet Punkt 5 der Lizenzierungsrichtlinien:

The license must not discriminate against any person or group of persons.<sup>67</sup>

Dies verunmöglicht es, in der Lizenz beispielsweise festzuhalten, dass die Software nicht zu diskriminierenden oder rassistischen Zwecken verwendet

---

<sup>64</sup>Webb, *Coding Democracy*, S. 53 - 56.

<sup>65</sup>Levy, *Hackers: Heroes of the Computer Revolution*, Kapitel 2.

<sup>66</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 187 - 190, 195.

<sup>67</sup>[https://www.debian.org/social\\_contract](https://www.debian.org/social_contract), aufgerufen am 7. März 2023.

oder für Gruppen mit derartigen Absichten zur Verfügung gestellt werden dürfte.

Es gibt aber Werthaltungen, welche direkt mit der Verwendung, Entwicklung und Verbreitung freier Software zu tun haben und von allen Teilnehmer\*innen dieser Öffentlichkeit geteilt werden. Diese Werthaltungen sind nicht klar innerhalb eines klassischen politischen links-rechts-Schema zu verorten oder einer politischen Partei zuzuordnen, aber konstituieren dennoch eine geteilte politische Grundhaltung. Nicht überraschend haben diese Wertvorstellungen mit Freiheit zu tun - aber Freiheit ist ein breiter Begriff und kann sehr unterschiedlich ausgelegt werden, was gerade am Beispiel der Kultur der freien Software schön aufgezeigt werden kann. Sowohl Liberale wie auch Neoliberale schreiben sich den Begriff der Freiheit auf die Fahnen, der Fokus ist dabei unterschiedlich. Bei klassischen Liberalen steht der Schutz der individuellen Freiheiten (Rede- und Handlungsfreiheit) im Zentrum. Die neoliberale Freiheitsauffassung bildet sich um die unternehmerische Freiheit und damit verbunden den Schutz von Eigentum, wozu auch geistiges Eigentum gezählt wird. Während es oftmals so scheint, als wären liberal und neoliberal lediglich zwei unterschiedliche Ausprägungen einer und derselben Anschauung, so wird am Beispiel der freien Software deutlich, dass sich die beiden Anschauungen nicht nur unterscheiden, sondern sogar widersprechen. Der Schutz des geistigen Eigentums durch Copyright-Lizenzen steht einer freien Entfaltung und Handlungsfreiheit im Weg. Wie Coleman dies beschreibt:

As such, free software hackers not only reveal a long-standing tension within liberal legal rights but also offer a targeted critique of the neoliberal drive to make property out of almost anything, including software.<sup>68</sup>

Es verläuft ein Graben zwischen liberal und neoliberal, und wer der freien Software verpflichtet ist, ist eindeutig auf einer der beiden Seiten anzuordnen, nämlich der liberalen Seite. Dieser Graben lässt sich anhand von zwei unterschiedlichen Gütern, welche beide durch die Verfassung geschützt sind, aufzeigen: Redefreiheit und Schutz geistigen Eigentums. Sowohl Liberale wie auch Neoliberale stimmen selbstverständlich zu, dass beide Güter den Schutz durch die Verfassung verdienen. In Fällen, in denen die beiden jedoch kollidieren, würden sie sich für eine andere Abwägung dieser beiden Güter entscheiden. Dieser Konflikt zwischen zwei Auffassungen von Freiheiten ist in

---

<sup>68</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 4.

verschiedenen Auseinandersetzungen um freie Software und Copyright sichtbar geworden. Deutlich zeigt sich dies bei den Bemühungen von Vertretern freier Software, diese mit dem Recht auf freie Rede in Verbindung zu bringen, was mit dem Slogan *Code is Speech* zum Ausdruck gebracht wird.<sup>69</sup> Es ist eine innerhalb der Bewegung der freien Software geteilte Werthaltung, dass der übertriebene Schutz von geistigem Eigentum zu einer Einschränkung der individuellen Freiheit führt und den Fortschritt hemmt.

Eng verknüpft mit Freiheit ist ein zweites Ideal: Privatsphäre. Erst Privatsphäre ermöglicht es, die Freiheit zu handeln und zu sprechen auszuüben. Die Kultur freier Software ist um das Internet gebildet, entsprechend wichtig ist für sie die Privatsphäre im Internet. Es ist eine weit akzeptierte Ansicht innerhalb der Bewegung der freien Software, dass der Schutz der Privatsphäre im Internet durch Aktivitäten staatlicher und kommerzieller Akteure bedroht ist und es gehört zu den geteilten Werthaltungen dieser rekursiven Öffentlichkeit, dass dem Schutz der Privatsphäre im digitalen Raum grosses Gewicht beigemessen wird. Die verbreitete Verwendung proprietärer Software wird dabei als eine der Hauptgefahren für die Privatsphäre identifiziert. Wenn es unter Androhung massiver Strafen verboten ist, die verwendete Software zu studieren, ist es für deren Hersteller\*innen prinzipiell einfach möglich, sich über diese unerkannt Zugang zu Daten der Benutzer\*innen zu verschaffen. Wie daraus nicht nur direkt die Privatsphäre, sondern viel allgemeiner die individuelle Freiheit bedroht wird, zeigen die umfangreichen Studien von Shoshanna Zuboff<sup>70 71</sup> zu Googles Überwachungskapitalismus, sowie die Enthüllungen von Edward Snowden zur umfangreichen weltweiten Massenüberwachung durch staatliche Behörden. Ein zentrales Element in den Bemühungen um den Schutz der Privatsphäre ist deshalb die Verwendung von freier Software.<sup>72 73</sup> Bestes Zeugnis für die Wirksamkeit von freier Software zum Schutz der Privatsphäre sind die Bemühungen der amerikanischen Sicherheitsbehörden, die Entwicklung des offenen Verschlüsselungssystems PGP zu verhindern.<sup>74</sup>

---

<sup>69</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S.161 - 184.

<sup>70</sup>Zuboff, „Big Other: Surveillance Capitalism and the Prospects of an Information Civilization“.

<sup>71</sup>Zuboff, *The Age of Surveillance Capitalism*.

<sup>72</sup>Moglen, „Snowdon and the Future. Part IV: Freedom’s Future“, S.9.

<sup>73</sup>Lessig, *Code Version 2.0*, S. 138 - 155.

<sup>74</sup><https://philzimmermann.com/EN/background/index.html>, aufgerufen am 16. März 2023.

Privatsphäre ist eine geteilte Werthaltung innerhalb der Bewegung der freien Software. Und ein zentrales Instrument zu diesem Schutz ist die Entwicklung und Förderung freier Software. Dabei zeigt sich deutlich der rekursive Charakter dieser Öffentlichkeit: Das Anliegen und die Mittel, mit welchen dieses umgesetzt wird, bedingen sich gegenseitig.

Eine weitere geteilte Werthaltung innerhalb der Bewegung freier Software ist eine Vorliebe für dezentrale Systeme und ein gewisses Misstrauen gegenüber Autoritäten. Dezentral organisierte Projekte zur Entwicklung freier Software sind Beispiel und Beweis dafür, dass auch dezentrale Systeme erfolgreich funktionieren können. Diese sind am besten in der Lage den Informationsfluss zu garantieren, welcher Bedingung für die Entwicklung komplexer Systeme ist.<sup>75</sup> Dies ist aber nicht in erster Linie eine ideologische Haltung, sondern der Erfahrung in der Entwicklung von freier Software geschuldet, welche nicht nur den Beweis erbrachte, dass Systeme ohne eine zentrale Führung möglich sind, sondern suggeriert, dass dezentrale Systeme sogar am besten in der Lage sind, den Informationsfluss so zu gestalten, dass ein optimaler Austausch von Ideen und Inspiration stattfindet:

The best way to promote this free exchange of information is to have an open system, something which presents no boundaries between a hacker and a piece of information or an item of equipment that he needs in his quest for knowledge, improvement, and time on-line. The last thing you need is a bureaucracy. Bureaucracies, whether corporate, government, or university, are flawed systems, dangerous in that they cannot accommodate the exploratory impulse of true hackers. Bureaucrats hide behind arbitrary rules (as opposed to the logical algorithms by which machines and computer programs operate): they invoke those rules to consolidate power, and perceive the constructive impulse of hackers as a threat.<sup>76</sup>

Dabei ist es wichtig zu verstehen, dass als Alternative zu einer zentralen Kontrollinstanz nicht gefordert wird, dass alle einfach machen was sie wollen. Im Gegenteil: Dezentrale Systeme brauchen Regeln. *Protokolle statt Kontrolle* lautet der Slogan, der dieses Prinzip zusammenfasst.<sup>77</sup> Den Vorzug von Regeln gegenüber Kontrolle zeigt sich exemplarisch in der Entwicklung und

---

<sup>75</sup>Raymond, *The Cathedral and the Bazaar*, S. 52 - 54.

<sup>76</sup>Levy, *Hackers: Heroes of the Computer Revolution*, Kapitel 2.

<sup>77</sup>Vaidhyanathan, *The Anarchist in the Library*, S. 33 - 34.

Organisation des Internets.<sup>78 79</sup> Statt einer autoritären Instanz vertraut man auf die Wirksamkeit von Regeln. Diese Tendenz ist aktuell auch deutlich sichtbar bei Blockchains, bei denen statt Vertrauen in eine Institution wie eine Bank auf den Code und die dahinterliegenden mathematischen Gesetze vertraut wird. Auch bei Blockchains kommt mehrheitlich freie Software zum Einsatz.

Wenn also eine Tendenz zu Vorlieben für dezentrale Systeme, der Vorzug von Protokollen gegenüber Kontrolle und eine damit einhergehende Skepsis gegenüber Autoritäten in der Bewegung freier Software verbreitet ist, so ist dies eine direkte Folge aus der Erfahrung in der Arbeit an freier Software. Die Ideologie folgt der Praxis.

Ebenfalls geteilt in der Bewegung freier Software ist eine meritokratische Einstellung, neben Freiheit und Privatsphäre ein weiteres Kernelement des Liberalismus. In einem meritokratisch organisierten System werden Personen in erster Linie nach der von ihnen erbrachten individuellen Leistung beurteilt und die Qualifikation für bestimmte Ämter oder Positionen wird ausschließlich über die erbrachte Leistung vorgenommen. Die Kultur freier Software verkörpert den meritokratischen Ansatz exemplarisch. Anerkennung hängt nicht von Abschlüssen, Beziehungen oder Statussymbolen ab, sondern allein von den Fähigkeiten und dem Engagement. Leitende Positionen in Projekten zur Entwicklung freier Software werden allein dadurch vergeben, wer nach der Ansicht der anderen Entwickler\*innen sich durch ihre\*seine Fähigkeiten und bereits geleisteten Beiträge für eine Position qualifiziert.<sup>80 81</sup> Diese meritokratische Einstellung stellt Levy als einen der Grundsätze der Hackerethik fest:

Hackers should be judged by their hacking, not by bogus criteria such as degrees, age, race, or position.<sup>82</sup>

Noch deutlicher als darin, dass allein die Leistung als Qualifikation für eine Position zählt, zeigt sich die meritokratische Gesinnung in der Belohnung für erbrachte Leistungen. Denn auch wenn in vielen Fällen Entwicklungsarbeit

---

<sup>78</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 166 - 177.

<sup>79</sup>Vaidhyanathan, *The Anarchist in the Library*, S. 36 - 38.

<sup>80</sup>Moody, *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*, S. 112 - 114.

<sup>81</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 141 - 149.

<sup>82</sup>Levy, *Hackers: Heroes of the Computer Revolution*, Kapitel 2.

bezahlt oder zumindest teilweise bezahlt ist, so ist sie dies in jedem Fall weniger stark, als vergleichbare Arbeit in proprietären Unternehmungen. Die Belohnung der Arbeit und somit der Anreiz diese zu leisten, liegt zu einem grossen Teil in der Anerkennung der Leistung, welche von der Community dieser Arbeit entgegengebracht wird. In der Entwicklung freier Software sind viele namenlose Mitarbeiter\*innen am Werk. Es ist zwar nicht zu übersehen, dass es um einige Personen einen gewissen Kult gibt, aber der dominierende Teil der Arbeit wird von einer Vielzahl von Beitragenden geleistet, welche nicht namentlich bekannt sind.<sup>83</sup> Als Anreiz genügt offenbar vielen Entwickler\*innen die Anerkennung in der Gemeinschaft oder gar nur die Befriedigung darüber, einen Beitrag zu einem als sinnvoll betrachteten Produkt geleistet zu haben.<sup>84</sup>

Verwandt mit der meritokratischen Einstellung ist eine Form von intellektueller Demut, welche ebenfalls verbreitet ist in der Kultur freier Software und direktes Produkt dieser Art der Entwicklung ist. Das Entwickeln von freier Software basiert so gut wie immer auf bereits bestehendem Code, ist eine Fortsetzung und fortlaufende Erweiterung. Dies trägt zu einem Eindruck bei, nachdem eine Person alleine nur wenig bewerkstelligen kann, sich aber einordnet in einen Entwicklungszusammenhang, als Teil eines offenen Projektes. Dies wird manchmal Issac Newton zitierend als *standing on the shoulders of giants* bezeichnet.<sup>85</sup> <sup>86</sup> Durch freie Lizenzen, welche zum Weiterverarbeiten ermutigen, zugleich aber deutlich machen, dass es immer schon ein Erbe ist, welches weitergeführt wird, wird eine solche Haltung in die Praxis umgesetzt. Zur meritokratischen Einstellung gehört auch eine Haltung gegenüber Fehlern und eine Praxis für den Umgang damit. Fehler passieren, wenn gearbeitet wird und sind an sich weder schlimm noch schändlich. Wichtig ist es, dass diese transparent kommuniziert werden, damit sie nicht wiederholt werden und behoben werden können. So heisst es auch wieder im Debian Social Contract:

We will not hide problems[.] We will keep our entire bug report database open for public view at all times. Reports that people file online will promptly become visible to others.<sup>87</sup>

---

<sup>83</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 196 - 198.

<sup>84</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 130.

<sup>85</sup>Benkler, *The Wealth of Networks*, S. 38.

<sup>86</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 103.

<sup>87</sup>[https://www.debian.org/social\\_contract](https://www.debian.org/social_contract), aufgerufen am 7. März 2023.

Auch die Haltungen zur Ökonomie sind nicht einheitlich innerhalb der Geeks und decken von kapitalismuskritisch bis marktliberal das ganze Spektrum ab. Aber die Entwicklung freier Software hat zu zwei wichtigen Erkenntnissen geführt, welche mit herkömmlichen Lehrmeinungen aus der Ökonomie in Widerspruch stehen. Erstens haben grosse Projekte wie Debian oder Apache gezeigt, dass die Entwicklung von komplexen technologischen Produkten ohne eine starre hierarchische Organisation möglich ist. Und zweitens hat die Vielzahl von freier Software gezeigt, dass es nicht stimmt, dass ohne den Schutz von geistigem Eigentum und dem damit einhergehenden Versprechen monetären Gewinns keine Anreize für Innovation und Leistung vorhanden sind. Beides sind prägende Erfahrungen für die Bewegung freier Software und bilden einen geteilten Kern von Ansichten über die Organisation von und Anreize für Arbeit.

Im Zuge der Entwicklung freier Software haben sich neue Formen der Zusammenarbeit entwickelt und herausgebildet. Auch hier gibt es unterschiedliche Ansätze, Modelle mit Abstimmungen, wie sie bei Apache zur Anwendung kommen oder ein hierarchisches Modell mit einem gütigen Diktator, wie es während langer Zeit bei der Entwicklung von Linux zur Anwendung kam.<sup>88</sup> Allen diesen Formen ist aber gemein, dass sie keine genauen Zielformulierungen und weit vorausschauende Planungen kennen, sondern flexibel sind und durch die Bedürfnisse und Fähigkeiten der an der Entwicklung beteiligten Individuen geprägt sind. Es liegt in der Natur freier Software, dass sie (zumindest im Grossen und Ganzen) nicht top-down organisiert werden können, sondern geprägt durch die Interessen, Bedürfnisse und Fähigkeiten der an der Entwicklung beteiligten Individuen immer einem bottom-up-Ansatz folgen. Eine Wertschätzung dieser Arbeitsweise gehört zur Kultur freier Software.

Die Kultur der Geeks, welche sich um freie Software und das Internet bildet, wird ebenfalls durch eine geteilte Ästhetik zusammengehalten, welche weit über die geteilte Wertschätzung für eleganten Code hinausgeht.<sup>89</sup> Selbstverständlich haben nicht alle Geeks den gleichen Geschmack in jeder Hinsicht, jedoch lassen sich bestimmte Vorlieben ausmachen, die sehr breit geteilt werden. Eine wesentliche geteilte Vorliebe ist ein bestimmter Sinn für Humor.<sup>90</sup> Dieser Humor zeigt sich beispielsweise und prominent in der Wahl der Namen von Programmen, häufig sind diese zirkuläre Akronyme wie zum Beispiel

---

<sup>88</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 180, 202.

<sup>89</sup>Levy, *Hackers: Heroes of the Computer Revolution*, Kapitel 2.

<sup>90</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 6 - 7, 92 - 122.



GNU (Gnu's not Unix) oder PHP (PHP: Hypertext Processor), erzählen auf amüsante und lautmalerische Weise von der Entstehung der Software wie Unix (das als "kastrierte" Version seines Vorgängers und deshalb ein Eunuch betrachtet wurde) oder Apache (a patchy server), drücken ein selbstironisches Understatement aus wie PGP (pretty good privacy) oder verweisen direkt auf schwarzen Humor wie Python (benannt nach der englischen Comedy-Truppe Monty Python). Diese Namensgebungen zeugen von Ironie sich selbst und dem eigenen Tun gegenüber. Ein weiteres geteiltes Element betrifft die Art, wie auch in teilweise ernsten Diskussionen über Vorstellungen und Ziele popkulturelle Verweise und insbesondere historische Analogien bemüht werden (zum Beispiel in dem die Geschichte der freien Software verglichen wird mit der Reformation im Christentum, wobei wahlweise proprietäre Softwareanbieter oder andere als böse betrachtete Elemente die Rolle der katholischen Kirche einnehmen).<sup>91</sup>

Freie Software ist mehr als Code unter einer freien Lizenz. Freie Software ist eine Kultur - eine rekursive Öffentlichkeit -, welche mit der Verwendung und Entwicklung des Codes und den technischen, juristischen, finanziellen und organisatorischen Bedingungen beschäftigt ist. Diese vielfältige Kultur freier Software hat einen Einfluss auf verschiedene andere Gebiete entwickelt. Einige davon werden im nächsten Kapitel dargestellt.

## 2.3 Einfluss freier Software

Freie Software ist mittlerweile weit verbreitet und respektiert. Wenn auch im Konsument\*innenbereich nach wie vor fast keine freie Software verwendet wird, so ist diese in der Forschung und der Entwicklung von Software allgegenwärtig. Freie Software hat darüber hinaus auf Gebiete Einfluss genommen, welche nicht oder nicht direkt mit Software zu tun haben.

Die bei der Entwicklung freier Software entwickelte Arbeitsweise und Organisationsform hat Arbeitsbereiche beeinflusst, welche nicht direkt mit der Entwicklung von Software zu tun haben: Bottom-up basierte Arbeitsorganisation, flachere Hierarchien und eine offene Fehlerkultur gehören zu den Konzepten, welche vielerorts in der Arbeitswelt Einzug gehalten haben. Die Entwicklung freier Software dient als Beispiel und hat neue Wege eröffnet.

Die spezifische Zusammenarbeitsform bei der Entwicklung von freier Software Projekten erforderte neben neuen Formen der Organisation auch tech-

---

<sup>91</sup>Vgl. Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 64 - 76.



nische Hilfsmittel. Der von Einzelnen geschriebene Code musste für alle an der Entwicklung beteiligten zugänglich gemacht werden, die Veränderungen am Code mussten nachvollziehbar sein und es musste verhindert werden, dass Beiträge verschiedener Entwickler\*innen zu Komplikationen und Inkompatibilitäten führten. Hierzu wurden verschiedene Source-Code-Management-Systeme entwickelt, welche schnell breite Anwendung fanden und heute bei keiner Softwareentwicklung fehlen.<sup>92</sup>

Der Einfluss der Kultur freier Software ist auch bei konkreten Unternehmen deutlich erkennbar. Als gutes Beispiel dient IBM. Als eines der traditionsreichsten Tech-Unternehmen, das auf einer langen Tradition von Patenten und geschütztem geistigem Eigentum basierte, hat IBM früh das Potential von freier Software erkannt. Bereits 2004 war ein grosser Teil des Umsatzes direkt auf Produkte und Dienstleistungen zurückzuführen, welche im Zusammenhang mit freier Software standen.<sup>93</sup> Aber IBM profitiert nicht nur von den Entwicklungen freier Software, sondern finanziert zu einem bedeutenden Anteil verschiedene Entwicklungen. Bereits 2006 kam Benkler deshalb zu folgendem Urteil über das Verhältnis von IBM und freier Software:

It's strategy is, if not symbiotic, certainly complementary to free software.<sup>94</sup>

Während diese Integration von Open Source Entwicklung während langer Zeit parallel zu den Aktivitäten mit proprietärer Software und Patenten war und IBM seit Jahrzehnten das Unternehmen mit den meisten Patentanmeldungen weltweit war, so hat sich in neuerer Zeit ein Umdenken abgezeichnet. Es wird zwar weiterhin an der Entwicklung von Patenten und Copyright festgehalten, aber dem wird zugunsten von der noch stärkeren Involviertheit in Open-Source-Projekte weniger Gewicht eingeräumt.<sup>95</sup> Exemplarisch dafür ist die Forschung im Bereich von Quantencomputern, in welcher IBM mit qiskit zu den führenden Teilnehmern gehört und dieses neuerdings unter einer Apache Lizenz veröffentlicht.<sup>96</sup> Als weiterer Schritt in dieser Entwicklung kann der Erwerb von RedHat genannt werden.

---

<sup>92</sup>Die heute am weitesten verbreiteten Systeme basieren auf git, welches von Linux-Gründer Linus Torvalds entwickelt wurde.

<sup>93</sup>Benkler, *The Wealth of Networks*, S. 46 - 47.

<sup>94</sup>Benkler, *The Wealth of Networks*, S.47.

<sup>95</sup><https://fortune.com/2023/01/06/ibm-patent-record-how-to-measure-innovation-open-source-quantum-computing-tech/>, aufgerufen am 11. März 2023.

<sup>96</sup><https://github.com/Qiskit/qiskit/blob/master/LICENSE.txt>, aufgerufen am 11. März 2023.

Ein weiterer wichtiger Einfluss hat freie Software auf die Forschung und den Umgang mit geistigen und kreativen Erzeugnissen und deren Behandlung als Gemeingüter. Seit den 1970-Jahren widmete sich eine wachsende Gemeinschaft von Forschenden aus den Bereichen Ökonomie und Ökologie dem Studium von Commons.<sup>97</sup> Commons sind Güter, auf welche niemand Eigentumsanspruch erheben kann und welche von allen Menschen frei genutzt werden können. Luft, Wasser und andere natürliche Ressourcen sind Beispiele für Commons. Die Forschung zu Commons wurde geprägt durch die Arbeiten der Ökonomin Elinor Ostrom, welche dafür 2009 mit dem Nobelpreis für Ökonomie geehrt wurde.

Güter, auf die niemand exklusiven Besitz- oder Gebrauchsanspruch erheben kann, drohen schnell ausgebeutet zu werden. Was niemandem gehört, dafür fühlt sich auch niemand verantwortlich. Was keinen Wert hat, dem wird auch kein Wert beigemessen. Dieses Problem wurde bereits Mitte des 20. Jahrhunderts formuliert und als Tragik der Commons beschrieben. Demnach ist es für ein einzelnes Individuum rational eine Entscheidung zu treffen, welche für die Gesamtheit der betroffenen Individuen nachteilig ist. Konkret bedeutet dies, dass es für das einzelne Individuum rational ist, die Ressourcen der Commons grösstmöglichst abzuschöpfen, obwohl dieses Vorgehen für die Gesamtheit der Individuen (und also auch für das einzelne Individuum) nachteilig ist.<sup>98</sup> Als Lösung für dieses Problem wurden in der klassischen Forschung lediglich zwei Ansätze in Betracht gezogen: Staatliche Regulierung oder Privatisierung. Die empirischen und theoretischen Forschungen von Ostrom zeigten aber, dass diese zwei Ansätze das Feld der Handlungsmöglichkeiten nicht erschöpfen. Dabei zeigte sich, dass die Annahmen über das Verhalten der Menschen, welche bei der Formulierung der Tragik vorausgesetzt wurden, nicht in jedem Fall zutreffen. Es stellt sich deshalb für Ostrom und andere Forscher\*innen die Frage, welche institutionellen und sozialen Voraussetzungen einen nachhaltigen Umgang mit Commons erlauben.

Es gibt neben natürlichen Ressourcen wie Wasser und Luft weitere Commons, nämlich immaterielle Güter wie Wissen oder kulturelle und kreative Entwicklungen. Wie die studierten Commons gehören auch diese nicht einzelnen Individuen und können von allen Menschen frei genutzt werden. An-

---

<sup>97</sup>Die deutsche Übersetzung von *commons* ist *Allmende*. Da der englische Ausdruck aber auch auf Deutsch verbreitet ist, verwende ich diesen.

<sup>98</sup>Ostrom, *Die Verfassung der Allmende. Jenseits von Staat und Markt*, S. 1 - 19.

ders als die natürlichen Ressourcen sind diese aber nicht erschöpfbar.<sup>99 100</sup> Wenn mehr Personen diese nutzen, nimmt deren Wert oder Bestand nicht ab. Entsprechend entsteht die Tragik der Commons in Bezug auf diese nicht. Aber es gibt andere Schwierigkeiten in ihrem Umgang. Denn anders als bei natürlichen Ressourcen bedarf es eines Aufwandes, um solche Güter zu produzieren. Wer ist aber bereit in deren Produktion zu investieren, wenn diese danach frei verfügbar sind? Für Güter, bei denen eine zusätzliche Nutzung keine zusätzlichen Gewinne verspricht, tendieren die Grenzkosten gegen Null. Aus Sicht der klassischen Ökonomie fehlen in einer solchen Situation Anreize für die Produktion und diese findet nicht statt.

Public goods have always attracted the interest of economists, because the price system is, at least theoretically, an inferior way to provide such goods. When the marginal cost of providing a good is zero, the most economically efficient price, on the margin, is also zero. But such goods are often not costless to create. Therein lies the dilemma. How does one allocate resources to create goods that will have a zero price?<sup>101</sup>

Um die Investition in die Produktion solcher Güter sicherzustellen, müssen diese deshalb der freien Zugänglichkeit entzogen werden. In der Praxis geschieht dies durch Copyright und Patentschutz. Diese künstliche Verknappung eines an sich nicht erschöpfbaren Gutes stellt die Produktion sicher. Gleichzeitig ist dies ein Eingriff in den freien Markt, welcher aus liberaler ökonomischer Sicht nicht wünschenswert sein kann. Also auch hier ein tragisches Dilemma:

On any given day, enforcing copyright law leads to inefficient underutilization of copyrighted information. However, looking at the problem of information production over time, the standard defense of exclusive rights like copyright expects firms and people not to produce if they know that their products will be available for anyone to take for free.<sup>102</sup>

---

<sup>99</sup>Bollier, „The Growth of the Commons Paradigm“.

<sup>100</sup>Hess, *Understanding Knowledge as Commons*.

<sup>101</sup>Love, „Paying for Public Goods“, S. 208.

<sup>102</sup>Benkler, *The Wealth of Networks*, S. 37.

Aus Sicht der klassischen Ökonomie entsteht im Zusammenhang mit solchen Gütern deshalb ebenfalls eine Tragik.

Software ist ebenfalls ein nicht-erschöpfbares Gut. Wenn jemand ein Programm geschrieben hat, wird dieses nicht weniger brauchbar, wenn ich dieses ebenfalls brauche. Abgesehen vom verschwindend kleinen Aufwands für das Kopieren und Verbreiten des Codes entstehen keine zusätzlichen Kosten, die Grenzkosten tendieren gegen Null. Nach der ökonomischen Lehrmeinung dürfte es also nicht der Fall sein, dass Software entwickelt wird, ohne dass diese künstlich verknappt wird. Die funktionierenden freien Software Projekte widersprechen dieser Annahme:

[...] free software practices denaturalize the assumption that intellectual property instruments hold a singular relationship between means and ends - a relationship that can only be established by institutions of authority, notably courts and governments.<sup>103</sup>

Dadurch weist die Praxis freier Software auf eine dritte Möglichkeit und einen möglichen Ausweg aus dem Dilemma hin. Dies hat das Interesse der Forscher\*innen geweckt, welche sich dem Studium von Commons widmeten. Und parallel zum Interesse von Forscher\*innen sind Projekte entstanden, welche den Umgang mit geistigem Eigentum von freier Software auf andere Bereiche übertragen.<sup>104 105 106 107 108</sup>

Das Projekt Connexions (später OpenStax)<sup>109</sup> ermöglicht es, Unterrichtsmaterialien unter freien Lizenzen zu veröffentlichen und so leicht zugänglich und veränderbar zu machen.<sup>110</sup> Eine noch allgemeinere Initiative ist Creative Commons,<sup>111</sup> welche ebenfalls nach dem Vorbild der Copyleft-Lizenzen für freie Software Lizenzen für andere kreative Erzeugnisse erstellt.<sup>112</sup> Gegründet 2001,<sup>113</sup> waren bis zum Jahr 2019 beinahe 2 Milliarden Werke unter einer

---

<sup>103</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 201.

<sup>104</sup>Heinz, „Freie Software - Vom Elfenbeinturm auf unseren Schreibtisch“.

<sup>105</sup>Schweik, „Free/Open-Source Software as a Framework for Establishing Commons in Science“.

<sup>106</sup>Webb, *Coding Democracy*, S. 157.

<sup>107</sup>Benkler, „Coase's Penguin, or, Linux and the Nature of the Firm“.

<sup>108</sup>Bollier, „Protecting and Enlarging the Digital Republic“.

<sup>109</sup><https://openstax.org/>, aufgerufen am 14. Juli 2023.

<sup>110</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S. 244 - 300.

<sup>111</sup><https://creativecommons.org/>, aufgerufen am 29. April 2023.

<sup>112</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 197.

<sup>113</sup>Kelty, *Two Bits: The Cultural Significance of Free Software*, S.258 - 276.

Creative-Commons-Lizenz veröffentlicht.<sup>114</sup> Das Ziel von Creative Commons ist es, den immer restriktiveren Copyright-Bestimmungen entgegenzuwirken und der Kultur und Kreativität mehr Spielraum zu geben.<sup>115</sup> Creative Commons ist eine Antwort auf eine ähnliche Situation mit Copyright, welche am Anfang der Entwicklung freier Software war und für welche die GNU General Public License die Antwort war.

Zwischen der Wissenschaft und freier Software bestehen vielfältige Beziehungen, Beeinflussungen, Gemeinsamkeiten und Abhängigkeiten.<sup>116</sup> Die Kultur der freien Software ist zuerst im universitären Umfeld entstanden. In der Frühzeit der Entwicklung von Software gab es keine Computer für den privaten Gebrauch, Computer standen ausschliesslich in grossen Unternehmen und in universitären oder anderen Forschungseinrichtungen. Hacker, die freie Software entwickeln wollten, waren also während einer langen Zeit auf die Möglichkeit angewiesen, diese Geräte zu verwenden und am ehesten zugänglich waren diese an den Universitäten. Aber die Universitäten und die Wissenschaft boten nicht nur Zugang zur Infrastruktur, sondern prägten auch den Geist freier Software.<sup>117</sup> Einige der im letzten Kapitel als Kernelemente der Kultur der freien Software genannten Ideale, Verfahren und Arbeitsweisen haben ihren Ursprung in den Wissenschaften.<sup>118 119</sup>

Das Bewusstsein dafür, dass jede gewonnene Erkenntnis auf den Erkenntnissen der Vergangenheit beruht und der Wert einer neuen Entdeckung vor allem darin besteht, dass sie die Grundlage für weitere sein kann, ist ein zentraler Bestandteil wissenschaftlichen Arbeitens. Dies widerspiegelt sich in einem über Jahrhunderte elaborierten Verfahren des Zitierens und Referenzierens und der grundsätzlichen Freiheit, die Arbeiten auf bestehenden Erkenntnissen aufzubauen. Beides gilt auch für die Entwicklung freier Software. Daraus folgt ein liberales Verständnis von geistigem Eigentum, welches die Grundlage sowohl wissenschaftlichen Fortschritts, als auch der Entwicklung freier Software ist. Auch in den Wissenschaften ist eine meritokratische Einstellung weit verbreitet und dies kann als einer der Gründe angesehen werden, dass diese auch in der Kultur freier Software verbreitet ist. Peer-

---

<sup>114</sup>[https://wiki.creativecommons.org/images/2/20/CC\\_AnnualReport\\_2019.pdf](https://wiki.creativecommons.org/images/2/20/CC_AnnualReport_2019.pdf), aufgerufen am 29. April 2023.

<sup>115</sup>Lessig, *Code Version 2.0*, S. 282 - 286.

<sup>116</sup>Stallman, „Free Community Science and the Free Development of Science“.

<sup>117</sup>Torvalds, *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*, S. 103.

<sup>118</sup>Benkler, „Coase's Penguin, or, Linux and the Nature of the Firm“, S. 172.

<sup>119</sup>Vaidhyanathan, *The Anarchist in the Library*, S. 131 - 135.

Review-Verfahren haben ihren Ursprung in den Wissenschaften, eben so die Einsicht in die Wichtigkeit einer funktionierenden Fehlerkultur.

Wissenschaftliche Erkenntnisse sind weitgehend frei verfügbar und werden als Commons betrachtet. Dass wissenschaftliche Erkenntnisse als gemeine Güter behandelt werden ist im Einklang mit der Tatsache, dass diese zu einem grossen Teil durch öffentliche Gelder gefördert und ermöglicht werden. Während dies in der wissenschaftlichen Community weitgehend Konsens und Praxis ist, ist der Zugang zu diesen Erkenntnissen bzw. zu den Artikeln, in welchen diese veröffentlicht werden, stark eingeschränkt. Verlage, welche wissenschaftliche Magazine veröffentlichen, beschränken den Zugang zu diesen mit Paywalls. Und auch der Eingang in diese Magazine ist mehrheitlich an finanzielle Bedingungen geknüpft. So kommt es zur Situation, dass ein Teil der Forschungsgelder investiert werden müssen, um die Artikel publizieren zu können. Und andererseits ein Teil des für die Universitäten und somit für Ausbildung und Forschung zur Verfügung stehenden Gelder investiert werden, um Zugänge zu diesen Artikeln zu schaffen. In den Augen vieler Wissenschaftler\*innen ist dies eine absurde Situation, insbesondere in Anbetracht dessen, dass durch digitale Technologie Artikel sehr einfach veröffentlicht und zugänglich gemacht werden könnten. Nach dieser Auffassung ist das Ideal, dass es sich bei wissenschaftlichen Erkenntnissen um Allgemeingüter handelt erst umgesetzt, wenn auch der Zugang zu den Artikeln, in denen diese festgehalten werden, öffentlich ist. So sind verschiedene Initiativen entstanden um wissenschaftliche Arbeiten einfacher zugänglich zu machen. Diese sind sehr stark inspiriert durch die Praxis der freien Software.<sup>120</sup> Zudem verwenden sie oftmals auch Technologie, welche unter freien Lizenzen veröffentlicht wurden.

Im letzten Kapitel habe ich betont, dass die Kultur freier Software über den Raum dessen, was den Kern dieser Kultur selbst betrifft, sich explizit zu einem politischen und weltanschaulichen Agnostizismus bekennt. Das heisst aber nicht, dass es nicht eine Vielzahl von Menschen aus dieser Öffentlichkeit gibt, die sich neben diesem Engagement ebenfalls explizit politisch engagieren.<sup>121 122 123</sup> Die Kämpfe für ein freies Internet und Free Speech im

---

<sup>120</sup>Schweik, „Free/Open-Source Software as a Framework for Establishing Commons in Science“.

<sup>121</sup>Coleman, *Coding Freedom. The Ethics and Aesthetics of Hacking*, S. 196.

<sup>122</sup>Caffentzis, „A Tale of Two Conferences: Globalization, the Crisis of Neoliberalism and the Question of the Commons“.

<sup>123</sup>Schoonmaker, *Free Software, The Internet, and Global Communities of Resistance*, S. 222 - 229.

Zusammenhang mit Free Code verallgemeinern sich zu allgemeinerer Kritik am globalen Kapitalismus. Dadurch entsteht eine Beeinflussung von politischen Bewegungen durch die Methoden und Ideale und auch die technischen Möglichkeiten von freier Software.<sup>124</sup> Relativ direkt kommen diese Einflüsse beim Kampf für mehr Transparenz bei öffentlichen Angelegenheiten und mehr Schutz der Privatsphäre zusammen.

Viele der in diesem Kapitel beschriebenen Einflüsse von freier Software kommen bei Wikipedia zum Ausdruck. Sämtliche Inhalte von Wikipedia werden unter Copyleft-Lizenzen herausgegeben, wobei die meisten Inhalte sowohl unter einer Version der Creative Commons Licenses und der GNU Free Documentation License lizenziert sind.<sup>125</sup> Zudem ist die Arbeitsweise von Wikipedia mit freiwilligen Autor\*innen, einem klar vorgegebenen Editionsprozess und einem Peer-Review-System von der Entwicklung freier Software inspiriert. Und wie freie Software stellt Wikipedia gleichzeitig mehrere zentrale Lehrmeinungen der Ökonomie in Frage, indem es ohne finanzielle Anreize und ohne eine starre Hierarchie gelingt, ein enorm komplexes und qualitativ hochwertiges Produkt zu schaffen. Und die Technologie, auf welcher Wikipedia basiert, ist freie Software, nämlich MediaWiki, welches unter einer GNU General Public License veröffentlicht wird.<sup>126</sup> Und auch andere zur Produktion von Wikipedia verwendete Software ist freie Software, beispielsweise wird in der Behandlung von Anfragen die freie Software Znuun verwendet.<sup>127</sup>

---

<sup>124</sup>Webb, *Coding Democracy*.

<sup>125</sup><https://en.wikipedia.org/wiki/Wikipedia:Copyrights>, aufgerufen am 7. März 2023.

<sup>126</sup><https://www.mediawiki.org/wiki/Download>, aufgerufen am 7. März 2023.

<sup>127</sup><https://www.znuun.org/en>, aufgerufen am 7. März 2023.

# Kapitel 3

## Freie Software am Gymnasium

In diesem zweiten Teil argumentiere ich dafür, dass die Praxis und Kultur freier Software gut zu den für die Bildung im Allgemeinen und dem Gymnasium im Speziellen vorgegebenen Werten passt und deshalb eingesetzt werden sollte. Werte sind für die Bildung wichtig, diese zu bestimmen und ihre Rolle zu beschreiben, ist allerdings nicht trivial. Einerseits braucht Bildung Werte zur Orientierung. Andererseits dürfen in der Bildung weder religiöse noch politische Überzeugungen vermittelt werden und allgemein gilt die Forderung, die Bildung möglichst wertneutral zu gestalten. Dies ergibt ein Spannungsfeld in welchem sich die Schulen und die Bildungspolitik bewegen. Eine fein austarierte und über lange Zeit etablierte Praxis sichert einen Umgang, der das Handeln in diesem schwierigen Spannungsfeld nicht perfekt, aber doch auf eine Art und Weise ermöglicht, welche breit akzeptiert ist.

Um dem Anspruch an Wertneutralität gerecht zu werden, kann bei einer wertebasierten Begründung von Bildungszielen und Inhalten nur auf einen minimalen Kanon von Werten Bezug genommen werden. Konkret können dazu Werte wie Demokratie und Rechtsstaatlichkeit, gesellschaftliche Solidarität, aber auch Wissenschaftlichkeit und eine Vorstellung von persönlicher Autonomie gezählt werden. Die so formulierten Werte entsprechen jenen der Aufklärung, in deren Tradition sich die Bildung und in besonderem Mass die gymnasiale Bildung sieht und verordnet. Diese Werte finden sich in den allgemeinsten Bestimmungen wieder, welche für die Bildung formuliert werden.

In diesem Teil der Arbeit soll aus den für das Gymnasium geltenden Rahmenbedingungen und Vorgaben abgeleitet werden, dass freie Software eingesetzt werden sollte. Bei der Basis für die Begründung wird vom Speziellen zum Allgemeinen gegangen: Als erstes werden konkrete auf kantonaler



und nationaler Ebene geltende Gesetzesartikel und Umsetzungsmassnahmen für die Verwendung digitaler Geräte untersucht, welche für den Einsatz von digitalen Geräten im Unterricht eine doppelte Zielsetzung formulieren. Der nächste Untersuchungsgegenstand sind die im Artikel 5 des Maturitätsverordnung formulierten allgemeinen Bildungsziele für das Gymnasium. Und letztlich wird auf die Ideale der Aufklärung Bezug genommen, welchen der gymnasiale Unterricht seit jeher verpflichtet ist. Zu diesem Zweck wird der programmatische Aufsatz von Immanuel Kant *Beantwortung der Frage: Was ist Aufklärung?* herangezogen.

Die Kapitel präsentieren je ein Argument für den Einsatz freier Software an der Schule. Die Form der Argumente ist dabei insofern gleich, dass ausgehend von einem für die gymnasiale Bildung geltenden Ziel gezeigt wird, dass der Einsatz freier Software dessen Erreichen begünstigt und deshalb eingesetzt werden sollte. Es handelt sich entsprechend ausschliesslich um positive Argumente für freie Software: Es wird nicht damit argumentiert, dass der Einsatz von proprietärer Software problematisch ist, sondern dass der Einsatz von freier Software den Bildungsidealen entspricht. Während eines dieser Argumente für sich genommen möglicherweise nicht genügt, um den Einsatz von freier Software am Gymnasium zu begründen, so sollten sie im Verbund dazu in der Lage sein.

### 3.1 Doppelte Zielsetzung

Mit dem Einsatz von Computern an der Schule wird eine doppelte Zielsetzung verfolgt. Einerseits soll damit die didaktische Vielfalt erhöht und neue Formen des Lernens sollen ermöglicht werden. Andererseits werden Computer in der Schule eingesetzt, um die Fähigkeiten im Umgang mit Computern zu trainieren und das Wissen über Computer zu schulen. Dies geschieht im Hinblick darauf, dass solide Computerkenntnisse für die meisten Studien und Berufe Voraussetzung sind. Computer sollen also einerseits dem Lernen dienen und sind somit ein Mittel zu einem Zweck. Andererseits sind sie selbst der Gegenstand des Lernens und ihr Einsatz ein Selbstzweck.

Beide Zielsetzungen finden sich in den Strategiepapieren auf allen Ebenen des Bildungswesens. So schreibt die Konferenz der Eidgenössischen Erziehungsdirektor\*innen in den Massnahmen zur Digitalisierungsstrategie im Juni 2019:

Schülerinnen, Schüler und Lernende sind kompetent im Umgang

mit Digitalisierung. Sie werden in ihrer Entwicklung zu eigenständigen und verantwortungsbewussten Bürgerinnen und Bürgern in einer digitalen Welt unterstützt. [...] Die Schülerinnen, Schüler und Lernende verfügen über geeignete digitale Lehr- und Lernmittel. Das Potenzial dieser Unterrichtsmittel im Hinblick auf Barrierefreiheit und die Gestaltung individuell angepasster Lehr- und Lernprozesse wird ausgeschöpft. <sup>1</sup>

Ins Legislaturprogramm des Kantons Luzerns für die Jahre 2019 - 2023 finden ebenfalls beide Ziele Eingang. Die Digitalisierung verspricht Chancen für die Didaktik, welche genutzt werden sollen:

Wir nutzen die Chancen, die sich aus der Digitalisierung im Bildungsbereich ergeben. Wir unterstützen die Einführung digitaler Lernmedien und die Anwendung digitaler Lernformen, damit Lehrende und Lernende auf allen Stufen vielfältige und adäquate Lernformen einsetzen und mit Hilfe der Digitalisierung zielführende und abwechslungsreiche Lernprozesse gestalten können. <sup>2</sup>

Und die Digitalisierung und die Technologien dahinter müssen verstanden werden, um in einer durch diesen Wandel geprägten Welt leben zu können:

Der digitale Wandel verändert die Gesellschaft schnell und tiefgreifend. [...] Mit geeigneten Bildungs-, Ausbildungs- und Weiterbildungsangeboten schaffen wir die Voraussetzungen dafür, dass die Bevölkerung die digitalen Chancen nutzen und mit den Risiken umgehen kann. <sup>3</sup>

Und in der Strategie der Kantonalen Dienststelle Gymnasium wird diese doppelte Zielsetzung ebenfalls festgehalten:

---

<sup>1</sup><https://edudoc.ch/record/204727>, S.5, aufgerufen am 28. März 2023.

<sup>2</sup>[https://www.lu.ch/-/media/Kanton/Dokumente/Regierung/Kantonsstrategie/B3\\_Legislaturprogramm\\_2019\\_2023.pdf?rev=b4fc577c937047fca71af507131a9eb8&hash=074424EAA86DF2EDF780FFDD0B231F4D](https://www.lu.ch/-/media/Kanton/Dokumente/Regierung/Kantonsstrategie/B3_Legislaturprogramm_2019_2023.pdf?rev=b4fc577c937047fca71af507131a9eb8&hash=074424EAA86DF2EDF780FFDD0B231F4D), S. 8 , aufgerufen am 11. Juli 2023.

<sup>3</sup>[https://www.lu.ch/-/media/Kanton/Dokumente/Regierung/Kantonsstrategie/B3\\_Legislaturprogramm\\_2019\\_2023.pdf?rev=b4fc577c937047fca71af507131a9eb8&hash=074424EAA86DF2EDF780FFDD0B231F4D](https://www.lu.ch/-/media/Kanton/Dokumente/Regierung/Kantonsstrategie/B3_Legislaturprogramm_2019_2023.pdf?rev=b4fc577c937047fca71af507131a9eb8&hash=074424EAA86DF2EDF780FFDD0B231F4D), S. 18, aufgerufen am 11. Juli 2023.

Die gesellschaftlichen Erwartungen an künftige Maturandinnen und Maturanden implizieren, dass die Abgänger/-innen der allgemeinbildenden Schulen über Kenntnisse und Fertigkeiten zur Bewältigung des digitalen Wandels (Informatik, Anwendungskompetenzen und Medienbildung) verfügen. Die Schulen sind bestrebt, die Möglichkeiten der Digitalisierung für die Gestaltung von Lehr- und Lernprozessen zu erkennen und zu nutzen.<sup>4</sup>

Diese beiden Zielsetzungen betreffen nicht den Informatikunterricht, sondern den Unterricht aller Fächer. Der angestrebte Nutzen vom Einsatz von Computern betrifft alle (oder zumindest viele) Fächer und bei den zu erwerbenden Fähigkeiten, welche durch ihren Einsatz gefördert werden, handelt es sich um transversale Kompetenzen, zu deren Förderung und Entwicklung alle Fächer einen Beitrag leisten müssen.

Die beiden Zielsetzungen widersprechen sich nicht, in der konkreten Umsetzung können sie aber zu Konflikten führen. Ein Mittel zum Zweck muss in erster Linie diesem Zweck dienen. Wenn Computer im Unterricht als Mittel zum Zweck höherer didaktischer Vielfalt oder neuer Lernformen eingesetzt werden, müssen diese und die gesamte dafür benötigte Infrastruktur möglichst störungsfrei und einfach funktionieren. Wenn Lehrpersonen Computer zu diesem Zweck einsetzen, empfinden sie technische Störungen als Störung des Unterrichts. Für den Umgang mit solchen Störungen sind keine Ressourcen im Fachunterricht eingeplant und den Lehrpersonen fehlt die Kompetenz für den Umgang damit.

Einfachheit und Störungsfreiheit - typischerweise werden diese Anforderungen von einer Lösung eines kommerziellen Anbieters mit verschiedenen Diensten aus einer Hand gut abgedeckt. Schliesslich sind es diese Anforderungen, welche Konsument\*innen an Produkte stellen. Diese Konsumhaltung Computern gegenüber steht aber möglicherweise der Entwicklung eines vertieften Verständnisses und somit dem Erreichen des zweiten Ziels im Weg.

Um zu verstehen, wieso der Einsatz von Computern dem vertieften Verständnis von Computern im Weg stehen könnte, soll dem bisherigen Einsatz ein alternatives Szenario gegenübergestellt werden. Dabei handelt es sich nicht um einen ausgearbeiteten Umsetzungsvorschlag, sondern um ein heuristisches Mittel, um den Raum der Möglichkeiten zu öffnen. Stellen wir uns vor, die Schüler\*innen bekommen keinen Laptop ausgehändigt, sondern einen

---

<sup>4</sup>[https://kantonsschulen.lu.ch/-/media/Kantonsschulen/Dokumente/Dienststelle/KT\\_LU\\_DGYM\\_Strategie\\_2019.pdf](https://kantonsschulen.lu.ch/-/media/Kantonsschulen/Dokumente/Dienststelle/KT_LU_DGYM_Strategie_2019.pdf), S. 7, aufgerufen am 11. Juli 2023.

Raspberry-Pi-Bausatz. Zuerst müssten sie die Hardwarekomponenten zusammensetzen und danach ein einfaches Betriebssystem installieren. Es sind keine weiteren Programme vorinstalliert und alle benötigten Programme müssen zuerst heruntergeladen und installiert werden. Bei diesem Prozess kann ein allgemeines Verständnis für Betriebssysteme geschaffen werden. Das Backup der Daten wird zuerst manuell über eine externe HD gemacht, später wird pro Klasse ein kleiner Server in Betrieb genommen, welcher das Backup und den Austausch von Daten ermöglicht.

Ein solcher Vorschlag mutet utopisch oder gar fantastisch an. Und wir sind weit von einer Realisierung entfernt. Es würde Anpassungen in der Stunden-dotation erfordern, Lehrpersonen müssten Kompetenzen haben, über welche sie aktuell nicht verfügen. Aber schon beinahe trivial ist die Behauptung, dass eine solche Lösung für das Erreichen des zweiten Ziels sehr viel besser geeignet wäre als es die momentan eingesetzten Lösungen sind. Daraus möchte ich die Einladung ableiten, beim Suchen nach Lösungen auch Alternativen in Betracht zu ziehen und aus diesem kleinen Gedankenexperiment für die Praxis zu lernen.

Dass eine solche Lösung zur Erreichung des zweiten Ziels beiträgt, ist offensichtlich. Inwiefern dies auch für das erste Ziel gilt, hängt von der Umsetzung ab. Die zeitlichen Ressourcen dürften ein gewisses Mass nicht überschreiten und ab einem bestimmten Punkt der Umsetzung müssten bestimmte Anwendungen relativ problemlos zur Verfügung stehen. Unter diesen Voraussetzungen dürfte eine solche Lösung dem Erreichen des ersten Ziels nicht im Wege stehen.

Eine Lösung hingegen, die für das erste Ziel ideal erscheint, weil alles einwandfrei funktioniert, steht dem zweiten Ziel im Weg. Geschlossene Systeme dienen nicht dazu, ein vertieftes Verständnis der verwendeten Technologie zu fördern. Die damit in Verbindung stehende Konsumhaltung der Technologie gegenüber, wonach es in erster Linie wichtig ist, dass die Technologie funktioniert, ist der Förderung einer kritischen Haltung der Technologie gegenüber abträglich. Umgekehrt gilt aber nicht, dass eine Lösung, welche das Erreichen des zweiten Ziels im Fokus hat, dem Erreichen des ersten im Weg stehen muss.

Für das vertiefte Verständnis ist eine Lösung, bei der alles selbst gemacht wird, am besten und eine Alles-aus-einer-Hand-Lösung am schlechtesten, dazwischen gibt es ein breites Feld an Varianten. Die Unterscheidung zwischen freier und proprietärer Software ist begrifflich unabhängig von der soeben besprochenen Unterscheidung zwischen Setzkastenlösungen und Fertiglösun-

gen. Faktisch ist es aber so, dass proprietäre Lösungen zu Fertiglösungen tendieren, während freie Software Setzkastenlösungen erfordert und fördert. Die Entwicklung zu Alles-aus-einer-Hand Lösungen liegt nicht direkt in der Natur proprietärer Software, aber in jener der kapitalistischen Logik vor deren Hintergrund und in deren Rahmen diese entwickelt werden. Das kapitalistische Wirtschaftssystem setzt Anreize zu Grösse und Dominanz am Markt. Eine Methode diese zu erreichen besteht darin, Produktpaletten zu entwerfen, deren Einzelteile so aufeinander abgestimmt und voneinander abhängig sind, dass Konsumentinnen dazu gedrängt werden, mit vielen Produkten eines einzelnen Anbieters oder sogar ausschliesslich mit Produkten dieses Anbieters zu arbeiten.

Umgekehrt steht die Idee freier Software nicht im Widerspruch zu einer Alles-aus-einer-Hand-Lösung, es liegt aber in ihrer Natur, Setzkastenlösungen zu befördern. Mit der expliziten Aufforderung, Software nicht nur zu verwenden, sondern weiterzuentwickeln und zu verbreiten steht sie im konkreten Gegensatz zur Logik der Alles-aus-einer-Hand-Lösung. Abgesehen davon kann generell gesagt werden, dass freie Software ein vertieftes Verständnis stärker fördert als proprietäre Software. Durch die Aufforderung, diese zu studieren, wird dies explizit gemacht. Dagegen liegt es in der Natur proprietärer Software, ihre Funktionsweise so weit als möglich zu verschleiern. Der Einsatz von freier Software ist somit sehr gut geeignet, um das zweite Ziel zu erreichen. Und da es mittlerweile sehr weit entwickelte und erprobte freie Software auf allen Ebenen (Betriebssystem, Software-Pakete, einzelne Anwendungen) gibt, ist es auch realistisch, dass damit auch das erste Ziel erreicht werden könnte.

Eine Lösung, welche für das Erreichen des zweiten Ziels optimal ist, erfordert eine Änderung in der Haltung und Einstellung zu den im Unterricht eingesetzten technischen Mitteln. Solange eine technische Störung als Störung des Unterrichts aufgefasst wird, statt als Möglichkeit, etwas zu lernen, kann der Einsatz der technischen Mittel nur bedingt zu einem vertieften Verständnis der eingesetzten Technologie beitragen. Diese Haltung muss von der Lehrperson verkörpert und vorgelebt werden. Ein bewusster Einsatz von freier Software in der Schule macht diese Haltung sichtbar und fördert eine Einstellung zur Technologie, wonach wir dieser nicht ausgesetzt und ausgeliefert sind, sondern sie verstehen und zu unserem Nutzen einsetzen und gestalten können.

Der Einsatz von freier Software ist geeignet, um die doppelte Zielsetzung für den Einsatz von digitalen Geräten im Unterricht zu erreichen. Deshalb

sollte am Gymnasium mit freier Software gearbeitet werden.

## 3.2 Nicht für die Schule lernt man ...

Gemäss der Maturitätsverordnung soll das Gymnasium die Grundlage für ein lebenslanges Lernen schaffen:

Ziel der Maturitätsschulen ist es, Schülerinnen und Schülern im Hinblick auf ein lebenslanges Lernen grundlegende Kenntnisse zu vermitteln sowie ihre geistige Offenheit und die Fähigkeit zum selbständigen Urteilen zu fördern.<sup>5</sup>

Hintergrund dieser Forderung ist die Vorstellung, dass es sich bei der Bildung nicht um ein abschliessbares Projekt handelt. Dies gilt insbesondere für die gymnasiale Bildung, welche auf wissenschaftliche Tätigkeiten und verantwortungsvolle Positionen in der Gesellschaft vorbereiten soll. Bei der Wissenschaft insgesamt handelt es sich um ein offenes Projekt, welches die Fähigkeit voraussetzt, Neues zu lernen. Ebenso ist die Gesellschaft als Ganzes in stetem Wandel. Eine Bildung, welche für Aufgaben in dieser Gesellschaft vorbereitet, muss diese Offenheit betonen und die Fähigkeit, sich neuen Begebenheiten anzupassen und adäquate Reaktionen zu entwickeln, fördern.

Die Idee des lebenslangen Lernens steckt im Kern der oft bemühten Aussage, dass man nicht für die Schule, sondern für das Leben lerne. Soll eine Schule diesem Anspruch gerecht werden, kann sie ihren Erfolg nicht daran bemessen, einen fixen Kanon von Wissen oder Fähigkeiten vermittelt zu haben, sondern daran, ob es gelingt, dass die Schüler\*innen in der Lage sind, weiter zu lernen und neugierig zu bleiben. Eine solche Schule muss die Neugierde fördern. Und in einer solchen Schule muss gelernt werden zu lernen.

Dennoch: Lernen ist immer konkret. Auch Lernen lernen heisst deshalb zuerst, etwas konkretes lernen. Lernen lernen heisst konkrete Lernerfahrungen zu machen. Deshalb vermittelt auch eine Schule, welche auf lebenslanges Lernen vorbereitet, konkrete Inhalte und trainiert bestimmte Tätigkeiten. Deren Charakter ist jedoch exemplarisch und die Auswahl letztlich arbiträr.

Um Schülerinnen und Schüler für eine sich im Wandel befindende Gesellschaft und eine sich stetig weiterentwickelnde Wissenschaft vorzubereiten

---

<sup>5</sup>[https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1995/1001\\_1001\\_1001/20130101/de/pdf-a/fedlex-data-admin-ch-eli-cc-1995-1001\\_1001\\_1001-20130101-de-pdf-a.pdf](https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1995/1001_1001_1001/20130101/de/pdf-a/fedlex-data-admin-ch-eli-cc-1995-1001_1001_1001-20130101-de-pdf-a.pdf), aufgerufen am 11. Juli 2023.

muss kein fixer Kanton an Wissen und Fähigkeiten vermittelt werden, sondern die Kompetenz, sich neues Wissen und neue Fähigkeiten anzueignen, Bestehendes weiterzuentwickeln und anzupassen. Eine zentrale Rolle spielt dabei die Fähigkeit, kritisch denken zu können. Sie ermöglicht es, Bestehendes zu hinterfragen und analysieren und Neues zu bewerten und einordnen zu können. Kritisches Denken hat im Rahmen der gymnasialen Bildung nicht nur einen Wert an sich, sondern ihr kommt auch instrumenteller Charakter zu, insofern sie eine Grundvoraussetzung für lebenslanges Lernen ist.

Kritisch denken heisst Unterscheiden und Einordnen. Dies kann nur vor dem Hintergrund von bereits Akzeptiertem und auf der Grundlage von bereits Feststehendem geschehen. Auch deshalb muss eine auf lebenslanges Lernen ausgerichtete Schule konkrete Inhalte, Fähigkeiten und Werte vermitteln. Dies muss jedoch immer im Bewusstsein darum geschehen, dass auch das Feststehende und Akzeptierte immer nur als provisorisch betrachtet werden kann. Dies geschieht dadurch, dass die vermittelten Inhalte hinterfragt werden, dass verglichen wird und immer auch die Genese von Wissen aufgezeigt wird.

Die Wissenschaft und die Gesellschaft sind lebendige Systeme. Die Schule, die auf diese vorbereitet, muss diesen lebendigen Charakter vermitteln und erlebbar machen, dass es sich bei beiden um offene Projekte handelt, bei denen sich einzubringen möglich und lohnenswert ist.

Die Fähigkeit zu lebenslangem Lernen ist notwendig, weil sich die Umstände, in denen das Leben stattfindet, verändern. Ein Haupttreiber dieser Veränderung ist die Entwicklung von Computern und ihr zunehmender Einsatz in praktisch allen Bereichen des Lebens. Computer können aber nicht nur eine Ursache für die Notwendigkeit zu lebenslangem Lernen sein, sondern auch ein besonders geeignetes Anwendungs- und Übungsfeld dafür bieten. Das Arbeiten mit Computern setzt bestimmte Fähigkeiten und Kenntnisse voraus. Technische Entwicklungen, nur schon in der Zeitspanne einer gymnasialen Schulkarriere, zeigen auf, dass diese immerfort angepasst und weiterentwickelt werden müssen. Dies erfordert nicht nur konkrete Anwendungsfähigkeiten zu beherrschen, sondern ein grundlegendes Verständnis der Technologie zu entwickeln, welches eine Weiterentwicklung der Anwendungsfähigkeiten ermöglicht.

Anwendungswissen und grundlegendes Verständnis gehen Hand in Hand und werden im gemeinsamen Wechselspiel erworben. Das Arbeiten mit konkreten Anwendungen fördert das Verständnis, ein vertieftes Verständnis ermöglicht das Entwickeln weiterer Anwendungsfähigkeiten. Die beiden Arten

von Fähigkeiten können begrifflich nicht klar getrennt werden. Allgemein gilt aber, dass die Grundlagen relativ konstant sind und ihre Kenntnis die Voraussetzung bildet, um Anpassungen im Anwendungswissen vorzunehmen und dieses zu erweitern. Die Arbeit mit Computern bildet also ein Übungsfeld für das Zusammenspiel von konkretem Wissen und allgemeineren Kompetenzen, welche den Erwerb und die Entwicklung von neuem Wissen ermöglichen und somit die Grundlage für lebenslanges Lernen bilden.

Computertechnologie entwickelt sich schnell und macht es notwendig, neue Fähigkeiten zu erwerben oder bereits erworbene anzupassen und weiterzuentwickeln. Dies gilt allgemein, im Bereich von Software gilt es für proprietäre und freie Software im gleichen Mass. Aber freie Software ist darüber hinaus in besonderem Mass geeignet, diese Entwicklung zu schulen. Freie Software kann studiert werden, Veränderungen in der Software können nachvollzogen werden. Dies führt zu einem vertieften Verständnis der Grundlagen.

Insofern Freie Software ein vertieftes Verständnis der Grundlagen fördert, trägt ihr Einsatz in der Schule zur Ausildung einer Basis bei, welche der fortwährende Erwerb neuer Anwendungsfähigkeiten ermöglicht. Und darüber hinaus macht es erlebbar was es heisst, auf der Grundlage von bestehendem Wissen und Fähigkeiten auch selbständig neue Fähigkeiten zu entwickeln und sich neues Wissen anzueignen. Der Einsatz von freier Software unterstützt die Vorbereitung auf ein lebenslanges Lernen. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

### **3.3 Wissenschaftspropädeutik**

Das Gymnasium befähigt seine Schüler\*innen zu einem Hochschulstudium und ist somit eine erste Etappe der Wissenschaftspropädeutik. Im Sinne der Wissenschaftspropädeutik gibt es zwei Argumente für den Einsatz freier Software am Gymnasium.

In zumindest einigen Wissenschaften ist die Verwendung freier Software weit verbreitet. So wird in vielen naturwissenschaftlich geprägten Studienfächern mit der Statistiksoftware R gearbeitet, Python ist in vielen Wissenschaften für die unterschiedlichsten Zwecke verbreitet. Und die meisten Universitäten in der Schweiz arbeiten mit dem Learning Management System Moodle. Die Verwendung freier Software am Gymnasium ist eine Vorbereitung auf deren Einsatz an den Universitäten. Dieses Argument ist zumindest gleich stark wie das oft zitierte Argument für den Einsatz bestimmter pro-



prietärer Produkte, dass diese an den Universitäten verwendet werden und deshalb die Fähigkeit in ihrem Umgang vorausgesetzt wird. Beide Argumente scheinen insgesamt nicht sehr überzeugend zu sein. Die Schule sollte sich nicht zur Schulung konkreter Tätigkeiten verpflichtet fühlen, sondern das Vertrauen haben, dass die von ihr vermittelten Kompetenzen den Umgang mit diesen ermöglichen. Das Argument sei hier aber genannt als Entkräftigung eines Gegenargumentes.

Zur Wissenschaftspropädeutik gehören neben dem Vermitteln von inhaltlichen und transversalen Kompetenzen auch das Einüben von wissenschaftlichem Arbeiten und das Vermitteln einer Grundhaltung, welche für das wissenschaftliche Arbeiten konstituierend ist. Zu den wissenschaftlichen Arbeitsweisen gehören unter anderem der Umgang mit Quellen und ein Verständnis für geistiges Eigentum. Verbunden damit ist ein Verständnis, dass wissenschaftlicher Fortschritt ein generationenübergreifendes Gemeinschaftsprojekt ist. Jeder Erfolg in der Wissenschaft kann nur in Zusammenarbeit mit anderen passieren. Dazu gehört auch ein Verständnis von Wissenschaft als einem offenen Projekt, wonach sich eine konkrete wissenschaftliche Arbeit einreicht in ein Kontinuum von Arbeiten und Einflüssen, weitzurückreichend in die Menschheitsgeschichte und immer offen in die Zukunft weisend. Weiter gehört dazu eine bewusst gelebte Fehlerkultur. Es ist ein verbreitetes wissenschaftliches Prinzip, alle Resultate zu veröffentlichen, auch die Daten zu nicht erfolgreichen Versuchen zu veröffentlichen und Fehlversuche zu dokumentieren. Weil auch Fehler zu Einsichten führen und Ursprung neuer Ansätze sein können. Wissenschaftliches Arbeiten ist durch spezifische Anreize motiviert. Zumindest ein Teil des Anreizes besteht darin, einen Beitrag zu einem grösseren Projekt zu leisten und etwas Neues zu schaffen. Diese möglicherweise etwas romantisierte Vorstellung von wissenschaftlichem Arbeiten und der Motivation dafür ist sicher nicht in allen Fällen zutreffend und leitend. Aber wenn es darum geht, möglichst allgemein zu beschreiben, was wissenschaftliches Arbeiten auszeichnet, so gehört dieser Aspekt dazu.

All diese Haltungen und Arbeitsweisen zeigen sich exemplarisch in der Kultur freier Software. Freie Software widerspiegelt und verkörpert diese Haltungen. Dies hat damit zu tun, dass freie Software sehr stark aus einem wissenschaftlichen Millieu entstammt und dass die kollaborative Entwicklung freier Software und die dafür verwendeten Organisationsformen ähnlich ist wie das Arbeiten in den Wissenschaften. Die Verwendung und damit einhergehende Thematisierung freier Software am Gymnasium könnte diese auch für die Wissenschaften fundamentalen Haltungen und Prinzipien exempla-

risch vermitteln.

Freie Software verkörpert die Ideale und Arbeitsweisen der Wissenschaft und ihr Einsatz dient somit der Wissenschaftspropädeutik. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

### 3.4 Gesellschaftsreife

Neben der allgemeinen Studierfähigkeit wird eine vertiefte Gesellschaftsreife als das zweite übergeordnete Ziel einer gymnasialen Bildung definiert. Artikel 5 der Maturitätsverordnung aus dem Jahr 1995 hält fest:

Maturandinnen und Maturanden finden sich in ihrer natürlichen, technischen, gesellschaftlichen und kulturellen Umwelt zurecht, und dies in Bezug auf die Gegenwart und die Vergangenheit, auf schweizerischer und internationaler Ebene. Sie sind bereit, Verantwortung gegenüber sich selbst, den Mitmenschen, der Gesellschaft und der Natur wahrzunehmen.<sup>6</sup>

Um die geforderte Verantwortung in einer digitalisierten Gesellschaft wahrnehmen zu können, wird ein vertieftes Verständnis für die die Gesellschaft prägenden Technologien vorausgesetzt. Um beurteilen zu können, welche Technologien eingesetzt werden sollen, wie diese eingesetzt werden sollen, welche Technologie wie entwickelt werden soll, braucht es ein vertieftes Verständnis für das Funktionieren dieser Technologien und die Fähigkeiten, diese zu beurteilen und zu gestalten. Es genügt ausdrücklich nicht, Technologie nur als Konsument\*in benutzen zu können. Die zu treffenden Entscheidungen setzen ein Verständnis der Technologie voraus, welche über die reine Anwendungskompetenz hinausgeht, da es ja eben diese Anwendung zu hinterfragen, beurteilen und bewerten gilt.

Zum öffentliche Raum einer digitalisierten Gesellschaft gehört das Internet wesentlich dazu. Der Kampf für eine freie Gesellschaft und für Teilhabe wird auch ein Kampf um die Freiheit des Internets sein. Die Möglichkeiten zu einem freien und selbstbestimmten Leben hängen deshalb entscheidend davon ab, ob das Internet als ein Ort der Freiheit und der Teilhabe funktioniert. Die bis anhin bestehende Freiheit im Internet verdankt sich einer

---

<sup>6</sup>[https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1995/1001\\_1001\\_1001/20130101/de/pdf-a/fedlex-data-admin-ch-eli-cc-1995-1001\\_1001\\_1001-20130101-de-pdf-a.pdf](https://fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/cc/1995/1001_1001_1001/20130101/de/pdf-a/fedlex-data-admin-ch-eli-cc-1995-1001_1001_1001-20130101-de-pdf-a.pdf), aufgerufen am 11. März 2023.

dezentralen Organisation. Das Internet wird nicht durch eine zentrale Instanz kontrolliert, sondern Protokolle organisieren die Kommunikation. Zudem ist das Internet ausschliesslich auf freier Software basiert, was Transparenz sicherstellt. Es gibt verschiedene Kräfte die versuchen, diese einzuschränken, dazu gehören sowohl staatliche wie auch private und kommerzielle Akteure. Die Motive für die Einschränkung der Freiheit im Internet mögen teilweise begründet sein, jedoch ist es wichtig, dass eine Debatte darüber geführt wird. Um an dieser Debatte teilnehmen zu können und Lösungsvorschläge einordnen und beurteilen zu können, braucht es ein Verständnis der Technologien und Prinzipien, welche das Internet ausmachen. Und das heisst insbesondere, es braucht Kenntnis der auf freier Software und dezentralen Protokollen aufgebauten Infrastruktur.

Der Einsatz von freier Software in der Schule führt zu einem vertiefteren Verständnis der Technologie, welches diese Beurteilung ermöglicht. Der Einsatz freier Software in der Schule führt zu einem Umgang mit Software, in welcher nicht fertige Produkte eingesetzt werden, sondern Lösungen selber gefunden werden müssen und zusammengestellt werden. Dabei geht es nicht in erster Linie um konkrete technische Fähigkeiten, sondern um Haltungen gegenüber Technologie. In Anbetracht der verschiedenen Herausforderungen, welchen sich die Gesellschaft gegenüber sieht, ist es notwendig, Technologie positiv einzusetzen. Dies setzt eine Haltung voraus, welche der Technologie dieses Potential grundsätzlich zutraut. Im Bewusstsein darum, dass Technologie genauso gut zu negativen Entwicklungen beitragen kann und negative Entwicklungen beschleunigen kann, braucht es eine kritische Haltung gegenüber der Technologie und die Einstellung, dass die Richtung der technologischen Entwicklung beeinflusst werden kann. All diese Aspekte finden sich in der Kultur der freien Software verankert. Es braucht aber auch die Erfahrung, dass ein Einfluss auf die technologische Entwicklung genommen werden kann. Der Einsatz von freier Software könnte exakt diese Erfahrung ermöglichen.

Der Einsatz freier Software trägt zur Bildung jener Haltungen und Kompetenzen bei, welche Voraussetzung für eine verantwortungsvolle Rolle in der Gesellschaft sind. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

## 3.5 Reife Gesellschaft

Mitgestaltung einer digitalisierten Gesellschaft setzt vertiefte technologische Kenntnisse voraus, welche durch den Einsatz freier Software gefördert werden kann. Die im letzten Kapitel vorgetragene Argumentation forderte die Verwendung freier Software in der Schule, weil dies einen Beitrag zur Gesellschaftsreife leisten kann. Daraus, wie diese Gesellschaft organisiert ist und funktioniert, lässt sich ableiten, welche Kompetenzen eine Schule vermitteln muss. Die geforderte Fähigkeit zur aktiven Teilhabe an der Gesellschaft bietet die normative Prämisse des Argumentes, die faktische Gesellschaftsordnung und die Bedingungen zur Teilhabe dienen als deskriptive Prämissen.

Im Zusammenhang mit der Gesellschaftsreife lässt sich noch eine andere Argumentation für die Verwendung freier Software an der Schule entwickeln: Wenn die Feststellung der Gefährdung der Möglichkeit der Teilhabe als deskriptive Prämisse formuliert wird, kann zusammen mit der normativen Prämisse, wonach die Schule die Möglichkeit zur Teilhabe sicherstellen soll, abgeleitet werden, dass die Schule nach ihren Möglichkeiten die Entwicklung der Gesellschaft in eine Richtung beeinflussen soll, welche diese Teilhabe überhaupt möglich macht. Und insofern gezeigt werden kann, dass die Verwendung freier Software an der Schule einen Beitrag zu dieser Entwicklung leistet, kann diese damit begründet werden.

Es gibt viele Formen der aktiven Teilhabe an der Gesellschaft. Dies reicht von der Teilnahme an politischen Prozessen über die Mitwirkung in weniger formellen Gremien und Vereinen. Insgesamt kann dieser Bereich, in welchem Mitgestaltung möglich ist, als öffentlicher Raum bezeichnet werden. Die Grenzen dieses Raums können nicht klar gezogen werden. Der öffentliche Raum überlappt mit Bereichen, welche von privaten oder kommerziellen Akteuren kontrolliert werden und in welchen keine Mitgestaltung möglich ist. Ohne den Bereich des öffentlichen Raums genau eingrenzen zu können, kann doch festgehalten werden, dass es einen solchen Bereich gibt und dass das Vorhandensein eines solchen Bereichs notwendig ist für die Möglichkeit der Mitgestaltung.

Wenn sich die gesellschaftliche Aktivität zunehmend in digitale Räume verlagert, besteht die aktive Teilhabe an der Gesellschaft in der Mitgestaltung dieser digitalen Räume. Der schnelle Takt digitaler Erneuerungen eröffnet zwar immer wieder Gestaltungsmöglichkeiten für Einzelne, gleichzeitig findet im digitalen Raum eine Konzentration der Macht statt, die keinen Raum für institutionalisierte und offene Formen der Mitgestaltung lässt. Die gros-

sen Big-Tech-Konzerne vereinen sehr viel Macht in ihren Händen. Sie stellen die Infrastruktur zur Verfügung, welche sie dann auch kontrollieren. Diese Infrastruktur wird immer komplexer und ressourcenintensiver, so dass sich eine zunehmende Konzentration abzeichnet. Wenn immer mehr Bereiche des Lebens in diesen Räumen stattfinden, so ist die Möglichkeit zur Mitgestaltung immer kleiner. Die etablierten Kanäle der Mitgestaltung finden keinen Zugang zu diesen Räumen. Die durch die Digitalisierung neu entstandenen Räume sind oftmals keine öffentlichen Räume. Die Möglichkeit zur aktiven Teilhabe und Mitgestaltung nimmt ab.

Aktuell zeigt sich dies mit der Entwicklung von KI-Chatbots wie ChatGPT. Es ist unbestritten, dass der Einsatz von KI in sehr vielen Hinsichten unsere Gesellschaft stark prägen und verändern wird in den kommenden Jahren. Auch wenn es der Fall ist, dass Deep Learning nicht mehr nachvollziehbar ist, so ist es doch nicht so, dass dieses einfach so passiert. Jeder KI liegen Modelle zugrunde und die verwendeten Trainingsdaten und Algorithmen beeinflussen, wie die Ergebnisse letztlich aussehen. Wer die Modelle, Trainingsdaten und Algorithmen wählt und gestaltet, hat Kontrolle über die Ergebnisse. Daraus resultiert eine grosse Macht für jene, welche diese Kontrolle ausüben. Um dem entgegenzuwirken bräuchte es offene KI Systeme. Dafür plädiert beispielsweise auch KI-Forscher und Präsident der EPFL Martin Vetterli:

Wir können uns nicht davon abhängig machen, ob Unternehmen Lust verspüren, etwas zu stoppen. KI ist zu gross, um sie allein den Unternehmen zu überlassen. Unsere Idee ist, Transparenz zu schaffen – und die Sache selbst in die Hand zu nehmen. [...] In Zusammenarbeit mit anderen Universitäten wollen wir ein Open-Source-AI-Projekt starten, bei dem offene Daten und ein offener Quellcode verwendet werden, um LLMs zu trainieren. Wir müssen unbedingt Standards für Transparenz und Fairness setzen.<sup>7</sup>

Am Beispiel von KI zeigt sich deutlich: Die Gefahr für die Gesellschaft geht nicht direkt von der Technologie aus, sondern von der Konzentration von Macht in den Händen jener, welche diese Technologie kontrollieren. Offene Systeme können dem entgegenwirken. Offene Systeme werden entstehen, wenn eine Nachfrage danach besteht. Schulen können sich weigern, intransparente Systeme zu verwenden. Wenn sie stattdessen auf offene Alternativen

---

<sup>7</sup>Magazin des TagesAnzeigers, 12. Mai 2023.

setzen, so vermitteln sie ihren Schüler\*innen nicht nur, dass es Alternativen gibt, sondern tragen auch aktiv zum Entstehen solcher Alternativen bei.

Der vermehrte Einsatz freier Software kann ein Gegenmittel gegen drohende Einschränkungen der Freiheit und Mündigkeit bilden. Verbreiteter Einsatz von freier Software kann zur Entwicklung einer auf Freiheit basierten und auf Teilnahme orientierter Gesellschaft beitragen. Die Schule wiederum kann zur Verbreitung und Entwicklung freier Software beitragen, indem es diese einsetzt. Würden Schüler\*innen in der Schule automatisch mit freier Software in Kontakt kommen oder für schulische Zwecke gar ausschliesslich freie Software verwenden, würde dies sehr nachhaltig zu einer Verbreitung von freier Software führen.

Die Schule kann sich nicht darauf beschränken, Schüler\*innen auf die Teilhabe in einer freien Gesellschaft vorzubereiten, wenn die zukünftige Möglichkeit der Teilhabe bedroht ist. Wenn die Schule eine Möglichkeit hat, die Entwicklung der Gesellschaft in positiver Richtung zu beeinflussen, so sollte sie dies tun. Freie Software hat das Potential zu einer freiheitlichen und auf Teilhabe ausgerichteten Gesellschaft beizutragen. Deshalb sollte die Schule die Entwicklung freier Technologien fördern, indem sie diese verwendet und die Schüler\*innen mit ihnen vertraut macht. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

## 3.6 Digital Commons

Das Argument des letzten Kapitels forderte die Schule auf, freie Software zu verwenden, um diese zu fördern und so zum Entstehen von Alternativen zu den durch die Big-Tech-Konzerne kontrollierten Systemen beizutragen. Dies kann sich darin erschöpfen, diese zu verwenden und so zur Verbreitung beizutragen. Es könnte aber darüber hinaus auch bedeuten, dass sich Schulen, Kantone oder andere staatliche Organe direkt an der Entwicklung von freier Software beteiligen. Dies führt zur Fragestellung, ob staatliche Organe direkt in die Entwicklung von Software investieren sollen. In diesem Kapitel wird dafür argumentiert, dass es als Auftrag von Schulen oder anderer staatlichen Organe betrachtet werden kann, in die Entwicklung einer digitalen Infrastruktur zu investieren.

Nicht nur im Bildungswesen, sondern in vielen Bereichen staatlicher Institutionen findet eine starke Digitalisierung von Prozessen statt. Dies ist in jedem Fall mit grossen finanziellen Investitionen verbunden. Dabei wer-

den Lizenzen für Produkte grosser Softwareanbieter gekauft. In vielen Fällen müssen zusätzlich Firmen beauftragt werden, die Software in Betrieb zu nehmen oder zu adaptieren. So hat der Kanton Luzern im Bildungsbereich Lizenzen für Microsoft 365 - für das Einrichten des Sharepoints und die Migration von Daten auf den Sharepoint musste eine weitere Firma hinzugezogen werden, und dies obwohl es eine grosse kantonale Informatikdienststelle gibt.<sup>8</sup>

Zudem ist beim Einsatz von fixen Softwareprodukten nicht gewährleistet, dass diese die spezifischen Anforderungen erfüllen. Dies zeigt sich deutlich beim Einsatz der Microsoft 365 Produkte im Bildungsbereich, welche für den Bürogebrauch entwickelt wurden. Zwar wurden einige Anpassungen für den Einsatz in Schulen vorgenommen, aber es bleibt deutlich, dass diese nicht für den Einsatz in der Bildung entwickelt worden sind. Da es sich um Produkte eines weltweit tätigen Konzerns handelt, ist es nur sehr begrenzt möglich, Einfluss auf deren Weiterentwicklung zu nehmen und da es sich um proprietäre Software handelt, gibt es keine Möglichkeiten, diese selbst zu adaptieren.

Das für Lizenzen und die Einrichtung von lizenzierten Produkten ausgegebene Geld ist nicht nachhaltig investiert. Beim Kanton wird kein Wissen gebildet, welches für weitere Einsätze genutzt werden könnte. Es entsteht ein starkes Abhängigkeitsverhältnis. Und es besteht auch keine Garantie, dass nicht zu einem bestimmten Zeitpunkt die Lizenzgebühren stark angehoben werden. Zudem besteht immer das Risiko, dass die Produkte nicht mehr weiter entwickelt oder unterstützt werden. Alle Investitionen im Zusammenhang mit diesen Produkten, insbesondere die entwickelten Unterrichtsszenarien und Materialien wären nicht mehr zu gebrauchen.

Dass in Lizenzgebühren investierte Geld führt nicht zur Bildung von IT-Kompetenzen bei den kantonalen Stellen. In den letzten Jahren konnten wir immer wieder beobachten, wie mangelnde Kompetenzen im digitalen Bereich dazu führten, dass grosse Softwareprojekte der öffentlichen Hand als Debakel endeten und horrenden Kosten verursachten.<sup>9</sup>

---

<sup>8</sup>Die Finanzierung ist darüberhinaus äusserst intransparent. Mir ist es auch mit viel Aufwand nicht gelungen in Erfahrung zu bringen, welche Kosten für das Bildungswesen im Kanton Luzern für den Erwerb der Microsoft Lizenzen und für deren Einsatz anfallen.

<sup>9</sup>Beispielsweise das Luftüberwachungssystem Skyview der Schweizer Armee, welches mehr als doppelt soviel kostete wie budgetiert: <https://www.netzwoche.ch/news/2022-11-24/it-projekt-der-armee-wird-doppelt-so-teuer-wie-gedacht>. Oder die Schulsoftware Educase, welche im Kanton Luzern für grosse Probleme sorgte, welche unter anderem auf fehlende IT-Kompetenzen auf Seiten der kantonalen Verwaltung zurückzuführen sind: <https://www.luzernerzeitung.ch/zentralschweiz/luzern/vor>

Stattdessen sollte der Staat versuchen, in nachhaltige Softwareprojekte zu investieren und diese selber oder in Kooperation mit anderen Institutionen zu entwickeln. Dies hätte den Vorteil, dass die Produkte unter Berücksichtigung der spezifischen Bedürfnisse konzipiert und weiterentwickelt werden könnten. Es stellt sicher, dass die Produkte längerfristig nutzbar sind. Und es führt dazu, dass die dabei erworbenen Kompetenzen bei den staatlichen Institutionen gebildet werden.

Die von staatlicher Stelle entwickelte oder mitentwickelte Software müsste freie Software sein. Es sollte als eine Investition des Staates in digitale Commons verstanden werden, wie es bereits in bestimmten Bereichen passiert. Das beste Beispiel dafür ist die Investition in Bildung und Forschung. Investitionen in Bildung und Forschung sind exemplarisch für Investitionen, deren Früchte von allen genutzt werden können. Analog dazu sollten auch die Investitionen des Staates in die Entwicklung von freier Software verstanden werden. Öffentliches Geld sollte in öffentlichen Code investiert werden: *Public Money, public Code!*<sup>10</sup> Von diesen Entwicklungen können andere profitieren - und wenn sie diese als Basis weiterer Entwicklungen verwenden, kann zu einem späteren Zeitpunkt eine staatliche Institution wiederum davon profitieren.<sup>11</sup>

Schulen könnten hier eine entscheidende Rolle spielen. Im Bildungsbereich gibt es eine grosse Nachfrage nach geeigneter Software, welche von proprietären Lösungen bisher nur teilweise abgedeckt wird. Schulen, Kantone oder andere Institutionen könnten selbst bestehende freie Software adaptieren oder sich an deren Entwicklung beteiligen. Wenn dies zuerst die vorhandenen Kompetenzen übersteigt, so können zuerst externe Anbieter miteinbezogen werden, allerdings immer mit dem Ziel, die entsprechenden Kompetenzen bei den Behörden selbst aufzubauen.

Ein gutes Beispiel wie dies tatsächlich funktionieren kann ist der Einsatz von Moodle. Moodle ist freie Software unter einer GNU GPL.<sup>12</sup> Plugins für Moodle werden von sehr unterschiedlichen Stellen geschrieben, unter anderem Universitäten. Momentan ist der Kanton Luzern daran, Moodle für die Gymnasien einzuführen. Auch wenn es sich um freie Software handelt, ist

---

stoss-pannensoftware-fuer-luzerner-schulen-jetzt-sollen-externe-fachleute-projekt-pruefen-ld.2236108.

<sup>10</sup>Genau dies ist die Forderung einer Initiative der Free Software Foundation Europe: <https://publiccode.eu/de/>, aufgerufen am 14. Juli 2023.

<sup>11</sup>Bollier, „Protecting and Enlarging the Digital Republic“, S. 8.

<sup>12</sup><https://docs.moodle.org/dev/License>.



dies nicht gratis. Es wird unter anderem ein externer Anbieter hinzugezogen, welcher die Login-Daten vorbereitet. Und es braucht einigen Aufwand der kantonseigenen Dienststelle Informatik für das Aufsetzen und Betreiben des Servers. Aber die Erfahrungen, welche hierbei gemacht werden, bleiben im Kanton. Zukünftig kann Moodle mit relativ wenigen Mitteln betrieben werden. Das sind nachhaltige Investitionen.

Zudem besteht bei Moodle im Vergleich zu proprietärer Software eine grössere Sicherheit, dass diese auch in Zukunft unterstützt und weiterentwickelt wird. Die Organisationen, welche an der Entwicklung von Moodle arbeiten, sind zu einem grossen Teil gerade die Bildungsinstitutionen, welche Moodle auch verwenden. Es ist zudem vorstellbar, dass darüber hinaus in die Entwicklung von Moodle investiert wird. Wenn ein spezifisches Bedürfnis nach einem Service besteht, ist es möglich, dass der Kanton selbst oder in Zusammenarbeit mit anderen Institutionen entsprechende Software schreibt.

Statt in Lizenzen für Bürosoftware zu investieren, sollte besser in die Entwicklung von Software investiert werden, welche auf die spezifischen Bedürfnisse von Schulen ausgerichtet ist. Dabei entstehen digitale Kompetenzen im Kanton und an der Schule, welche für die zukünftige Entwicklung von Softwareprojekten wichtig sein wird und zur längerfristigen Wahrung der Autonomie der Bildung beiträgt. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

### 3.7 Nachhaltigkeit

Bildung zur Nachhaltigkeit ist ein Ziel der gymnasialen Bildung.<sup>13</sup> Zur Nachhaltigkeit gehört insbesondere der Umgang mit Ressourcen. In einer digitalisierten Welt gehören die Herstellung und Verwendung von digitalen Geräten zu den bedeutendsten Verbrauchsressourcen. Ein möglichst schonender Umgang mit diesen Ressourcen ist entsprechend entscheidend. Bildung zur Nachhaltigkeit muss dies vermitteln. Es genügt aber nicht, dies als Unterrichtsinhalt theoretisch zu behandeln, sondern es muss auch versucht werden, den Einsatz digitaler Geräte an der Schule möglichst nachhaltig zu gestalten.

Die Digitalisierung ist ein grosser Verbraucher von Ressourcen und ein Treiber von steigendem Energiebedarf. Gemäss einer Studie des französischen Think-Tanks The Shift Project beträgt der Energiebedarf der weltweiten di-

---

<sup>13</sup><https://education21.ch/de/bne>, aufgerufen am 14. Juli 2023.

gitalen Infrastruktur mehr als jener von den meisten Ländern der Welt.<sup>14</sup> Einerseits brauchen die Endgeräte selbst Strom, wobei dies ein vergleichsweise kleiner Anteil des durch die Digitalisierung anfallenden Ressourcenverbrauchs ist. Der grössere Anteil der verbrauchten Ressourcen entfällt auf die Herstellung der Geräte. Und in zunehmendem Mass werden sehr viele Ressourcen für den Betrieb der Infrastruktur für alle jene Dienstleistungen verbraucht, welche auf eine Serverinfrastruktur aufbauen. In allen drei Bereichen könnte der vermehrte Einsatz von freier Software zu einem geringeren Ressourcenverbrauch und mehr Nachhaltigkeit beitragen.

Oftmals ist es der Fall, dass freie Software auf weniger leistungsstarken Geräten funktioniert und ihr Betrieb somit weniger Energie benötigt, als vergleichbare proprietäre Lösungen. Die Geräte, welche wir verwenden, sind immer leistungsstärker und schneller. Die Anwendungen, welche wir auf ihnen im schulischen Kontext ausführen, sind aber grossmehrheitlich die gleichen und sie laufen auch in der gleichen Geschwindigkeit ab. Das heisst letztlich auch, dass wir immer mehr Energie brauchen, um die gleichen Prozesse auszuführen. Der Einsatz von freier Software könnte dem entgegenwirken, da es den Einsatz von weniger leistungsstarken und somit ressourcenschonenden Geräte ermöglichen würde. Zudem bietet die Entwicklung freier Software für den Einsatz an Schulen die Möglichkeit, die verwendete Energie als ein Kriterium für die Entwicklung zu formulieren.

Weil ein grosser Teil des Ressourcenverbrauchs eines digitalen Gerätes auf seine Herstellung entfällt, sollten diese Geräte möglichst lange verwendet werden können. Die durchschnittliche Lebensspanne von digitalen Geräten ist aber sehr tief. Ein wichtiger Faktor für die kurze Lebensspanne digitaler Geräte ist das Zusammenspiel von Soft- und Hardware. Software wird häufig erneuert und neuere Versionen von Software brauchen mehr Ressourcen und leistungsstärkere Geräte. Das führt dazu, dass für ältere Geräte kein Softwareupdate mehr gemacht werden kann oder neuere Software nicht mehr eingesetzt werden kann. Dies kann die Beschaffung eines neuen Gerätes notwendig machen, auch wenn das alte Gerät noch funktionieren würde. Diese Folge des Zusammenspiels zwischen Hard- und Software ist von den Herstellern so gewollt, da es zum Verkauf von mehr Geräten führt.

Die kapitalistische Wachstumslogik führt dazu, dass Geräte oft ersetzt werden und somit sehr viele Ressourcen verbraucht werden. Da die Ent-

---

<sup>14</sup>[https://theshiftproject.org/wp-content/uploads/2019/03/Lean-ICT-Report\\_The-Shift-Project\\_2019.pdf](https://theshiftproject.org/wp-content/uploads/2019/03/Lean-ICT-Report_The-Shift-Project_2019.pdf).

wicklung freier Software relativ unabhängig von direkten Marktinteressen ist, unterliegt sie dieser Wachstumslogik in viel geringerem Mass. Im Vergleich zu proprietärer Software ist die Möglichkeit auch für ältere Versionen freier Software Updates durchzuführen grösser und neuere Versionen laufen auch auf weniger starken Geräten. Der Einsatz von freier Software kann somit sehr direkt zu einem schonenderen Umgang mit Ressourcen beitragen, indem Geräte länger eingesetzt werden können.

Anbieter proprietärer Software bevorzugen zunehmend Modelle, bei welchen die Programme nicht lokal auf dem Gerät der Benutzer\*innen, sondern auf Servern der Firma ausgeführt werden (Software as a Service). Für die Benutzer\*innen hat dies den Vorteil, dass sie die Software nicht auf ihrem Gerät installieren müssen (und theoretisch auch, dass sie ein weniger leistungsstarkes Gerät verwenden könnten). Insgesamt ist aber der Ressourcenverbrauch dadurch grösser. Die ständige Synchronisation aller Daten ist sehr energieintensiv. Die Software proprietärer Anbieter ist zunehmend auf die ständige Verbindung mit Clouddiensten ausgerichtet, auch wenn dies für den Einsatz in der Schule oftmals nicht notwendig wäre. Dies führt zu einem unnötig hohen Energieverbrauch.

Beispielsweise werden alle Daten der Schüler\*innen ständig mit der Cloud von Microsoft synchronisiert. Selbstverständlich ist es wichtig, dass die Daten der Schüler\*innen gesichert sind. Aber dies könnte auch auf anderem Weg erreicht werden, welcher nicht mit einer automatischen und ständigen Synchronisation einherginge. Dies hätte zudem den Vorteil, dass sich die Schüler\*innen mit der Sicherung ihrer Daten aktiv auseinandersetzen müssen und dabei auch ein Bewusstsein für die mit der Datenspeicherung einhergehenden Themen wie Privatsphäre und Datenschutz entwickeln.

Der Einsatz freier Software bietet die Möglichkeit, Server nur für jene Aufgaben zu verwenden, in denen dies einen Mehrwert bietet und die restlichen Aufgaben lokal zu bewältigen. Dies würde viel weniger Energie verbrauchen.

Der Einsatz freier Software könnte direkt dazu beitragen, dass der Einsatz digitaler Geräte an der Schule weniger ressourcenintensiv wäre. Damit wird ein direkter Beitrag zur Nachhaltigkeit der Bildung geleistet. Viel entscheidender aber ist, dass dadurch ein ressourcenschonender Umgang vorgelebt und vermittelt werden kann und somit ein grosser Beitrag zur Bildung zur Nachhaltigkeit geleistet wird. Der Einsatz von freier Software in der Schule ist somit nachhaltig und konkret gelebte Bildung zur Nachhaltigkeit. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

## 3.8 Sapere Aude!

Unser Selbstverständnis als Bildungseinrichtung steht in der Tradition der Aufklärung und unsere Bildungsziele sind den Werten der Aufklärung verpflichtet. Eine berühmte und oft bemühte Definition von Aufklärung stammt von Immanuel Kant:

Aufklärung ist der Ausgang des Menschen aus seiner selbst verschuldeten Unmündigkeit. Unmündigkeit ist das Unvermögen, sich seines Verstandes ohne Leitung eines anderen zu bedienen. Selbstverschuldet ist diese Unmündigkeit, wenn die Ursache derselben nicht am Mangel des Verstandes, sondern der Entschliessung und des Mutes liegt, sich seiner ohne Leitung eines anderen zu bedienen. “Sapere aude! Habe Mut dich deines eigenen Verstandes zu bedienen!” ist also der Wahlspruch der Aufklärung.<sup>15</sup>

Die Aufklärung versteht es als Recht und auch als Pflicht jeder und jedes Einzelnen, selbst zu denken und entsprechend zu handeln. Eine Schule in der Tradition der Aufklärung hat somit das oberste Ziel, die Grundlagen für eigenständiges Denken zu schaffen, zu eigenständigem Denken zu ermuntern und eigenständiges Denken einzufordern. Es ist ein ambitioniertes Ziel, aber letztlich muss es das einzige Ziel sein: Schüler\*innen auf dem Weg zu mündigen Bürger\*innen zu unterstützen. Dazu braucht es die Fähigkeit zum kritischen Denken und Hinterfragen, aber auch den Mut, an als Richtig erkanntem Festzuhalten und dafür einzustehen.

Kant hatte neben einer autoritären und blinden Gehorsam verlangenden Kirche mehrere Quellen der Unmündigkeit vor Augen:

Habe ich ein Buch, das für mich Verstand hat, einen Seelsorger, der für mich Gewissen hat, einen Arzt, der für mich die Diät beurteilt, u.s.w., so brauche ich mich ja nicht selbst zu bemühen. Ich habe nicht nötig zu denken, wenn ich nur bezahlen kann; andere werden das verdrießliche Geschäft schon für mich übernehmen.<sup>16</sup>

Die Kritik Kants scheint auch für die heutige Zeit aktuell zu sein. Im Zeitalter der Digitalisierung kommen Anbieter von Technologie als weitere Quelle

---

<sup>15</sup>Kant, *Was ist Aufklärung?*, S. 20.

<sup>16</sup>Kant, *Was ist Aufklärung?*, S. 20.

der Unmündigkeit hinzu: Wieso soll ich selber verstehen wie ein Computer funktioniert, wenn es Firmen gibt, welche mir sagen, wie und wozu ich diesen verwenden soll? Technologie zu verwenden, ohne diese zu verstehen und ohne diese hinterfragen zu können, gehört mit zu den Quellen der modernen Unmündigkeit.

Dabei ist es wichtig zu verstehen, an wen Kants Kritik und Aufforderung gerichtet ist. Die Ursache der kritisierten Unmündigkeit liegt nicht in der Unfähigkeit, sich des Verstandes zu bedienen, sondern am fehlenden Mut und der Faulheit:

Faulheit und Feigheit sind die Ursachen, warum ein so großer Teil der Menschen, nachdem sie die Natur längst von fremder Leitung frei gesprochen (naturaliter maiorenes), dennoch gerne zeitlebens unmündig bleiben; und warum es Anderen so leicht wird, sich zu deren Vormündern aufzuwerfen. Es ist so bequem, unmündig zu sein. <sup>17</sup>

Kant kritisiert also nicht jene, die mit Gewalt an der Ausübung ihrer Gedankenfreiheit gehindert werden oder die Fähigkeiten ermangeln, diese auszuüben. Kant richtet sich an jene, welche sich nicht zum selber denken und selbstverantwortlichen Handeln entscheiden, obwohl sie dazu in der Lage wären.

Bezogen auf die selbst gewählte Unmündigkeit durch die Verwendung von Technologie meint er deshalb genau uns - uns als Gesellschaft, als Schule, als Kollegium, als einzelne Lehrperson. Wir hätten durchaus die Möglichkeit, Technologie selbstverantwortlich einzusetzen, aber fehlender Mut und Faulheit hindern uns daran. Die wenigsten Lehrpersonen oder Schulleitungsmitglieder glauben, dass es gut ist, mit proprietärer Software zu arbeiten und sich als Schule in die Abhängigkeit von einem monopolistischen Softwaregiganten zu begeben. Dennoch tragen die meisten dieses System mit, indem sie die vermeintliche Alternativlosigkeit als Ausrede akzeptieren, sich mit einer schlechten Lösung zu arrangieren. Die gefühlte Ohnmacht und der Mangel an Handlungsoptionen ist selbstverschuldet. An uns richtet sich also Kant, wenn er sagt: Sapere Aude! Wage zu wissen.

Freie Software bietet einen Ausweg aus der Unmündigkeit gegenüber der Technologie. Statt sich von einer Firma vorschreiben zu lassen, wozu wir

---

<sup>17</sup>Kant, *Was ist Aufklärung?*, S. 20.

Technologie verwenden, ohne das wir diese verstehen, können wir freie Software verstehen, wir können sie mitgestalten und uns autonom für die Art und Weise ihres Einsatzes entscheiden.

Aufklärung ist nicht nur ein Privileg, sondern auch eine Verpflichtung. Wer selbstverschuldet unmündig ist, soll sich gegen diese Situation auflehnen. Der Einsatz freier Software ist ein konkreter Schritt zu mehr Mündigkeit. Deshalb sollte am Gymnasium mit freier Software gearbeitet werden.

# Kapitel 4

## Fazit

Im ersten Teil der Arbeit habe ich beschrieben, dass freie Software mehr ist als nur Software. Freie Software verkörpert Ideale und hat das Potential, zu einer gerechteren und nachhaltigeren Gesellschaft beizutragen. Im zweiten Teil der Arbeit habe ich gezeigt, dass die Ideale und Ziele, welche hinter der gymnasialen Bildung stehen, im Einklang mit den durch freie Software exemplifizierten Werte sind. So habe ich dafür argumentiert, dass wir am Gymnasium mit freier Software arbeiten sollten.

In der Einleitung habe ich beschrieben, wie fest wir an meiner Schule und in meinem Kanton in die Produktwelt eines proprietären Softwareanbieters verstrickt sind und welche Probleme dies meiner Ansicht nach mit sich bringt. Der Einsatz freier Software wäre frei von diesen Problemen. Es wird notwendig sein, weiter auf die Probleme proprietärer Software im Bildungsbereich aufmerksam zu machen und den Einsatz freier Software als Alternative zu propagieren. In dieser Arbeit habe ich aber gezeigt, dass der Wert der freien Software für das Gymnasium weit über diesen relativen Vorteil gegenüber proprietärer Software hinausgeht. Ich hoffe, dass ich mit dieser Argumentation im Kollegium, bei der Schulleitung und bei den zuständigen Stellen im Kanton Gehör finden werde, so dass wir irgendwann eine Schule sein werden, in welcher ausschliesslich und selbstverständlich freie Software eingesetzt wird.

# Literatur

- Benkler, Yochai. „Coase’s Penguin, or, Linux and the Nature of the Firm“. In: *Collaborative Ownership and the Digital Economy*. Hrsg. von Rishah Aiyer Ghosh. Cambridge: MIT Press, 2005, 169 –206.
- „Coase’s Penguin, or, Linux and the Nature of the Firm“. In: *Collaborative Ownership and the Digital Economy*. Hrsg. von Rishah Aiyer Ghosh. Cambridge: MIT Press, 2005, 169 –206.
- *The Wealth of Networks*. New Haven: Yale University Press, 2006.
- Bollier, David. „Protecting and Enlarging the Digital Republic“. In: *Amsterdam Law Forum* 14.1 (2009), 3 –10.
- „The Growth of the Commons Paradigm“. In: *Understanding Knowledge as Commons. From Theory to Practice*. Hrsg. von Charlotte Hess und Elinor Ostrom. Cambridge: MIT Press, 2007, 27 –40.
- Caffentzis, George. „A Tale of Two Conferences: Globalization, the Crisis of Neoliberalism and the Question of the Commons“. In: *Borderlands* (2012).
- Coleman, Gabriella E. *Coding Freedom. The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press, 2013.
- Dewey, John. *The Public and Its Problems*. Chicago: Swallow Press, 1927.
- Habermas, Jürgen. *Strukturwandel der Öffentlichkeit. Untersuchungen zu einer Kategorie der bürgerlichen Gesellschaft*. Berlin: Luchterhand, 1962.
- Heinz, Federico. „Freie Software - Vom Elfenbeinturm auf unseren Schreibtisch“. In: *Wem gehört die Welt? Zur Wiederentdeckung der Gemeingüter*. Hrsg. von Silke Helfrich und der Heinrich-Böll-Stiftung. München: oekom Verlag, 2009, 51 –55.
- Hess, Charlotte und Elinor Ostrom, Hrsg. *Understanding Knowledge as Commons*. Cambridge: MIT Press, 2007.
- Kant, Immanuel. *Was ist Aufklärung?* Hamburg: Meiner, 1999.



- Kelty, Christopher M. *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press, 2008.
- Lessig, Lawrence. *Code Version 2.0*. New York: Basic Books, 2006.
- *Free Culture*. New York: Penguin Press, 2004.
- Levy, Steven. *Hackers: Heroes of the Computer Revolution*. New York: Anchorpress/Doubleday, 1984.
- Love, James und Tim Hubbard. „Paying for Public Goods“. In: *Collaborative Ownership and the Digital Economy*. Hrsg. von Rishah Aiyer Ghosh. Cambridge: MIT Press, 2005, 207–229.
- Moglen, Eben. „Snowdon and the Future. Part IV: Freedom’s Future“. In: *www.snowdenandthefuture.info* (2013).
- Moody, Glen. *Die Software Rebellen. Die Erfolgsstory von Linus Torvalds und Linux*. Landsberg/Lech: verlag moderne industrie, 2001.
- Ostrom, Elinor. *Die Verfassung der Allmende. Jenseits von Staat und Markt*. Tübingen: Mohr Siebeck, 1999.
- Raymond, Eric S. *The Cathedral and the Bazaar*. Sebastopol: O’Reilly Media, 1999.
- Schoonmaker, Sara. *Free Software, The Internet, and Global Communities of Resistance*. New York: Routledge, 2018.
- Schweik, Charles M. „Free/Open-Source Software as a Framework for Establishing Commons in Science“. In: *Understanding Knowledge as Commons. From Theory to Practice*. Hrsg. von Charlotte Hess und Elinor Ostrom. Cambridge: MIT Press, 2007, 277–310.
- Stallman, Richard. „Free Community Science and the Free Development of Science“. In: *PLoS Medicine* 2.2 (2005), 169–170.
- Thalheim, Lisa. „Trusted Computing“. In: *Wem gehört die Welt? Zur Wiederentdeckung der Gemeingüter*. Hrsg. von Silke Helfrich und der Heinrich-Böll-Stiftung. München: oekom Verlag, 2009, 199–202.
- Torvalds, Linus und David Diamond. *Just for Fun: Wie ein Freak die Computerwelt revolutionierte*. München: Hanser, 2001.
- Vaidhyanathan, Siva. *Copyrights and Copywrongs: The Rise of Intellectual Property and How It Threatens Creativity*. New York: New York University Press, 2001.
- *Intellectual Property. A Very Short Introduction*. New York: Oxford University Press, 2017.
- *The Anarchist in the Library*. New York: Basic Books, 2004.
- Webb, Maureen. *Coding Democracy*. Cambridge: MIT Press, 2020.

- Zuboff, Shoshana. „Big Other: Surveillance Capitalism and the Prospects of an Information Civilization“. In: *Journal of Information Technology* 30 (2015), 75 –89.
- *The Age of Surveillance Capitalism*. New York: Public Affairs, 2019.