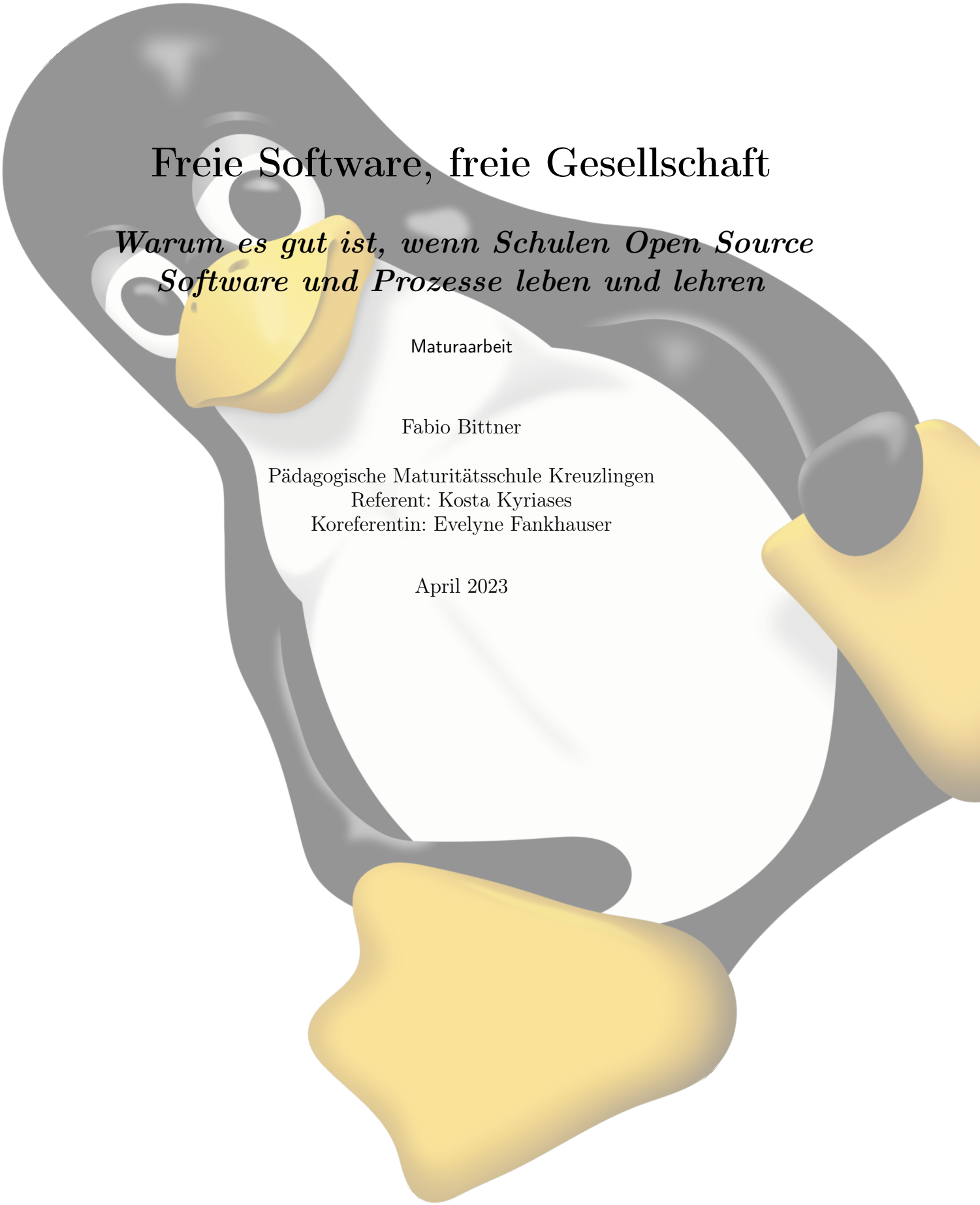

Freie Software, freie Gesellschaft

Warum es gut ist, wenn Schulen Open Source Software und Prozesse leben und lehren

Maturaarbeit

Fabio Bittner

Pädagogische Maturitätsschule Kreuzlingen
Referent: Kosta Kyriases
Koreferentin: Evelyne Fankhauser

April 2023

Zusammenfassung

Diese Arbeit soll Bildungsinstitutionen dazu ermutigen, *Freie Software* einzusetzen und die mit ihr verbundenen Prozesse aktiv zu leben. Sie möchte zu einem kollaborativen Modell der Zusammenarbeit anregen, in dem Schüler:innen gemeinsam mit Lehrpersonen und der Verwaltung den Schulbetrieb gestalten und erleben. Damit spielen Schulen eine wichtige Rolle im Wandel der Gesellschaft, der durch eine umfassende Bildung der Erwachsenen von Morgen geschieht.

Die Kontrolle der Informationstechnologie, die wir heute international tätigen Grossunternehmen überlassen haben, können unsere Schulen so zurück in die Hände unserer demokratischen Gesellschaft holen. Es handelt sich um kein Informatikthema, sondern um einen Weckruf zum Wahrnehmen unserer politischen Verantwortung.

Projektwebsite

Auf der begleitenden Projektwebsite für diese Arbeit finden Sie eventuelle Korrekturen, weiterführende Hinweise und Möglichkeiten zur Kontaktaufnahme und Diskussion.

`https://freie-software.bittner.it`

Titelblatt und Rückseite: Tux, von Larry Ewing und GIMP

Inhaltsverzeichnis

1 Einleitung **3**
- 1.1 Gefährliche Entwicklungen . . . 3
- 1.2 Zukunft gestalten . . . 3
- 1.3 Auftrag . . . 4

2 Vorgehensweise **5**
- 2.1 Materialien & Werkzeuge . . . 5

I Theorie **6**

3 Problemstellung **7**
- 3.1 Produktnamen statt Technologiebegriffe . . . 7
- 3.2 Wer kontrolliert unsere Daten . . . 7
- 3.3 Adaptive Learning . . . 7
- 3.4 Einfluss auf das Bildungswesen . . . 8

4 Was ist Freie Software **10**
- 4.1 Freie Software . . . 10
- 4.2 Open Source Bewegung . . . 11
- 4.3 Lizenzen . . . 12

5 Free/Libre Open Source Projekte **16**
- 5.1 Lösungen für den Schulbetrieb . . . 16
- 5.2 Wichtige FLOSS Beispiele . . . 21

II Anwendung **33**

6 Selbstversuche **34**
- 6.1 GNU/Linux am eigenen PC . . . 34
- 6.2 LaTeX zur Textverarbeitung . . . 35
- 6.3 Projektwebsite mit GitLab Pages . . . 36
- 6.4 Zur GNOME Dokumentation beitragen . . . 38
- 6.5 Korrektur im ILIAS Development README . . . 39
- 6.6 Umfragesoftware selbst entwickelt . . . 40

7 Software im Einsatz an Schulen **42**
- 7.1 Pädagogische Maturitätsschule Kreuzlingen . . . 42
- 7.2 Schulen in Grossbritannien . . . 43
- 7.3 Freie Software am Georg-Büchner-Gymnasium (Hannover) . . . 44
- 7.4 Free Upgrade for a digital Sustainable School (FUSS) . . . 45

8 Umfrage an Schweizer Schulen **47**
8.1 Kontaktaufnahme 47
8.2 Die Umfrage 48

III Analyse **59**

9 Wozu Bildung dient **60**
9.1 Definition 60
9.2 Geschichtliche Vorbilder 61

10 Massnahmen **64**
10.1 Schulhomepage selbst verwalten 64
10.2 Fokuswoche „Freie Software“ 65
10.3 Schüler:innen machen IT-Support 65
10.4 Lernstick als Schuldesktop 66
10.5 Einführung von LaTeX für grössere schriftliche Arbeiten 66

11 Schlusswort **68**

IV Appendix **69**

Literatur **70**

Abbildungsverzeichnis **74**

Tabellenverzeichnis **76**

Glossar **77**

Akronyme **80**

Kapitel 1

Einleitung

Die aktuelle Situation der Wissensvermittlung im Bereich Medien und Informatik ist unzufriedenstellend. Wir kennen nur einen Teil unserer Möglichkeiten, weil im Wesentlichen Produktschulung auf unseren Bildungsinstitutionen stattfindet. Dies vor allem in der allgemeinen Medienkompetenz, dem Arbeiten mit Dokumenten, der Informatik für die Allgemeinheit.

> Wir empfehlen Ihnen dringend, die verwendete Literatur mit Word zu erfassen, so dass das Literaturverzeichnis automatisch zusammengestellt werden kann (s. oben: MA Zitieranweisung mit Word 2013).

Abbildung 1.1: Auszug Wegleitung Maturaarbeit, PMS Kreuzlingen 2022/23

1.1 Gefährliche Entwicklungen

Wir unterschätzen die Probleme, die sich daraus ergeben. Abhängigkeit von Big Tech, ungewollte Überwachung statt Datenschutz, Verlust der Kostenkontrolle, Vernachlässigung der eigenen Wirtschaft, schlechter Einsatz von Steuergeldern, Gefährdung der Demokratie. All das ist selbstverschuldet durch unsere Bequemlichkeit, geködert durch scheinbar kostenlose Produkte.

Wir müssen aus diesem Problemsumpf heraus. Schulen müssen den Bildungsauftrag erfüllen und stärker Allgemeinwissen vermitteln. Die Kenntnis der Konzepte ist wichtiger als die der Produkte. Die Kenntnis aller Möglichkeiten schafft Freiheitsgrade und Unabhängigkeit, die wir für eine starke Wirtschaft und eine unabhängige Demokratie brauchen.

1.2 Zukunft gestalten

Auf der anderen Seite ist *Open Source* und *Freie Software* eine fantastische Chance, Methoden zu lernen und zu lehren, die unsere Zukunft verändern. Freie Software wird kollaborativ entwickelt, von Personen mit unterschiedlichen Kenntnissen und Interessen. Es ist kein Umfeld, das alleine den Nerds und Programmierern gehört.

Software gestaltet und bestimmt heute unsere Gesellschaft und wir haben gelernt, welche Erwartungen wir an sie haben dürfen. Wer nicht programmieren kann oder will, kann an der Benutzerfreundlichkeit mitarbeiten, die Dokumentation verbessern, Fehler melden, neue Features oder Verbesserungen vorschlagen, Koordinationsaufgaben, Kommunikation, Motivation und Fundraising übernehmen. Diese Prozesse, wenn wir sie beherrschen, müssen wir nützen, um die Software zu formen, die unsere Gegenwart beherrscht und unsere Zukunft gestaltet.

1.3 Auftrag

Diese Arbeit beschäftigt sich mit den genannten Problemen und zeigt Lösungsmöglichkeiten auf. Im Zentrum steht dabei das Erlernen der kollaborativen Prozesse von Freier Software (*Open Source*) im Schulbetrieb.[1]

> *We want our kids to be innovators and inventors, not technology tourists.*
> — *Charlie Reisinger, IT-Leiter des Penn Manor Schulbezirks, Pennsylvania*

Abbildung 1.2: Unsere Schüler:innen, unsere Mission

[1] Zitat aus Reisinger, *Enabling students in a digital age: Charlie Reisinger at TEDx Lancaster*, 6'40", und dem Kurzfilm Red Hat, *Penn Manor - Open Source Stories*

Kapitel 2

Vorgehensweise

Diese Arbeit entwickelt sich über drei Teile: Theorie, Anwendung und Analyse mit Lösungsvorschlägen.

Der erste Teil beschreibt die aktuelle Situation des Einsatzes von Informationstechnologie an unseren Schulen und in unserer Gesellschaft allgemein, und welche Probleme sich daraus ergeben. Das Thema *Freie Software* nimmt die Leser:innen mit auf eine Reise durch die Geschichte der IT der 80er- und 90er-Jahre. Es zeigt ein Bild der Computerbranche, das die meisten von uns nicht kennen. Nach dem Kennenlernen der unterschiedlichen Arten von Lizenzen werden freie Softwarelösungen für den Schulbetrieb vorgestellt und zum besseren Verständnis des Themas wichtige und bekannte Open Source-Projekte präsentiert und erklärt.

Im zweiten Teil sind zuerst Selbsterfahrungen beschrieben, die der Autor dieser Arbeit mit Freier Software und Open Source-Projekten gemacht hat. Danach wird der Softwareeinsatz an unterschiedlichen Schulen präsentiert und bei den Schulen, die *Freie Software* einsetzen, ihre Motivation und Details der praktischen Umsetzung erklärt. Abschliessend zeigt die Arbeit eine Umfrage, die im März 2023 an die Schulleiter von Schweizer Schulen ausgeschickt wurde, und die Auswertung der eingegangenen Antworten, die interessante Ergebnisse ans Licht bringen.

Freie Software erfordert einen unbequemen Geist. Ein eigenes Kapitel beschäftigt sich im dritten Teil daher mit den Kernaussagen geschichtlich relevanter Bildungspädagogen und versucht daraus einen Auftrag zur Nutzung von und der Beschäftigung mit Freier Software abzuleiten. Danach werden konkrete Massnahmen vorgeschlagen, die helfen sollen, *Freie Software* in der Schule und im Lehrplan zu verankern. Das Schlusswort gibt zusammenfassend einen Rückblick auf die Themen, die die Arbeit behandelt hat.

Ergänzend finden interessierte Leser:innen im Appendix (Teil 4) Literatur-, Abbildungs- und Tabellenverzeichnisse, sowie ein Glossar und eine Liste verwendeter Akronyme.

2.1 Materialien & Werkzeuge

Dieses Dokument wurde mit dem Programm LaTeX unter Verwendung des Texteditors *VSCodium* geschrieben (vgl. Abschnitt 6.2). Für Screenshots wurde das Screenshot-Tool von GNOME verwendet, zur Bearbeitung der Grafiken *GIMP*, zum Verkleinern der Dateigrösse das Online-Service *TinyPNG* (es verkleinert stärker als das freie Programm *Trimage*).

Für die Umfrage wurde als Notlösung *Google Forms* verwendet, wie in Kapitel 8 erläutert.

Gemeinsam mit dem Dokument wurde eine Projektwebsite aufgebaut. Dazu kam der Static Site Generator *Pelican* und *GitLab Pages* zum Einsatz (vgl. Abschnitt 6.3).

Teil I

Theorie

Kapitel 3

Problemstellung

Software nimmt in zunehmendem Mass Einfluss auf unser Leben. Gleichzeitig können wir – Schüler:innen, Lehrpersonen, Eltern und Entscheidungsträger in unserer Gesellschaft – immer weniger einschätzen, was die Software tut und welche Alternativen wir haben.

3.1 Produktnamen statt Technologiebegriffe

Wir beschreiben Technologien meist mit Produktnamen (z.B. „Word“, „Excel“, „PowerPoint“) und kennen in der Regel nur zwei Alternativen für Systeme (z.B. „Windows oder Apple“, „iOS oder Android“), beides meist Produkt- oder Markennamen von grossen IT-Konzernen. Müssten wir nicht eigentlich von „Textverarbeitung“, „Tabellenkalkulation“ und „Präsentationssoftware“ sprechen?

Woher kommt diese Produkthörigkeit? Ist es nur Gewohnheit? Oder kennen wir die Alternativen gar nicht, weil nicht darüber gesprochen wird?

3.2 Wer kontrolliert unsere Daten

Auch haben eine Reihe grosser Unternehmen (*Big Tech*) in den letzten Jahren begonnen, Einfluss auf beträchtliche Bereiche unseres Lebens zu nehmen. Die bekanntesten Namen davon sind: *Google* (*Alphabet*), *Amazon*, *Apple*, *Facebook* (*Meta*), *Microsoft*.

Inanspruchnahme von Dienstleistungen und sogar soziale Partizipation wird in vielen Bereichen bald nur mehr online möglich sein (z.B. e-Government, eHealth, Online Support bei diversen Unternehmen). Problematisch dabei ist, wie man immer wieder erfährt, dass für diese Dienstleistungen Technologie und Infrastruktur von *Big Tech* zur Verwendung kommt (z.B. *Google Analytics*, Amazon Cloud (*AWS*), Microsoft Cloud (*Azure*), Google Cloud (*GCP*)), weil Unternehmen zur Optimierung in die Cloud drängen. Man kann sich der Kontrolle von *Big Tech* deshalb kaum mehr entziehen.

3.3 Adaptive Learning

Stefano Barale, Informatik-Lehrer an einem Gymnasium mit Schwerpunkt Informatik in Turin, erklärt in einem Vortrag Ende März 2023 die Gefahren, die vom Konzept des *Adaptive Learning* ausgehen. Er nimmt dabei Bezug insbesondere auf die beiden *Big Tech*-Grössen *Microsoft* und

Google, die mit ihren Produkten *Microsoft 365 for Education* und *Google Workspace for Education* Vorreiter in dem Gebiet sind.[1]

Die Idee des *Adaptive Learning*, die den Frontalunterricht um das individuelle Lernen ergänzt, ist nicht neu und grundsätzlich gut. *Big Tech* zeigt, was mit der heutigen Technologie, Big Data und KI, schon möglich ist. Einen Eindruck darüber liefert die Dokumentation von *Microsoft*s *Education Insights*[2] und der *Microsoft Education Blog*.[3] Zu *Google Workspace for Education* finden sich weniger Details zu den Daten, die gesammelt werden, lediglich Beteuerungen, dass Datenschutzregeln eingehalten werden und keine Werbung ausgespielt wird.[4] Allerdings war *Google* mit seinem Produkt schon vor einem US-Gerichtshof und musste sich Ende 2021 verpflichten, Anpassungen an *Workspace for Education* umzusetzen und Administrationsmöglichkeiten für die Sammlung von Daten im Produkt einzubauen.[5]

Das Problem dabei ist, so Barale, dass die *Microsoft*- und *Google*-Tools Daten im grossen Stil sammeln. Details dazu gibt *Microsoft* auf seinen Supportseiten bekannt.[6] Barale meint, *Microsoft* hätte in diesem Thema im Vergleich zu *Google* aktuell die Nase vorn und bezieht sich dabei vermutlich darauf, dass das Unternehmen es geschafft hat, *Microsoft Teams* in der Zeit der COVID-Pandemie in vielen Institutionen als täglich benutztes Werkzeug zu etablieren.

Die angebotenen Tools bieten interessante und nützliche Funktionen. Es ist aber problematisch, dass man bei einem so wichtigen Aspekt wie der Bildung die Kontrolle grossen, global agierenden Unternehmen überlässt. Die Datenkontrolle sollte in der Kompetenz der Schulen oder zumindest der Nationalstaaten bleiben. Das Betreiben einer Software sollte also im öffentlichen Rahmen stattfinden. *Freie Software* wäre insofern ein passender Kandidat für eine diesbezügliche Lösung.

3.4 Einfluss auf das Bildungswesen

Das Thema *Digitalisierung* ist in den letzten Jahren auch in der Schule zum Schlagwort geworden. In der Schweiz ist es im *Lehrplan 21* mit dem Modul *Medien und Informatik* präsent.[7] Das ist auch den Digitalkonzernen nicht verborgen geblieben und der Einfluss dieser auf das Bildungswesen macht sich immer stärker bemerkbar.

Die Situation sieht derzeit so aus, dass Kinder und Jugendliche dem unmündigen Konsum von „Fertiggerichten“ der *Big Tech*-Konzerne *Apple* und *Microsoft* ausgesetzt sind. Bereits in der Primarschule wird ihnen ein Tablet der Marke *Apple* ausgehändigt und sie lernen im Informatik-Unterricht, das Office-Paket der Firma *Microsoft* zu bedienen.

Grosskonzerne statten Schulen mit ihren Produkten aus und lassen sich Schüler:innen bereits im jüngsten Alter an ihre Produkte heranzuführen. Produkte, die in der Schule im Einsatz sind, werden konsequent auch zuhause eingesetzt. Neukunden werden direkt in jungen Jahren erworben und an die Grosskonzerne gebunden.

Sie lernen somit nicht die Funktionsweise und die Grundsätze der Software zu studieren bzw. zu ergründen, was hinter der Software steckt. Sie erfahren nicht, dass Software modelliert werden kann und an eigene Wünsche und Bedürfnisse angepasst werden kann. Sie üben nicht

[1] Vgl. Barale, *Che volto avrà la scuola del futuro? Buone leggi e buone pratiche (Stefano Barale, docente, membro CUB SUR)*.
[2] Vgl. Microsoft Education Team, *Unlock students' full potential with Education Insights in Microsoft Teams*.
[3] Vgl. Microsoft Education Team, *How data and AI are changing the world of education*.
[4] Vgl. Google, *Produkte für Lehre und Lernen*.
[5] Vgl. Clark, *Google's children's data privacy settlement explained*.
[6] Vgl. Microsoft Education Team, *Leitfaden zum Education Insights Premium*.
[7] Vgl. Departement für Erziehung und Kultur Kanton Thurgau, *Lehrplan Volksschule Thurgau*.

das analytische Denken und die Problemlösung, sondern vielmehr lernen sie sich regungslos an die Gegebenheiten anzupassen und ihre Ansprüche auszuschalten.

> *Die Situation* [in der Schule] *ist vergleichbar mit der eines Bauern, der zukünftigen Bauern beibringt, wie man Produkte im Supermarkt einkauft und weiterverkauft, statt ihnen das Säen und Ernten beizubringen.*[8]

[8] Vgl. Wawrzyniak, *Der DigitalPakt Schule | Digitalcourage*.

Kapitel 4

Was ist Freie Software

Freie Software und *Open Source* sind wichtige Begriffe unserer Zeit. *Proprietäre Software* ist ihr Gegenspieler. Was als soziale Bewegung begann, ist mittlerweile zu einem bedeutenden Wirtschaftszweig geworden. Selbst die klassischen Beispiele für Produzenten proprietärer Software, beispielsweise *Microsoft*, investieren in *Open Source*-Software und machen massiv Umsatz mit ihr.[1]

4.1 Freie Software

Der Begriff *Freie Software* steht für Software mit einer Lizenz, die dem Empfänger bestimmte Freiheiten gewährt. Richard Stallman, der Gründer der *Free Software Foundation* hat diese vier Freiheiten verfasst.

Vier wesentliche Freiheiten[2]

1. Die Freiheit, das Programm auszuführen wie man möchte, für jeden Zweck.
2. Die Freiheit, die Funktionsweise des Programms zu untersuchen und eigenen Datenverarbeitungbedürfnissen anzupassen.[a]
3. Die Freiheit, das Programm weiterzugeben und damit Mitmenschen zu helfen.
4. Die Freiheit, das Programm zu verbessern und diese Verbesserungen der Öffentlichkeit freizugeben, damit die gesamte Gesellschaft davon profitiert.[a]

[a] Der Zugang zum Quellcode ist dafür Voraussetzung.

Freie Software hat nicht primär damit zu tun, dass sie gratis wäre. Stallman stellt dies klar, indem er explizit sagt, dass man für das Verteilen von Freier Software gerne eine Gebühr verlangen darf. Das würde sogar helfen, die Entwicklung dieser Software nachhaltig zu machen.[3]

[1] Vgl. Pietschmann, *Linux is Most Used OS in Microsoft Azure – over 50 percent of VM cores | Build5Nines*.
[2] Vgl. Stallman und Lessing, *Free Software, Free Society*, S. 41.
[3] Vgl. ebd., S. 18.

4.1.1 Die Sternstunde der Free Software Foundation

Alles begann mit einem Softwareentwickler namens Richard Stallman, einem Papierstau bei einem Drucker und der Unmöglichkeit, Verbesserungen bzw. den Einbau von weiteren Features am Quellcode anzubringen.

Im 20. Jahrhundert schrieben, gestalteten und benutzen Softwareentwickler ihre eigenen Betriebssysteme und stellten den Quellcode anderen zur Verfügung:

> *Sharing of software* [...] *is as old as computers, just as sharing of recipes is as old as cooking.*[4]

In den 80er-Jahren kamen moderne Computer mit einem eigenen Betriebssystem auf den Markt – der Quellcode war aber nicht mehr sichtbar und frei. Es konnten keine Änderungen und Anpassungen durchgeführt werden und beim Teilen mit anderen wurde man zum Piraten. So entstand das *Proprietary-Software Social System*.

Es wurde zur Norm, dass es verboten war, Software zu ändern oder zu teilen. Eine Gemeinschaft, in der zum Lösen von Softwareproblemen zusammengearbeitet wurde, wurde zur Piratengemeinschaft. Das Soziale wurde illegal. Die Idee der proprietären Software war: Bitte uns, die Anpassungen für dich zu machen.[5]

Richard Stallman und sein Drucker waren in diesem neuen System gefangen. Unser Softwareentwickler wollte sein Druckerproblem lösen und bat deshalb den Lieferanten um den Quellcode für das Drucker-Kontrollprogramm. Prompt erhielt er eine Absage mit dem Verweis auf das – von ihm unterzeichnete – Non-Disclosure Agreement (NDA).

Dieser Zwischenfall verärgerte Richard Stallman sehr. Er überlegte, wie er die Welt aus diesem Dilemma holen konnte. Er musste ein *Betriebssystem* entwickeln! Ohne ein solches konnte man einen Computer nicht verwenden. Es musste ein *Betriebssystem* sein, das Benutzer auf ihre Bedürfnisse anpassen können und mit ihren Nachbarn teilen durften. Damit konnte man wieder eine Gemeinschaft von Entwicklern bilden und jede:n dazu einladen – man würde wieder guten Gewissens Computer nutzen und gemeinsam Probleme lösen dürfen.[6]

> *Copyleft—all rights reversed.*

Abbildung 4.1: Das Gegenteil von *Copyright – all rights reserved*

1984 begann Stallman an GNU zu arbeiten. Ein Jahr später erfand er das *Copyleft*.[7] 1991 steuerte Linus Torvalds das Herzstück für das *Betriebssystem* bei, einen Unix-kompatiblen Kernel, der nach Linus *Linux* genannt wurde. Die Kombination dieser beiden wichtigen Teile ergab ein vollständiges, freies *Betriebssystem*. Das GNU-System mit dem *Linux*-Kernel wird seither *GNU/Linux* genannt.[8]

4.2 Open Source Bewegung

1998 entstand neben der Bezeichnung *Freie Software* auch die von *Open Source*. Ein Teil der Community wollte das Missverständnis von „free = gratis“ aus der Welt schaffen. Andererseits

[4] Stallman und Lessing, *Free Software, Free Society*, S. 15.
[5] Vgl. ebd., S. 16.
[6] Vgl. ebd., S. 17.
[7] Ebd., S. 21.
[8] Vgl. ebd., S. 26.

gab es Leute wie Eric S. Raymond, die mehr die praktischen Aspekte der Software-Freiheit in den Vordergrund stellen wollten als den gesellschaftspolitischen. Sein Buch *The Cathedral & The Bazaar* liefert dazu den notwendigen Anstoss.[9]

Bruce Perens, der gemeinsam mit Raymond 1998 die Open Source Initiative (OSI) gründete, erklärt im Film *Revolution OS* das Hauptanliegen der Anhänger der Bewegung: Programmierer möchten beim Verbessern von Software zusammenarbeiten, ohne sich um *Geistiges Eigentum*, Verträge und Anwälte kümmern zu müssen. Dafür verzichten sie gerne auf einige ihrere Rechte in diesem Zusammenhang und lassen einfach die ganze Welt ihre Software nutzen.[10]

Open Source Definition Analog zu den *Vier Freiheiten* der FSF hat die Open Source Initiative die *Open Source Definition*. Sie ist etwas umfangreicher als erstere.[11]

1. Freie Weitergabe (auch Verkauf)
2. Quellcode (zusätzlich zum Binärcode)
3. Abgeleitete Werke (müssen erlaubt sein)
4. Integrität des Quellcodes des Autors (die Lizenz kann verlangen, dass geänderter Code nicht mit dem Namen des ursprünglichen Autors gekennzeichnet wird)
5. Keine Benachteiligung von Personen oder Gruppen
6. Keine Benachteiligung von Arbeitsgebieten (darf überall genutzt werden)
7. Verteilung der Lizenz (Gültigkeit für alle Empfänger)
8. Die Lizenz darf nicht auf ein Produkt beschränkt sein (Unabhängigkeit vom ausgelieferten Zustand)
9. Die Lizenz darf andere Software nicht einschränken
10. Die Lizenz muss technologieneutral sein (Allgemeingültigkeit)

Die Bezeichnungen *Freie Software* und *Open Source* beschreiben grundsätzlich die gleiche Software-Kategorie und unterscheiden sich lediglich in der Philosophie, in den Werten und nicht zuletzt in den verschiedenen Kriterien, für die Lizenzen akzeptiert werden. Die GNU-Community verwendet weiter die Bezeichnung *Freie Software*, um die Tatsache zu unterstreichen, dass nicht nur der technologische Nutzen wichtig ist, sondern vor allem die Freiheit für die Gesellschaft.[12]

Eine ausschöpfende Liste von OSI-kompatiblen Lizenzen findet sich auf der Homepage der Open Source Initiative.[13]

4.3 Lizenzen

Eine Software-Lizenz ist ein rechtliches Instrument, das die Nutzung und den Vertrieb von Software, sowie auch deren Modalitäten und Grenzen definiert, die ein Eigentümer einem Nutzer

[9] Vgl. Raymond, *The Cathedral and the Bazaar*.
[10] Vgl. Moore, *Revolution OS*, 3'12".
[11] Vgl. Open Source Initiative, *The Open Source Definition – Open Source Initiative*.
[12] Vgl. Stallman und Lessing, *Free Software, Free Society*, S. 30.
[13] Vgl. Open Source Initiative, *Licenses – Open Source Initiative*.

einräumt. Dabei ist zu beachten, dass der Nutzer kein Eigentumsrecht an der Software erwirbt, sondern nur das Recht, sie zu nutzen.

Diese Rechte können beinhalten:[14]

- Allgemeines Nutzungsrecht an der Software; Zahl der Nutzer, Art der Nutzung usw.
- Recht auf Weiterverbreitung der Software
- Recht auf Veränderung der Binärdateien
- Recht auf Veränderung des Quellcodes, sofern vorhanden
- Recht auf Weiterverbreitung veränderter Versionen der Binärdateien oder des Quellcodes
- Recht auf Verbindung (*Linking*) der Binärdateien oder des Quellcodes mit anderer Software; statisches Linking vs. dynamisches Linking

Es gibt eine grosse Anzahl an Arten von Software-Lizenzen. Der Einfachheit halber kann man Software-Lizenzen in zwei grosse Gruppen unterteilen: *Proprietäre Software* und Freie & *Open Source* Software.

4.3.1 Proprietäre Software-Lizenzen

Proprietär bedeutet „im Eigentum befindlich“ bzw., im juristischen Sinn, „urheberrechtlich geschützt“.

Für *proprietäre Software* bleibt der Code in der Regel geheim und wird nicht mit dem Programm mitgeliefert. Somit kann der Benutzer den Code nicht inspizieren, also weder kontrollieren, was der Code macht, noch von der Implementierung lernen, und auch nicht den Code ändern und verbessern.

Zusätzlich ist es allgemein bekannt, dass *proprietäre Software* die Kontrolle zu ihren Vorteilen nutzt und diese den Nutzern als „Security“ verkauft.[15] Wir haben keine Kontrolle über die Updates, die automatisch durchgeführt werden.

Freeware-Lizenzen Freeware-Lizenzen zeichnen sich dadurch aus, dass die Software kostenlos weitergegeben wird, wobei jedoch nur der Zugriff auf den ausführbaren Code, nicht aber auf den Quellcode gewährt wird. Der Verzicht des Autors beschränkt sich also auf die Nutzungsvergütung, nicht aber auf das Urheberrecht.

Shareware-Lizenzen Software mit Shareware-Lizenz ist nur anfangs und für einen begrenzten Zeitraum kostenlos, danach ist für die weitere Nutzung eine Gebühr zu entrichten, andernfalls verweigert die Software ihren Dienst. Das Programm wird dazu mit einem geeigneten Algorithmus vorbereitet, der ohne gültige(n) Lizenz(schlüssel) den Betrieb des Programms beendet oder verweigert. Dass dazu der Quellcode nicht offengelegt wird, erklärt sich von selbst.

4.3.2 Freie und Open Source Software (FOSS) Lizenzen

FOSS- oder FLOSS-Lizenzen sind Lizenzen, die die Nutzung, Weiterverbreitung und Änderung urheberrechtlich geschützter Werke erlauben.[16]

[14] Vgl. Kambach, *Open-Source-Lizenzen Vergleich & Übersicht*.
[15] In Entwicklerkreisen sagt man zu dieser Praxis scherzhaft, „Security by obscurity“.
[16] Vgl. Wikipedia, *Freie Lizenz*.

Creative Commons-Lizenzen Die Creative Commons-Lizenzen wurden 2001 eingeführt, um den Urheber eines Werkes zu schützen, aber gleichzeitig dem Nutzer Rechte zu gewähren. CC-Lizenzen basieren auf vier Klauseln, die je nach Bedarf kombiniert werden können, woraus sechs verschiedene Arten von Lizenzen entstehen können (Abbildung 4.2).[17]

- Namensnennung (Attribution, *BY*)
- keine Überarbeitung erlaubt (Share-alike, *SA*)
- nur nicht-kommerzielle Nutzung (Non-commercial, *NC*)
- Weitergabe unter gleichen Bedingungen (No derivative works, *ND*)

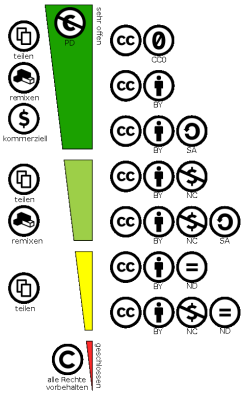

Abbildung 4.2: Kombinationsmöglichkeiten für Creative Commons-Lizenzen

Open Source-Lizenzen *Open Source*-Lizenzen werden ohne Zahlung durch den Nutzer gewährt, und der Nutzer erhält auch den Quellcode, sodass das Programm vollständig zur Verfügung steht; diese Lizenzen sind nicht exklusiv, unterliegen keinen zeitlichen und territorialen Beschränkungen, und jede von *Open Source*-Programmen abgeleitete Software muss unter den gleichen Bedingungen, also der gleichen Lizenz, zur Verfügung gestellt werden.

Public Domain Diese Lizenz macht aus der Software eine „vogelfreie“ Software. Sie erlaubt den Benutzern, das Programm inklusive ihrer Änderungen mit anderen zu teilen. Sie erlaubt den Benutzern aber auch das Programm in *proprietäre Software* zu verwandeln. Man kann im Programm Änderungen und Verbesserungen anbringen und es als proprietäres Produkt verwerten. Die Personen, die dieses modifizierte Programm erhalten, haben nicht die Freiheit, die der ursprüngliche Autor seinen Nutzern gab.[18]

Copyleft

Eine *Copyleft*-Lizenz garantiert jedem Benutzer die vier Freiheiten von Freier Software und stellt sicher, dass auch jede geänderte und erweiterte Software wiederum die *Copyleft*-Lizenz trägt.[19]

[17] Vgl. Creative Commons, *Share your work - Creative Commons*.
[18] Vgl. Stallman und Lessing, *Free Software, Free Society*, S. 89.
[19] Vgl. ebd., S. 89.

GNU General Public Licence (GPL) 1989 entwickelte das GNU-Projekt für seine Software eine einheitliche Lizenz, die folgende Kriterien erfüllen sollte:[20]

- Freiheit das Programm für jeden Zweck auszuführen
- Freiheit den Quellcode zu studieren und anzupassen
- Freiheit, das Programm zu kopieren
- Freiheit, das veränderte Programm zu kopieren

Diese Freiheiten sollten auch langfristig sichergestellt werden. Zu diesem Zweck enthält die GPL zwei wesentliche Klauseln:

- Jedes Derivat muss ebenfalls vollständig unter der GPL lizenziert werden.
- Bei Weitergabe des Programmes in Binärform muss der Quellcode des gesamten Programms mitgeliefert oder auf Anfrage ausgehändigt werden.

LGPL (Light GPL) Diese Lizenz entstand als *Library General Public License*, sie wurde jedoch in *Lesser* umbenannt. Gedacht ist diese Lizenz für *Freie Software*, die in kommerzielle Projekte eingebracht werden soll.[21]

Mozilla Public Lizenz 1998 wurde der Quellcode des Web-Browsers Mozilla freigegeben und erhielt eine neue Lizenz: Die *MPL*. Diese Lizenz ähnelt aber grundsätzlich der LGPL hat aber einige neuartige Klauseln.

Permissive Licenses

Open Source-Lizenzen, die weniger Auflagen haben als *Copyleft*-basierte Lizenzen, nennt man *freizügige Lizenzen*. Ableitungen von solcher Software muss nicht unter derselben Lizenz veröffentlicht werden wie Quellcode des ursprünglichen Werkes. Man darf diese Werke sogar zu proprietärer Software machen.

Beispiele für freizügige Lizenzen sind *Apache 2.0*, *BSD*, *BSD-2*, *BSD-3*, *MIT*

Die *MIT* gilt als die populärste *permissive license*. Gemäss eigener Recherche verwenden viele Projekte auf GitHub diese Lizenz.

[20] Vgl. Kambach, *Open-Source-Lizenzen Vergleich & Übersicht*.
[21] Vgl. ebd.

Kapitel 5

Free/Libre Open Source Projekte

Es gibt eine Unzahl von Freier Software, teilweise von erstaunlicher Qualität. Im Gegensatz zu proprietärer Software, bei der es im Normalfall ein Produkt pro Funktionsgebiet gibt, sind für eine Funktion meistens eine Menge FLOSS-Projekte vorhanden, aus denen man aussuchen kann.

5.1 Lösungen für den Schulbetrieb

Für den Schulbetrieb gibt es schon eine Vielzahl von Anstrengungen, einen *Software-Stack* zusammenzustellen. Ein schon 20 Jahre laufendes Grossprojekt in Südtirol ist FUSS. Die im FUSS-Projekt enthaltenen Applikationen sind in seiner Projektdokumentation aufgelistet.[1]

Die Lokalgruppe Zürich der Free Software Foundation Europe (FSFE) hat auf der Website „Lernen wie die Profis“ eine Aufstellung von Software zusammengetragen, die als Blaupause für die Schule dienen kann.[2] Die Tabellen in diesem Abschnitt zeigen eine überarbeitete Version davon.

Tabelle 5.1 führt Basis-Software für den Betrieb einer Infrastruktur an, z.B. *Betriebssystem-Distributionen* und Druckermanagement. Hilfreich zum Verständnis der Zusammenhänge zwischen den Linux-*Distributionen* ist hierbei übrigens die grafische Darstellung der geschichtlichen Entwicklung derselben.[3] Die Tabelle 5.2 erwähnt Schulinformations- und Verwaltungssysteme, die zur Administration von Schulen Verwendung finden. In Tabelle 5.3 sind Kommunikationsinstrumente angeführt, wie sie spätestens seit der COVID-Pandemie jede Schüler:in kennt. Lernmanagementsysteme (Tabelle 5.4) sind Plattformen zur Unterrichtsgestaltung. Diese werden, wie die Umfrage in Kapitel 8 bestätigt, schon heute an vielen Schulen der Schweiz eingesetzt. Tabelle 5.5 listet Office-Programme auf, die veritable Alternativen zu *Microsoft Office 365* darstellen. Schliesslich sind in den Tabellen 5.6 und 5.7 die wichtigsten Vertreter von freien Programmen für Grafik, Desktop Publishing und Multimedia aufgelistet.

Eine gute Aufstellung von Argumenten für den Einsatz von Freier Software an Schulen findet sich übrigens in einem Blogpost von Digitalcourage aus dem Jahr 2020.[4]

[1] Vgl. FUSS, *FUSS - Documentazione*.
[2] Vgl. FSFE, *Lernen wie die Profis - Lösungen*.
[3] Vgl. Wikipedia, *List of Linux distributions*.
[4] Vgl. Wawrzyniak, *Freie Software für Schulen | Digitalcourage*.

Tabelle 5.1: Infrastruktur

	Arch Linux Rolling Release GNU/Linux-Distribution
	Debian DPKG-basierte GNU/Linux-Distribution
	DebianEdu (Skolelinux) GNU/Linux-Distribution Debian angepasst an die Bedürfnisse von Schulen
	FUSS-Projekt Komplettlösung für 360-Grad-Management des Schulnetzwerks
	Manjaro Linux Arch Linux-basierte Distribution
	Nextcloud Cloudspeicher und Kollaborationssuite (Files, Talk, Groupware, Office).
	Red Hat RPM-basierte GNU/Linux-Distribution
	SavaPage Druckportal Software
	Ubuntu Debian-basierte Distribution

Tabelle 5.2: Schulinformations- und Verwaltungssysteme

	AlekSIS Schulinformations- und Organisationssystem. Erlaubt das Verwalten von Personendaten, Stundenplänen, Klassenbuch, Sitzplänen, Vertretungspläne, Speisepläne für Mensa, etc.
	odoo Unternehmensverwaltungssoftware (ein ERP-System). Verwaltet Personendaten, Finanzdaten (Ein- und Ausgänge), Inventar, Einkäufe, Zeiterfassung, Projektmanagement, etc.

Tabelle 5.3: Kommunikation

	BigBlueButton Softwarelösung für das virtuelle Klassenzimmer
	Jitsi Videokonferenz Software
	Nextcloud Talk Chat, Audio, Video, kollaboratives Arbeiten an Aufgaben. Ein Ersatz für *Microsoft Teams*.
	Matrix Chat- und Kollaborationsserver
	Mattermost Kommunikations- und Kollaborationsplattform (Chat, Audio, Video, zahlreiche Integrationen von externen Plattformen)
	Rocket Chat Kommunikationsplattform (Chat, Audio, Video, zahlreiche Integrationen von externen Plattformen)

Tabelle 5.4: Lernmanagementsysteme

	ILIAS Webbasiertes Learning Management System, das an Schweizer Schulen und Unternehmen populär ist. Verwaltet Personendaten, Ausbildungsstand, Ausbildungspläne, Aufgaben und Abgaben.
	Mahara Portfolio Managementsystem, mit dem man Unterrichtsmaterialien verwalten, teilen und diskutieren, sowie miteinander über die Plattform kommunizieren kann. Leichtgewichtig in der Bedienung. Kann von Lehrpersonen für die Unterrichtsgestaltung und von Schüler:innen zur Erledigung von Kursarbeiten verwendet werden.
	Moodle Webbasiertes Learning Management System, ähnlich ILIAS, das an Schweizer Schulen populär ist. Die Benutzeroberfläche ist simpler als die von ILIAS, aber man bekommt dafür das Gefühl von Übersicht und Ruhe.

Tabelle 5.5: Office

AbiWord
Plattformunabhängiges Textverarbeitungsprogramm. Sehr schnell und schlank. Unterstützt alle gängigen Textdokumentformate, inklusive *Microsoft Word* und *LibreOffice Writer*.

Gnumeric
Microsoft Excel-Alternative für *Windows*, *Linux* und *macOS*. Kann *Microsoft Excel*- und *LibreOffice Calc*-Dokumentformate lesen und schreiben.

LibreOffice
Ersatz für *Microsoft Office 365* (Word, Excel, PowerPoint, usw.). Seit 2020 auch als Cloudversion verfügbar (*LibreOffice Online*).

Open Education Server
Komplette Ersatzlösung für Office 365 auf Open Source-Basis, bereitgestellt von CH Open. Enthält neben *LibreOffice Online* auch den Cloudspeicher *Nextcloud* und das Kollaborationstool *Nextcloud Talk*.

Thunderbird
Microsoft Outlook-Ersatz (E-Mail, Kalender, Aufgaben, etc.). Nur eine der vielen freien Alternativen für E-Mail & Co.!

Xournal++
Notizblock-Applikation. Alternative zu *Apples Notes* und *Microsoft OneNote*. Erlaubt neben Textnotizen und Grafiken auch handschriftliche Notizen, z.B. mit einem Stift auf Geräten mit Touchscreen.

Tabelle 5.6: Grafik und Desktop Publishing

Drawpile
Grafikprogramm, das kollaboratives Arbeiten ermöglicht.

GIMP
Ein mächtiges Bildverarbeitungsprogramm für Fotos und ähnliche Pixel-basierte Grafiken. Vergleichbar mit *Adobe Photoshop*.

Inkscape
Vektorbasiertes Grafikprogramm, z.B. zum Erstellen und Bearbeiten von `.svg` Dateien.

Scribus
Desktop Publishing-Programm zum Erstellen von Flyern, Magazinen und dgl. Vergleichbar mit *Adobe InDesign* oder *Apples Pages*.

Tabelle 5.7: Multimedia

Ampache
Audio/Video-Streaming Server mit Weboberfläche. Von der Nutzung her vergleichbar mit Amazons *Audible*. Erlaubt das Verwalten von Audio- und Videoinhalten, z.B. für Audiobooks oder Lernvideos an Schulen.

Audacity
Plattformunabhängiger Audioeditor und -rekorder.

Openshot
Mächtige Videobearbeitungs-Applikation für *Windows*, *macOS* und *Linux*.

PeerTube
Freie Video-Hosting- und Streamingplattform, vergleichbar mit *YouTube*. Ein PeerTube-Server, den man selbst installiert, kommuniziert mit anderen PeerTube-Servern. Man speichert nur seine eigenen Inhalte, abrufbar sind sie aber alle auf allen PeerTube-Servern.

5.2 Wichtige FLOSS Beispiele

Um in *Open Source*-Projekten mitarbeiten zu können, muss man verstehen, erstens welches Problem die einzelnen Projekte zu adressieren versuchen und zweitens wie die Zusammenarbeit in jedem Projekt funktioniert.

In diesem Abschnitt wird eine Auswahl der wichtigsten Projekte aufgelistet und beschrieben.

5.2.1 Linux Kernel

Der Linux Kernel ist das Herzstück des GNU/Linux *Betriebssystem*. Er kümmert sich um die Interaktion zwischen der Hardware und den Prozessen, das sind die Programme, die sich in Ausführung befinden. Dabei verwaltet es Ressourcen (Arbeitszeit der CPU, Speicherplatz auf angeschlossenen Festplatten, verfügbarer Arbeitsspeicher, Kapazität für Netzwerkkommunikation, Drucker Warteschlangen, etc.) so effizient wie möglich.[5][6]

Wie Linux geboren wurde

Linus Torvalds startete die Arbeit am Linux-Kernel 1991 während seines Studiums an der Universität Helsinki, inspiriert durch das Buch *Operating Systems: Design and Implementation* von Andrew S. Tanenbaum, Universitätsprofessor in Amsterdam, die Konzepte von Unix und seine Frustation mit *Minix*, dem *Betriebssystem*, mit dem Prof. Tanenbaum lehrte.[7]

Anfang 1991 kaufte sich Torvalds einen 386er-PC mit dem *Betriebssystem DOS* und bestellte *Minix*, das er auf diesem PC anschliessend installierte. Frustriert vom schlechten Terminalprogramm auf *Minix*, das er brauchte um sich über das Internet zur Universität zu verbinden, begann er, sein eigenes Terminalprogramm zu entwickeln.[8]

Im Frühling begann Linus Funktionalität zu ergänzen, die Dateien in das *Minix*-Dateisystem schreiben konnte. Damit begann sich sein Terminal Emulator zu einem *Betriebssystem* zu entwickeln.[9]

Im Sommer begann Linus, eine *Shell* – die Bourne-Again *Shell*, *Bash* – als Teil seines Betriebssystems zu booten. Die Software war frei in Internet verfügbar. Damit konnte er schnell Fortschritte machen und schon im Herbst, am 25. August 1991, konnte er seine heute berühmte Ankündigung machen. Der Linux-Kernel war geboren.[10]

[5] Vgl. Red Hat, *What is the Linux kernel?*
[6] Vgl. LinuxForDevices, *The Linux Kernel – Explained in an Easy to Understand Way.*
[7] Vgl. Torvalds und Diamond, *Just for Fun*, S. 51 f.
[8] Vgl. ebd., S. 60 f.
[9] Vgl. ebd., S. 77 f.
[10] Vgl. ebd., S. 82 f.

> *Hallo alle da draussen, die Minix einsetzen -*
>
> *Ich arbeite an einem (freien) Betriebssystem (nur ein Hobby, wird nicht gross und professionell sein wie GNU) für 386(486) AT-kompatible Rechner. Das Projekt entwickelt sich seit April und beginnt fertig zu werden. Ich hätte gern Rückmeldungen über Eigenschaften, die die Leute an Minix mögen/nicht mögen, da mein Betriebssystem diesem in einigen Merkmalen ähnelt (gleiches physikalisches Layout des Dateisystems (aus Praktikabilitätsgründen) und einige andere Dinge).*
>
> *Ich habe im Moment bereits die bash (1.08) und gcc (1.40) portiert, und es scheint zu funktionieren. Das bedeutet, dass ich in einigen Monaten etwas haben werde, womit man arbeiten kann, und mich interessiert, welche Eigenschaften die meisten Leute gerne sehen würden. Alle Vorschläge sind willkommen, aber ich kann nicht versprechen, dass ich sie auch einbaue :-)*
>
> *Linus (torvalds@kruuna.helsinki.fi)*
>
> *PS. Ja, es enthält keinerlei Minix-Code und das Dateisystem ist multi-threaded. Es ist NICHT portierbar (es benutzt das 386er-Taskswitching etc.) und wird vermutlich nie etwas anderes unterstützen als AT-Festplatten, da ich nur solche besitze :-(.*

Abbildung 5.1: Linus Torvalds in comp.os.minix (Usenet), 25. August 1991

Entwicklung & Kollaboration

Die Entwicklung des Linux Kernels findet in Git-Repositories auf `git.kernel.org` und unter Einsatz von Mailing-Listen statt. Auf *GitHub* findet man eine Kopie des Quellcodes (Mirror) und man kann Änderungsvorschläge, sogenannte *Pull Requests*, eröffnen, wird aber sogleich von einem Bot daran erinnert, wie der eigentliche Prozess ist. Als *Issue Tracker* kommt eine selbst verwaltete *Instanz* der Software *Bugzilla* zum Einsatz. Die Aktivität ist sehr hoch, es werden täglich *Bugs* und *Feature Requests* eingemeldet. Für die Abwicklung der Kontributionen und für Diskussionen gibt es eine grosse Anzahl Mailing-Listen, für jedes technische Thema eine eigene. Auf den genannten Plattformen kann man selbst Konten eröffnen, um am Entwicklungsprozess teilzunehmen. Analoges gilt für die Mailing-Listen, bei denen man sich per E-Mail ein- und austragen kann (subscribe/unsubscribe).

Website `https://kernel.org`

Source Code Repository `https://github.com/torvalds/linux` (Mirror)
`https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git`

Issue Tracker `https://bugzilla.kernel.org`

Mailinglisten `http://vger.kernel.org/vger-lists.html`

5.2.2 Debian GNU/Linux

Debian ist eine *Distribution* des GNU/Linux Betriebssystems, die aus Komponenten von Freier und Open Source Software besteht. Debian selbst ist die Basis mehrerer *Distributionen*, allen

voran Ubuntu, das selbst wiederum Unterdistributionen hat. Technisch gesehen, ist Debian durch das Debian Packaging System (dpkg, Dateiendung `.deb`) charakterisiert, das Ian Murdock für das Projekt 1994 eingeführt hat, um die Verwaltung von Software zu vereinfachen.[11]

Entwicklung & Kollaboration

Debian verwendet eine selbst verwaltete *Instanz* von *GitLab* als *Source Code Repository*. Der Quellcode für alle Projekte, an denen die Debian Community arbeitet, inklusive der Debian Website mit den Übersetzungen in alle Sprachen, findet man dort.[12] Ein grosser Teil der Kommunikation findet unter Nutzung von Mailing-Listen statt. Es gibt zwei Mailing-Listen, die speziell für Einsteiger konzipiert sind bzw. um am Laufenden zu bleiben, ohne sich technisch zu stark involvieren zu müssen.[13]

Website `https://www.debian.org`

Source Code Repository/Issue Tracker `https://salsa.debian.org/public`

Kommunikation `https://lists.debian.org` (*Mailinglisten*)

Dokumentation `https://wiki.debian.org` (Projekt-, Benutzer- und Entwicklerdokumentation)

5.2.3 Ubuntu

Ubuntu ist die populärste Debian-basierte GNU/Linux *Distribution*. Sie wird von der Firma Canonical entwickelt. Ihr Gründer, Mark Shuttleworth, hat Ubuntu im Jahr 2004 gestartet, nachdem er durch einen Internet-Exit ein Milliardenvermögen gemacht hatte.[14] Ubuntu besteht als eigenständiges Community-Projekt und als Produkt, mit Canonical als seinem kommerziellen Entwickler.[15][16][17]

Entwicklung & Kollaboration

Canonical hat seine eigene Kollaborationsplattform, Launchpad, in der *Versionskontrolle* für den Quellcode, *Issue* Tracking, Übersetzungen, *Mailinglisten*, Fragen & Antworten, usw. integriert sind. Als VCS-Tools werden Bazaar und Git unterstützt. Man kann selbständig ein Benutzerkonto eröffnen, um an den Projekten mitzuarbeiten.[18] Die Kommunikation ausserhalb von Launchpad findet auf einer von Ubuntu betriebenen *Discourse-Instanz* und auf einer grossen Anzahl von *Mailinglisten* und IRC-Kanälen (Chats) statt.

[11] Vgl. Software in the Public Interest, *Debian – Reasons to use Debian*.
[12] Vgl. Praveen Arimbrathodiyil, *gitlab - Debian Wiki*.
[13] Vgl. Debian Webmaster Team, *Debian – How to join Debian*.
[14] Vgl. Wikipedia, *Mark Shuttleworth*.
[15] Vgl. Canonical, *Mark Shuttleworth - Canonical Vision*.
[16] Vgl. Ubuntu, *The story of Ubuntu*.
[17] Vgl. Wikipedia, *Ubuntu*.
[18] Vgl. Canonical, *launchpad tour*.

Ubuntu ist ein äusserst grosses Projekt mit vielen Themen. Entsprechend unübersichtlich ist das Angebot an Information und Kommunikationskanälen. Viele Informationen, die die Community z.B. im Wiki zusammengetragen hat, erscheinen veraltet bzw. nicht mehr aktuell. Das hat vermutlich damit zu tun, dass Canonical mit der Zeit seinen Fokus ändert und damit nicht mehr alle Initiativen der Community gleich stark unterstützt. Die Ubuntu Community ist weltweit in lokalen Benutzergruppen, sogenannten *LoCos*, organisiert. Einige wenige davon sind als offiziell anerkannt gekennzeichnet und haben einen Webauftritt im Design von Ubuntu.com.

Website `https://ubuntu.com`

Source Code Repository/Issue Tracker `https://launchpad.net/ubuntu`

Kommunikation `https://discourse.ubuntu.com` (Diskussionsforum)
`https://wiki.ubuntu.com/IRC/ChannelList` (IRC Kanäle)
`https://lists.ubuntu.com` (*Mailinglisten*)

Dokumentation `https://wiki.ubuntu.com` (Projektdokumentation)
`https://help.ubuntu.com` (Benutzerdokumentation)
`https://askubuntu.com` (Q&A Forum)

Community `https://loco.ubuntu.com` (Lokale Ubuntu-Gruppen, weltweit)

5.2.4 GNOME Project

GNOME ist eine *Desktop-Umgebung* für Linux und Unix-ähnliche *Betriebssysteme*. Es ist die Standard-*Benutzeroberfläche* der populärsten Linux *Distributionen*, wie Debian, Ubuntu und *Red Hat Enterprise Linux*. Die einzelnen Komponenten stehen unter der GPL-Lizenz, das von GNOME verwendete GIMP-Toolkit (GTK) steht unter der LGPL-Lizenz.[19][20]

Entwicklung & Kollaboration

GNOME verwendet, wie Debian, eine selbst verwaltete *Instanz* von *GitLab* als *Source Code Repository*. Die Entwicklung aller Applikationen und Komponenten findet dort statt. Kommuniziert wird über *GitLab Issues*, *Merge Requests* und selbst gehosteten *Instanzen* von *Matrix/Element* und *Discourse*. *Matrix* ist, mit seinem Webclient *Element*, eine *Open Source* Kommunikationsplattform (ähnlich *Discord* oder *Microsoft Teams*). *Discourse* ist eine freie Diskussionsforensoftware, auf der systematisch über Probleme rund um GNOME diskutiert wird. Für die Projektdokumentation hat das GNOME Projekt ein eigenes Wiki. Zusätzlich gibt es zwei Informationsportale, eines für Endanwender, das andere für Entwickler.[21]

[19] Vgl. Wikipedia, *Gnome*.
[20] Vgl. GNOME, *Introducing GNOME 3.24 'Portland'*.
[21] Vgl. William Jon McCann, *Getting in Touch*.

Website `https://www.gnome.org`

Source Code Repository/Issue Tracker `https://gitlab.gnome.org`

Kommunikation `https://matrix.to/#/#newcomers:gnome.org` (diverse *Matrix* Kanäle)

`https://discourse.gnome.org` (Diskussionsforum)

Dokumentation `https://wiki.gnome.org` (Projektdokumentation)

`https://help.gnome.org` (Benutzerdokumentation)

`https://developer.gnome.org` (Entwicklerdokumentation)

5.2.5 GIMP

GIMP, ein Akronym für *GNU Image Manipulation Program*, ist ein mächtiges Bildverarbeitungsprogramm für Fotos und ähnliche Pixel-basierte Grafiken.[22] Das Programm ist plattformunabhängig, d.h. man kann es unter *GNU/Linux*, *Windows* und *macOS* gleichermassen verwenden. Häufig wird es mit seinem proprietären Konkurrenten und kommerziellen Platzhirschen Adobe Photoshop verglichen. Zur Ergänzung für das Arbeiten im Bereich Desktop Publishing empfiehlt die GIMP-Website weitere bekannte *Freie Software*, z.B. das Publishing-Programm *Scribus* und das Vektorgrafikprogramm *Inkscape*.

Geschichte

Das Projekt wurde 1995 von Peter Mattis und seinem Studienkollegen Spencer Kimball gestartet. Kimball konzentrierte sich dabei auf GIMP selbst, Mattis arbeitete (ab 1996) vorwiegend an GTK, dem GIMP-Toolkit, das die Basis für die *Benutzeroberfläche* von GIMP ist.[23] Larry Ewing zeichnete mit einem der ersten Releases von GIMP „Tux“, den Linux-Pinguin, das Maskottchen von GNU/Linux.[24] Die gesamte Geschichte des Projekts ist sehr ausführlich auf der GIMP-Website beschrieben.

Entwicklung & Kollaboration

GIMP hosted seinen Quellcode auf der *GitLab-Instanz* des GNOME-Projekts. Das gleiche gilt für GTK, das GIMP-Toolkit. Die Zusammenarbeit ist ähnlich dem GNOME-Projekt gestaltet und doch unabhängig davon. Zur Kommunikation wird neben (dem unfreien) *Discord* weiterhin der Internet Relay Chat (IRC) verwendet, der Zugang dazu ist aber durch eine sog. „*Matrix*-Bride“ erleichtert, d.h. man kann wie im GNOME-Projekt z.B. den *Matrix* Webclient (oder Smartphone App) *Element* verwenden. Als Diskussionsforen steht neben dem schon länger bestehenden Forum auf einer externen Website die *Discourse-Instanz* von GNOME zur Verfügung. Dokumentation steht für Endbenutzer und Entwickler separat zur Verfügung.

[22] Vgl. The GIMP Team, *GIMP - About GIMP*.
[23] Vgl. Wikipedia, *Peter Mattis*.
[24] Vgl. Ewing, *Penguin Tutorial - Notes on creation*.

Website `https://www.gimp.org`

Source Code Repository/Issue Tracker `https://gitlab.gnome.org/GNOME/gimp`

Kommunikation `https://discord.com` (mit Einladungslink von `gimp.org/discuss.html`)
`https://matrix.to/#/#gimp:gnome.org` (diverse IRC-Kanäle, mit *Matrix*-Bridge)
`https://discuss.pixls.us/c/software/gimp/24` (Haupt-Diskussionsforum)
`https://discourse.gnome.org/tag/gimp` (zusätzliches Forum im GNOME-Projekt)

Dokumentation `https://docs.gimp.org` (Benutzerdokumentation)
`https://developer.gimp.org` (Entwicklerdokumentation)

5.2.6 GitLab

GitLab ist eine Social Coding-Plattform mit umfangreichen Features für einen modernen Softwareentwicklungsprozess. Das Produkt wird in der Regel selbst gehostet, man kann aber auch einfach die öffentliche Instanz gitlab.com nutzen.

Geschichte

GitLab ist eine *Ruby on Rails*-Applikation, die die Ukrainer Dmitri Saparoschez und Valery Sizov 2011 als Konkurrent von *GitHub* zu entwickeln begannen. 2014 wurde die Firma GitLab B.V. in Utrecht gegründet, im Oktober 2021 ging das Unternehmen an der Börse.

Entwicklung & Kollaboration

Die Entwicklung von *GitLab* findet vollständig auf *GitLab* selbst statt.[27]

Website `https://about.gitlab.com`

Source Code Repository/Issue Tracker `https://gitlab.com/gitlab-org/gitlab`

Dokumentation `https://docs.gitlab.com` (Benutzerdokumentation)
`https://handbook.gitlab.com` (Organisationshandbuch)

5.2.7 ILIAS

ILIAS ist ein webbasiertes Learning Management System. Es wurde 1997 an der Universität zu Köln entwickelt. Ziel war, den Präsenzunterricht im Hörsaal durch IT-gestützte Unterrichts-

methoden zu erweitern und so zeit- und ortsunabhängiges Studieren zu ermöglichen. Die Softwarelösung erregte an Hochschulen bald reges Interesse und wurde aus diesem Grund im Jahr 2000 unter die freie GPL-Lizenz gestellt. Seit dem Jahr 2009 obliegt die Betreuung und Weiterentwicklung der Software dem Verein *ILIAS open source e-Learning e.V.*.[25] Heute wird ILIAS neben Hochschulen auch an Schulen für die Unterrichtsbegleitung und in Unternehmen für die Mitarbeiterschulung eingesetzt.[26]

Entwicklung & Kollaboration

Die Entwicklung von ILIAS findet auf *GitHub* statt, wo der Quellcode gehostet wird und man mit *Pull Requests* Quellcodeänderungen vorschlagen kann. Als *Issue Tracker* wird eine dedizierte *Instanz* der Freien Software mantis *Bug Tracker* verwendet. Die Chat-Plattform *Discord* wird für schnellen und niederschwelligen Austausch zwischen Benutzern und Entwicklern verwendet. Die restliche Projektkoordination und -dokumentation findet unter Einsatz von ILIAS selbst statt.[27]

Website `https://www.ilias.de`

Source Code Repository `https://github.com/ILIAS-eLearning/ILIAS`

Issue Tracker `https://mantis.ilias.de`

Kommunikation `https://discord.com` (mit Einladungslink von `docu.ilias.de`)

`https://docu.ilias.de/goto_docu_frm_9462.html` („ILIAS Developers“ Forum)

Dokumentation `https://docu.ilias.de` (Projektdokumentation)

`http://www.ilias.de/docu/goto_docu_cat_581.html` (Benutzerdokumentation)

`http://ildoc.hrz.uni-giessen.de` (Technische Dokumentation, automatisch generiert)

5.2.8 Moodle

Moodle ist eine *Open Source*-Lernplattform. Der Name ist ein Akronym und steht für *Modulare objekt-orientierte, dynamische Lernumgebung.*

Hintergründe & Geschichte

Moodle wird seit 1999 von Martin Dougiamas (Australien) entwickelt. Die Lernplattform entstand im Rahmen einer nie abgeschlossenen Promotionsarbeit. *Moodle* ging 2002 das erste Mal live. Heute ist *Moodle* an zahlreichen Schulen, Universitäten und privaten Unternehmen im Einsatz.

[25] Vgl. Wikipedia, *ILIAS*.
[26] Vgl. sr.solutions, *01. Ausbildungsplanung mit ILIAS in der Industrie*.
[27] In Entwicklerkreisen sagt man zu dieser Praxis scherzhaft, „Eating your own dog food“.

Entwicklung & Kollaboration

Das *Moodle*-Projekt hosted seinen Quellcode auf einer eigenen *GitLab-Instanz*. *Bug Reports* und *Feature Requests* werden über eine Cloud-*Instanz* von *Atlassian Jira* gemanaged (vermutlich mit einer Atlassian *Open Source*-Lizenz).

Produkt- und Community-Dokumentation findet sich auf der *Moodle*-Website. Für die technische Dokumentation wird ein Wiki verwendet, eine selbst gehostete *Instanz* der *Open Source*-Software *MediaWiki*.[28]

Website `https://moodle.com` (Product & Community Website)

Source Code Repository `https://git.in.moodle.com/moodle/moodle` (*GitLab Instanz*)

Issue Tracker `https://tracker.moodle.org` (Atlassian Jira *Instanz*)

Dokumentation `https://docs.moodle.org`

5.2.9 BigBlueButton

BigBlueButton ist eine *Open Source*-Lösung für Webkonferenzen und virtuelle Klassenzimmer. Über Plugins werden Lern- und Kollaborationsplattformen wie ILIAS, Moodle, Nextcloud u.v.m. einfach integriert.[29] Mehrere Universitäten, unter anderem in Österreich in Graz, Klagenfurt und Innsbruck, setzen das Produkt für Webkonferenzen und Distance-Learning ein.[30]

Die Plattform bietet starke Features an, wie z.B. das Präsentationstool, ein White Board mit der Möglichkeit zum Zoomen und Zeichnen, öffentliche und private Chats, *Breakout-Sessions*, eine Umfrage-Funktion, Up- und Download von Dokumenten und Notizen, VoIP-Integration (FreeSWITCH), Videoaufnahme, Streaming durch URL (z.B. über YouTube).

Das System kann auf eigenen Servern betrieben werden und läuft gemäss Installationsanleitung auf *Ubuntu Linux*.[31] Die Datenübertragung erfolgt verschlüsselt. Auf die Einhaltung des Datenschutzes wird Wert gelegt.[32] Es gibt zahlreichen Angebote im Internet, die das Produkt gehostet, als Service anbieten.

Geschichte & Philosophie

Gestartet wurde das Projekt im Jahr 2007 von Richard Alan an der Carleton University, Kanada.[33] Der Name BigBlueButton stammt von der ursprünglichen Idee, dass das Starten einer Webkonferenz so einfach sein sollte wie das Drücken eines grossen blauen metaphorischen Knopfes.[34] Diese Idee ist in der Software sichtbar umgesetzt. Beispielsweise wird beim Wechseln von

[28] Vgl. Wikimedia, *MediaWiki*.
[29] Vgl. BigBlueButton, *BigBlueButton LMS Integrations | Designed for Schools | BigBlueButton*.
[30] Vgl. Wikipedia, *BigBlueButton*.
[31] Vgl. BigBlueButton, *Install BigBlueButton | BigBlueButton*.
[32] Vgl. BigBlueButton, *Privacy | BigBlueButton*.
[33] Vgl. Wikipedia, *BigBlueButton*.
[34] Vgl. BigBlueButton, *FAQs | BigBlueButton*.

Räumen (z.B. bei *Breakout-Sessions*) automatisch eine Tonkontrolle durchgeführt, ohne dass man sich selbst darum kümmern muss.

Entwicklung & Kollaboration

BigBlueButton ist ein grosses Java-Projekt, das aus mehreren Komponenten besteht. Die Entwicklung und das *Issue* Tracking findet auf *GitHub* statt. Wie man in die Entwicklung einsteigt, ist ausführlich in der technischen Dokumentation beschrieben.[35] Für die optische Gestaltung gibt es einen ausführlichen Design Guide.[36] Weitere Möglichkeiten, zum Projekt beizutragen (z.B. Übersetzen, Testen) sind am Ende der Benutzerdokumentation[37] und in den FAQs[38] erklärt. Die Kommunikation findet in 3 Gruppen auf *Google Groups* statt.

Website `https://bigbluebutton.org`

Source Code Repository/Issue Tracker

`https://github.com/bigbluebutton/bigbluebutton`

Kommunikation

`https://groups.google.com/g/bigbluebutton-setup` (Setup Forums)

`https://groups.google.com/g/bigbluebutton-users` (Users Forums)

`https://groups.google.com/g/bigbluebutton-dev` (Developers Forums)

Dokumentation `https://docs.bigbluebutton.org`

5.2.10 LibreOffice

LibreOffice ist eine freie, leistungsstarke und quelloffene Software, die Applikationen für typische Büroarbeiten zur Verfügung stellt. Das Office-Paket beinhält Programme für Textverarbeitung, Tabellenkalkulation, Präsentation, zum Erstellen von Zeichnungen, ein Datenbank-Managementsystem und einen Formeleditor. Es wird häufig mit dem proprietären Konkurrenten *Microsoft Office 365* verglichen.

LibreOffice ist kompatibel mit vielen Dokumentformaten wie *Microsoft Word*, *Excel*, *PowerPoint* und *Publisher* und bietet einen modernen und offenen Dokumentenstandard, das *OpenDocument Format (ODF)* an.[39]

Bewegte Geschichte

Das *LibreOffice*-Projekt hat eine interessante Vorgeschichte. Die Software erblickte unter dem Namen *Star Writer* als Produkt der Hamburger Star Division GmbH das Licht der Welt. Die

[35] Vgl. BigBlueButton, *Development Guide | BigBlueButton.*
[36] Vgl. BigBlueButton, *Development Design Guide | BigBlueButton.*
[37] Vgl. BigBlueButton, *new-features | BigBlueButton.*
[38] Vgl. BigBlueButton, *FAQs | BigBlueButton.*
[39] Vgl. The Document Foundation, *Was ist LibreOffice? - LibreOffice.*

Firma wurde 1985 vom damals 16-jährigen Marco Börris gegründet, der durch einen Schüleraustausch mit dem Silicon Valley dazu inspiriert wurde, eine Alternative zum Office-Paket von *Microsoft* auf den Markt zu bringen.[40]

Das spätere *StarOffice* enthielt neben der Applikation für die Textverarbeitung viele weitere Programme, wie z.B. Grafiksoftware, Datenbank, Terminplaner, E-Mail Client und sogar einen Web-Browser. Die Stärke von *StarOffice* war, dass es auf unterschiedlichen Betriebssystemen lauffähig war.[41] Aus diesem Grund konnte es sich gegen die damals zahlreiche Konkurrenz durchsetzen, bis am Ende *Microsoft* als grössten Gegenspieler übrig war.

1998 wurde die Software schliesslich für alle Privatanwender zur kostenlosen Nutzung freigegeben.[42] Ein Jahr später verkaufte der „deutsche Bill Gates“ seine Firma und Software an den US-amerikanischen Computerkonzern Sun Microsystems. Plan von Sun war es, die Office Suite im Internet lauffähig zu machen, um *Microsoft* den Markt streitig zu machen.[43]

Sun stellte *StarOffice* ein Jahr nach dem Kauf auch für Unternehmen kostenlos zur Verfügung. Danach wurde der Quellcode offengelegt und unter dem Projektnamen *OpenOffice.org (OOo)* weiterentwickelt.[44] Im Jahr 2010 wurde die Firma Sun von Oracle aufgekauft.

Das Projekt *LibreOffice* wurde gegründet, weil Oracle die Namensrechte an *OpenOffice.org* nicht freigeben wollte und weil OOo-Entwickler mit Oracle generell unzufrieden waren.[45] Heute ist *LibreOffice* ein grosses, internationales Projekt des Vereins *The Document Foundation* mit Beteiligung bekannter Namen wie z.B. *Google*, *Red Hat*, *Intel* und der *Free Software Foundation*.[46]

Kritikpunkt Ästhetik

Gefühlt der grösste Kritikpunkt an *LibreOffice* ist die Ästhetik. Neben Problemen mit der „Kompatibilität“, wie aus Anworten der Umfage in Kapitel 8 hervorgeht, wird die Ablehnung häufig mit Hässlichkeit der Bedieneroberfläche der Software begründet.[47]

Selbst die Cloudversion *LibreOffice Online*, die man im *Open Education Server* vorfindet, schneidet im Vergleich zu den beiden Konkurrenten *Microsoft Office 365* und der *Google Workspace* Suite subjektiv schlechter ab, wie die Screenshots 5.2, 5.3, 5.4 zeigen.

LibreOffice ist aber *Freie Software*, deshalb ist die Entscheidung hier – zum Glück – nicht endgültig: Aktuelle Artikel der *LibreOffice*-Dokumentation zeigen, welche einfachen, rasch umsetzbaren Möglichkeiten für die Anpassung der *Benutzeroberfläche* es gibt.[48]

Das letzte Wort ist also nicht gesprochen und es liegt nur an einem Kontributor im Projekt, das aktuelle Default-Icon-Theme durch eines zu ersetzen, das sich breiter Beliebtheit erfreut. Im Ubuntu-Projekt beispielsweise wurde dies schon vor knapp drei Jahren versucht,[49] fehlt also nur, dies generell umzusetzen, um die Beliebtheit der Software nachhaltig zu steigern – und ihr damit eine faire Chance zu geben.

[40] Vgl. Wikipedia, *Marco Börries*.
[41] Vgl. Wikipedia, *StarOffice*.
[42] Vgl. Manfred Dworschak, *Kalkulierter Selbstmord - DER SPIEGEL*.
[43] Vgl. Klaus-Peter Kerbusk, *Brücke in die neue Welt - DER SPIEGEL*.
[44] Vgl. Wikipedia, *Star Division*.
[45] Vgl. Wikipedia, *Apache OpenOffice*.
[46] Vgl. Wikipedia, *LibreOffice*.
[47] Vgl. Community, *LibreOffice looks DATED and PIXELATED. Is this intentional? - English - Ask LibreOffice*.
[48] Vgl. Community, *Change LibreOffice Default Look and Feel - libreofficehelp.com*.
[49] Vgl. Sneddon, *LibreOffice Has a New Look in Ubuntu 20.10 - OMG! Ubuntu!*

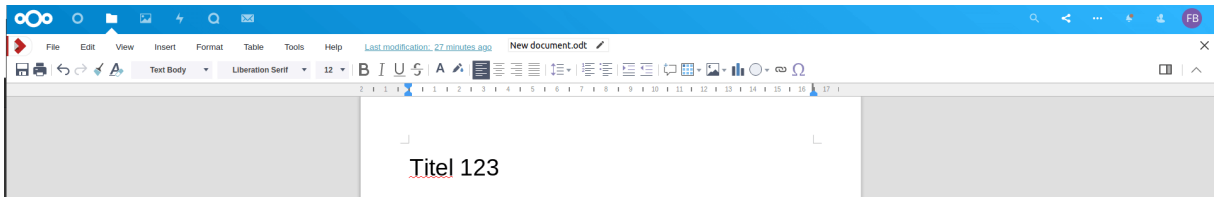

Abbildung 5.2: LibreOffice Online auf Nextcloud

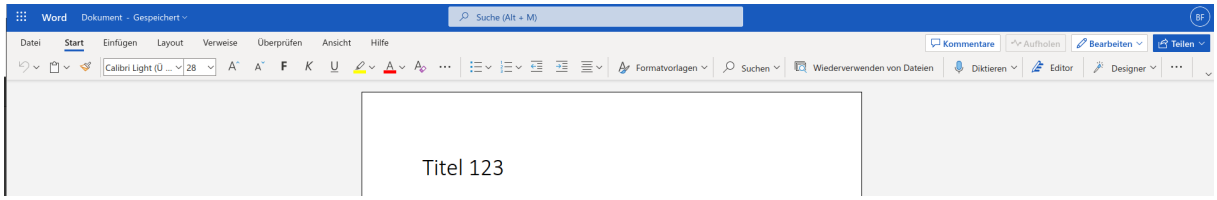

Abbildung 5.3: Microsoft Office 365

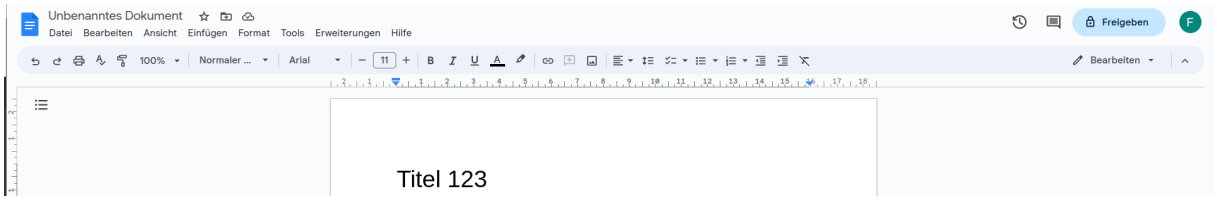

Abbildung 5.4: Google Workspace

Entwicklung & Kollaboration

Für die Entwicklung von *LibreOffice* verwendet die Document Foundation eine Reihe unterschiedlicher Werkzeuge, die alle über die offizielle Homepage gut verlinkt auffindbar sind.

Der Quellcode wird mit einer selbst gehosteten *Instanz* der *Open Source*-Software *Gerrit*, ein von *Google* weiterentwickeltes Code Review Tool,[50] verwaltet. Für *Bug Reports* kommt die alte, aber bewährte *Open Source*-Software *Bugzilla* zum Einsatz.[51] Die Kommunikation findet auf zahlreichen Kanälen der IRC-Plattform *Libera.Chat* statt.[52] Wie beim *GIMP*-Projekt kann man aber auch einen *Matrix*-Client zur Kommunikation verwenden, da eine *Matrix*-Bridge für IRC konfiguriert ist. Für die technische Dokumentation wird ein Wiki verwendet, eine selbst gehostete *Instanz* der *Open Source*-Software *MediaWiki*.[53]

Website `https://libreoffice.org`

Entwicklungswerkzeuge & Kommunikation

`https://devcentral.libreoffice.org` (z.B. *Gerrit*, *Bugzilla*, *Weblate*, IRC)

Source Code Repository / Code Review `https://gerrit.libreoffice.org`

[50] Vgl. Gerrit, *Gerrit's History | Gerrit Code Review*.
[51] Vgl. Bugzilla, *Installation List - Bugzilla*.
[52] Vgl. Libera Chat contributors, *Guides | Libera Chat*.
[53] Vgl. Wikimedia, *MediaWiki*.

Issue Tracker `https://bugs.documentfoundation.org`

Dokumentation `https://wiki.documentfoundation.org`

Teil II

Anwendung

Kapitel 6

Selbstversuche

Zu Hause verwenden wir auf den Computern *Freie Software*, wo immer möglich. Ich kann daher schon länger aus einem Erfahrungsschatz schöpfen. Zusätzlich konnte ich im Rahmen dieser Arbeit einige neue Probleme hands-on selbst lösen. Dieses Kapitel dokumentiert meine Erfahrungen.

6.1 GNU/Linux am eigenen PC

GNU/Linux ist ein vollwertiges *Betriebssystem*, das den beiden populären, proprietären Alternativen in Wenigem hintendrein steht. In vielen Aspekten ist es ihnen technisch deutlich überlegen (z.B. Systemsicherheit, Stabilität, Ressourcennutzung, Benutzerfreundlichkeit des Desktops).

Schon bevor ich an die Pädagogische Maturitätsschule kam, arbeitete ich an meinem Computer mit *Ubuntu Linux*. Ich war in der Klasse immer der Einzige, der weder Windows noch *macOS* hatte. Meinen Mitschülern ist dies zwar aufgefallen, es war ihnen aber scheinbar gleichgültig, denn ich wurde deshalb weder gehänselt noch bewundert. Teilweise kam das Vorurteil auf, GNU/Linux sei nur für Programmierer oder man müsste, um GNU/Linux zu benutzen, programmieren können.

Beim IT-Support gab es von einem Mitarbeiter abfällige Kommentare. Ich bekam die Empfehlung, *Windows* zu installieren, weil ich Schwierigkeiten beim Verbinden mit dem Wi-Fi und mit *Citrix* hatte. Ein zweiter Kollege war verständnisvoller und gab mir Tipps. Die Probleme konnte ich am Ende eigenständig lösen.

6.1.1 Fazit

Mit *Ubuntu Linux* kann ich alle Aufträge an der Schule ohne Einschränkung erfüllen. *Windows*-Software, die von der Schule zur Verfügung gestellt wird, kann ich mit *Citrix* problemlos nutzen. Bei Konfigurationsproblemen ist man im Wesentlichen auf sich alleine gestellt, weil sowohl den Kolleg:innen als auch dem IT-Personal das Know-how fehlt. Wie auch für *Windows* und *macOS* findet man aber im Internet Lösungsmöglichkeiten.

Das Hauptargument pro *Windows*, die Verfügbarkeit vieler Spiele, ist für mich wenig relevant. Es reicht mir, am Smartphone Spiele zu spielen. In Bezug auf Textverarbeitung, Tabellenkalkulation und Präsentationen greife ich meist auf *Google Docs* zurück, anstatt *Microsoft Office 365* in unserer Schulinfrastruktur zu nutzen. Leider ist die *Benutzeroberfläche* von *LibreOffice* im Vergleich zu *Microsoft Office 365* etwas unästhetisch, weswegen es mir keinen grossen Spass macht, es generell zu nutzen. In letzter Zeit habe ich das aber doch getan und habe dabei zu meiner

eigenen Überraschung keinen Unterschied in der Leistungsfähigkeit bemerkt. Das hat mich in meiner Vermutung bestärkt, dass es keine wahrlich sachlichen Argumente gibt, mit denen LibreOffice abgelehnt werden kann. Es handelt sich eher um eine Art Gruppenzwang, es gilt als „uncool“, etwas anderes als das etablierte *Office 365* zu verwenden.

6.2 LaTeX zur Textverarbeitung

Zum Verfassen von grösseren Arbeiten an der Schule habe ich in den letzten drei Jahren statt *Microsoft Word* LaTeX in Kombination mit *GitLab* verwendet.

LaTeX[1] ist kein Textverarbeitungsprogramm, sondern eine Art Programmiersprache. Man schreibt Text in einfachen Textdateien (mit der Endung `.tex`) mit einem Texteditor, wie ihn beispielsweise Programmierer verwenden. Teilen des Textes kann man mit bestimmten Textsequenzen („Befehlen“) eine besondere Bedeutung geben (z.B. Überschrift, Unterüberschrift, Grafik, Inhaltsverzeichnis, usw.). Mit einem Übersetzungsprogramm („LaTeX Compiler“) kann man aus dem Text und den enthaltenen Befehlen ein druckfertiges Dokument (z.B. PDF) generieren.

6.2.1 Einfaches Arbeiten mit VSCodium

Wir müssen ein Dokument abliefern, am besten ohne Rechtschreibfehler, und wollen schon während des Schreibens sehen, wie das fertige Dokument aussehen wird, ohne jedes Mal dafür ein PDF-Dokument generieren zu müssen. Ausserdem kennen wir die vielen LaTeX-Befehle nicht, um Text zu formatieren, Links und Bilder einzufügen, ein Inhaltsverzeichnis, Literaturverzeichnis, Abbildungsverzeichnis, usw. darzustellen. Gibt es Programme, die uns dabei unterstützen können?

VSCodium[2] ist ein freier Texteditor; es handelt sich um die von der Community frei paketierte Version von *Microsoft Visual Studio Code*. Mit zwei Extensions für LaTeX (*latex-workshop*, *vscode-ltex*) wird er zu einer vollwertigen Entwicklungsumgebung zum Schreiben von LaTeX-Dokumenten, wenn man sich eine gängige LaTeX-*Distribution* auf seinem Rechner installiert hat. Eine integrierte Vorschau aktualisiert das PDF-Dokument während man tippt. Falsch geschriebene Wörter werden blau unterwellt, auf Mausklick bekommt man passende Korrekturvorschläge. Ein Seitenpanel bietet LaTeX-Befehle, Symbole zum Einfügen und eine Dokumentstruktur zum Navigieren an.

Bei einem grossen Projekt ist es wichtig, die Übersicht nicht zu verlieren. Dazu kann man das Dokument in die einzelnen Kapitel aufteilen, also eine eigene Datei pro Kapitel verwalten, die im Hauptdokument inkludiert wird.

6.2.2 Einhalten formaler Vorgaben leicht gemacht

Bei einer wissenschaftlichen Arbeit ist es wichtig, formale Aspekte einzuhalten. Zum Beispiel müssen das Literaturverzeichnis und alle Referenzen im Dokument ein bestimmtes Format einhalten. LaTeX macht das Anpassen dieser Ausgabeformate leicht. Entweder gibt es eingebaute Befehle oder LaTeX-Pakete ("packages"), die eine schnelle Anpassung ermöglichen. Damit genügt meist das Einfügen weniger Zeilen, um das gewünschte Ergebnis zu erreichen.

[1] `https://www.latex-project.org`
[2] `https://vscodium.com`

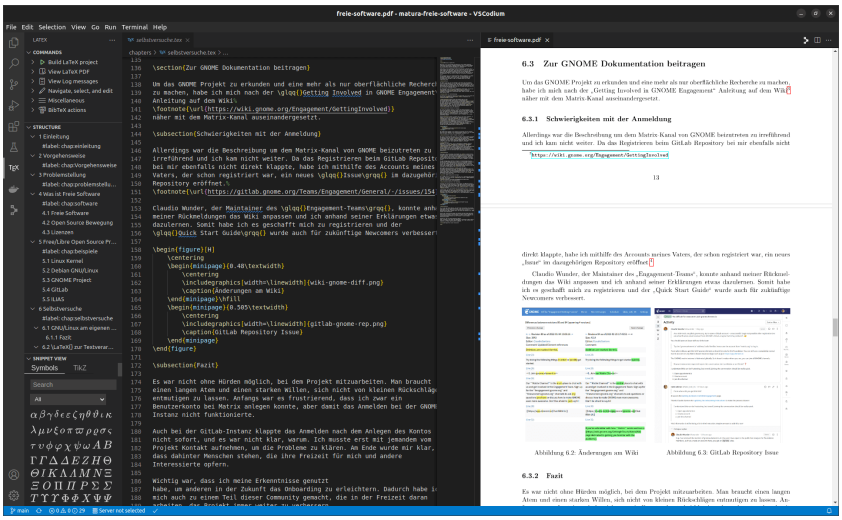

Abbildung 6.1: VSCodium mit *latex-workshop* Extension

Beispiel Angenommen, es ist gefordert, dass die Abbildungen durchgehend nummeriert werden (statt im Format „Abbildung *Abschnitt.Nummer*: *Beschreibung*“). Recherchiert man im Internet, z.B. mit „*latex list of figures continuous without section numbering*“, wird man schnell fündig: Ein Blog, ein Beitrag auf StackExchange und die Overleaf Dokumentation erklären Lösungsmöglichkeiten, die man einfach übernehmen kann. Fügt man den entsprechenden Code in das Hauptdokument ein, passt LaTeX sofort alle Referenzen und das Abbildungsverzeichnis der Arbeit automatisch an.

6.2.3 Fazit

Ursprünglich kam der Vorschlag zum Verwenden von LaTeX von meinen Eltern. Sie hatten in ihrer Studienzeit schon damit gearbeitet. Der Einstieg war aber trotzdem schwierig, denn ich musste ein geeignetes Programm zum Arbeiten mit LaTeX finden. Zum Glück hatte ich das Problem nicht alleine, denn meine ältere Schwester wollte auch schon mit LaTeX ihre Maturaarbeit schreiben. Die Suche nach den Arbeitsinstrumenten wurde somit zur Aufgabe für die ganze Familie.

Mit dem fertigen Setup, *VSCodium* mit Extensions für Rechtschreibprüfung und automatisches PDF-Generieren, sowie *GitLab* für die begleitende Projektwebsite, kam ich mit der Online-Dokumentation und Tutorials im Internet gut voran. Ich war aber dankbar, wenn ich auf meine Familienmitglieder zurückgreifen konnte, wenn es Layoutprobleme oder Fehlermeldungen gab, mit denen ich nicht sofort etwas anfangen konnte. Eine „Community“ braucht es also zum Überwinden von Schwierigkeiten und zum Teilen des Erfolgs!

Heute möchte ich das Arbeiten mit LaTeX nicht mehr missen. Das schöne, einheitliche Layout, die Leichtigkeit und Übersichtlichkeit beim Überarbeiten des Dokuments sind Dinge, die *Microsoft Word* nicht leisten kann. Wenn ich uneinheitlich gelayoutete Dokumente meiner Kolleg:innen sehe und daran denke, wie sie Rückmeldungen der Lehrperson kurz vor dem Abgabetermin mühsam einarbeiten müssen, denke ich, dass ich mit LaTeX auf das richtige Pferd gesetzt habe.

6.3 Projektwebsite mit GitLab Pages

GitLab stellt mit dem Feature *GitLab Pages* eine schnelle Möglichkeit zum Publizieren einer Website zur Verfügung. Man braucht also keinen Webserver anzumieten oder selbst zu betreiben, *GitLab* tut das schon für uns.

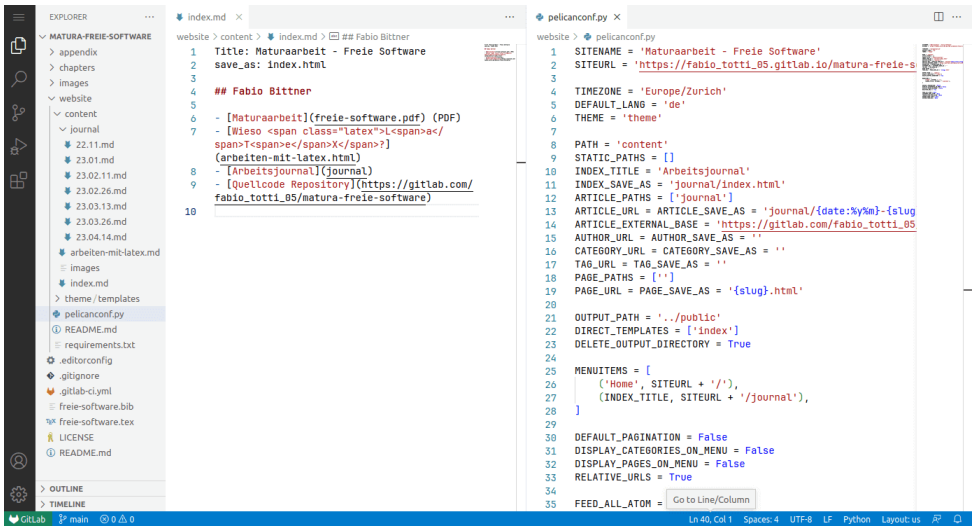

Abbildung 6.2: MarkDown-Eingabedateien und die SSG-Konfiguration

Technisch gesprochen ist diese Funktionalität auf statische Webseiten beschränkt, d.h. man kann keine Datenbanken und keine am Server laufenden Technologien (Programmiersprachen) verwenden. Um mit HTML, CSS und JavaScript eine vollwertige Website zu erstellen, kann man sich mit Static Site Generatoren (SSG) behelfen.[3]

6.3.1 Mit SSG von HTML zu MarkDown

Den Anfang der Projektwebsite habe ich rein mit HTML gemacht. Die Struktur und Konfiguration für *Gitlab Pages* konnte ich von einem meiner früheren Schulprojekte übernehmen.[4] Zum Aufhübschen habe ich das Stylesheet LaTeX.css von Vincent Dörig verwendet.[5] Damit sieht die Website vom Stil her so aus wie das PDF-Dokument.

Mit dem Arbeitsjournal ist immer mehr HTML hinzugekommen. Mit der Zeit machte es Sinn, das HTML zu generieren, damit ich mich auf den Inhalt konzentrieren und ich das Copy-und-Paste von HTML-Code vermeiden konnte. Ich entschied mich für den *Pelican* SSG, weil er populär ist und er auf Python basiert.

Begonnen habe ich mit der Umstellung auf *Pelican* mit dem Quickstart aus der *Pelican*-Dokumentation.[6] Damit habe ich die Projektstruktur mit den grundlegenden Dateien erstellt, um die bisherige Website aus MarkDown heraus zu generieren. Ohne besondere Konfiguration verwendet *Pelican* sein Default-Theme. Um die generierte Website wieder mit dem LaTeX-Design darzustellen, musste ich nach Anleitung der Dokumentation[7] ein einfaches *Pelican*-Theme erstellen. Der Einfachheit halber habe ich Blog-Funktionalität wie Autoren, Kategorien, Tags und Archiv weggelassen.

6.3.2 Fazit

Das Aufsetzen des SSG-Setups ist nicht ganz einfach. Man braucht etwas Erfahrung oder Unterstützung von jemandem, der so etwas schon einmal gemacht hat. Sobald man diese Hürde einmal

[3] Vgl. Netlify, *Site Generators - A List of Static Site Generators for Jamstack Sites*.
[4] Vgl. Bittner, *GitLab - Fabio Bittner / Deutsch SLA*.
[5] Vgl. Dörig, *LaTeX.css — Make your website look like a LaTeX document*.
[6] Vgl. Justin Mayer, *Quickstart - Pelican 4.8.0*.
[7] Vgl. Justin Mayer, *Themes - Pelican 4.8.0*.

genommen hat, ist das Editieren von Seiten aber sehr einfach. Man kann sich wirklich auf den Inhalt konzentrieren. Die Syntax von MarkDown kann man leicht lernen bzw. von bestehenden Dokumenten (z.B. in *Open Source*-Projekten) abschauen oder im Web recherchieren.

6.4 Zur GNOME Dokumentation beitragen

Um das GNOME Projekt zu erkunden und eine mehr als nur oberflächliche Recherche zu machen, habe ich mich nach der „Getting Involved in GNOME Engagement“ Anleitung auf dem Wiki[8] näher mit dem Matrix-Kanal auseinandergesetzt.

6.4.1 Schwierigkeiten mit der Anmeldung

Allerdings war die Beschreibung um dem *Matrix*-Kanal von GNOME beizutreten zu irreführend und ich kam nicht weiter. Da das Registrieren beim GitLab Repository bei mir ebenfalls nicht direkt klappte, habe ich mithilfe des Accounts meines Vaters, der schon registriert war, ein neues *Issue* im dazugehörigen *Source Code Repository* eröffnet.[9]

Claudio Wunder, der Maintainer des „Engagement-Teams“, konnte anhand meiner Rückmeldungen das Wiki anpassen und ich anhand seiner Erklärungen etwas dazulernen. Somit habe ich es geschafft mich zu registrieren und der „Quick Start Guide“ wurde auch für zukünftige Newcomers verbessert.

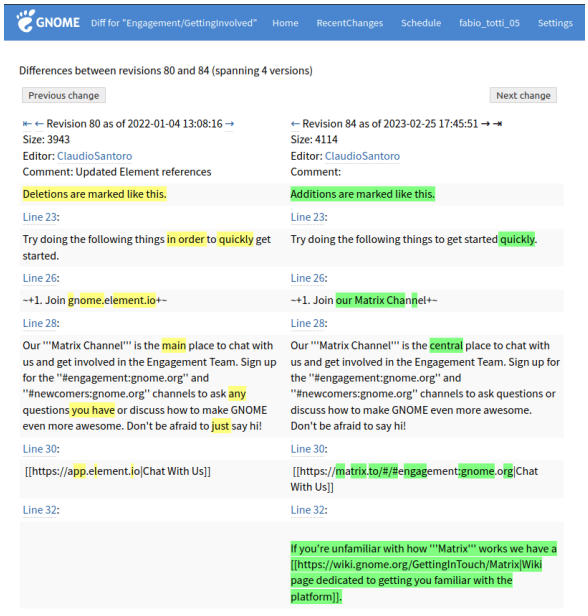

Abbildung 6.3: Änderungen am Wiki

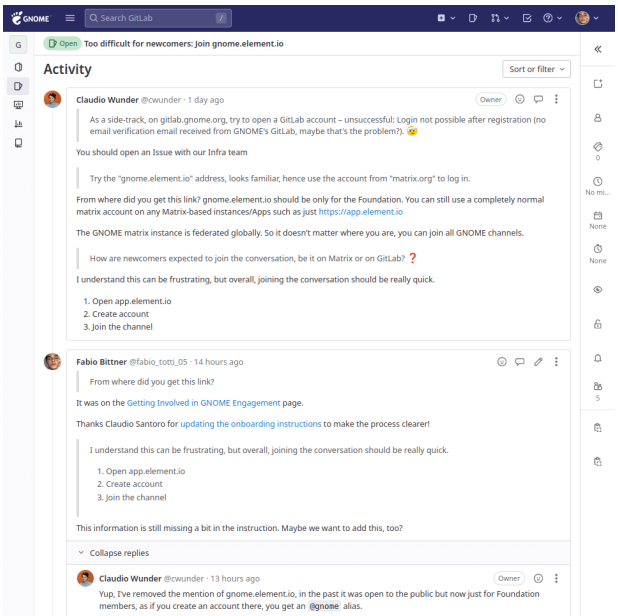

Abbildung 6.4: GitLab Repository Issue

6.4.2 Fazit

Es war nicht ohne Hürden möglich, bei dem Projekt mitzuarbeiten. Man braucht einen langen Atem und einen starken Willen, sich nicht von kleinen Rückschlägen entmutigen zu lassen. Anfangs war es frustrierend, dass ich zwar ein Benutzerkonto bei *Matrix* anlegen konnte, aber damit das Anmelden bei der GNOME *Instanz* nicht funktionierte.

[8] `https://wiki.gnome.org/Engagement/GettingInvolved`
[9] `https://gitlab.gnome.org/Teams/Engagement/General/-/issues/154`

Auch bei der *GitLab Instanz* klappte das Anmelden nach dem Anlegen des Kontos nicht sofort, und es war nicht klar, warum. Ich musste erst mit jemandem vom Projekt Kontakt aufnehmen, um die Probleme zu klären. Am Ende wurde mir klar, dass dahinter Menschen stehen, die ihre Freizeit für mich und andere Interessierte opfern.

Wichtig war, dass ich meine Erkenntnisse genutzt habe, um anderen in der Zukunft das Onboarding zu erleichtern. Dadurch habe ich mich auch zu einem Teil dieser Community gemacht, die in der Freizeit daran arbeiten, das Projekt immer weiter zu verbessern.

6.5 Korrektur im ILIAS Development README

Um die Mitarbeit beim ILIAS-Projekt auszuprobieren, habe ich einen *Pull Request* auf *GitHub* eröffnet, mit dem ich die Korrektur von einem Schreibfehler im README vorschlage.[10]

Der *Pull Request*, eine einfache Textänderung, wurde drei Tage später von Richard Klees, einem Mitarbeiter des ILIAS-Projekts, gemerged.

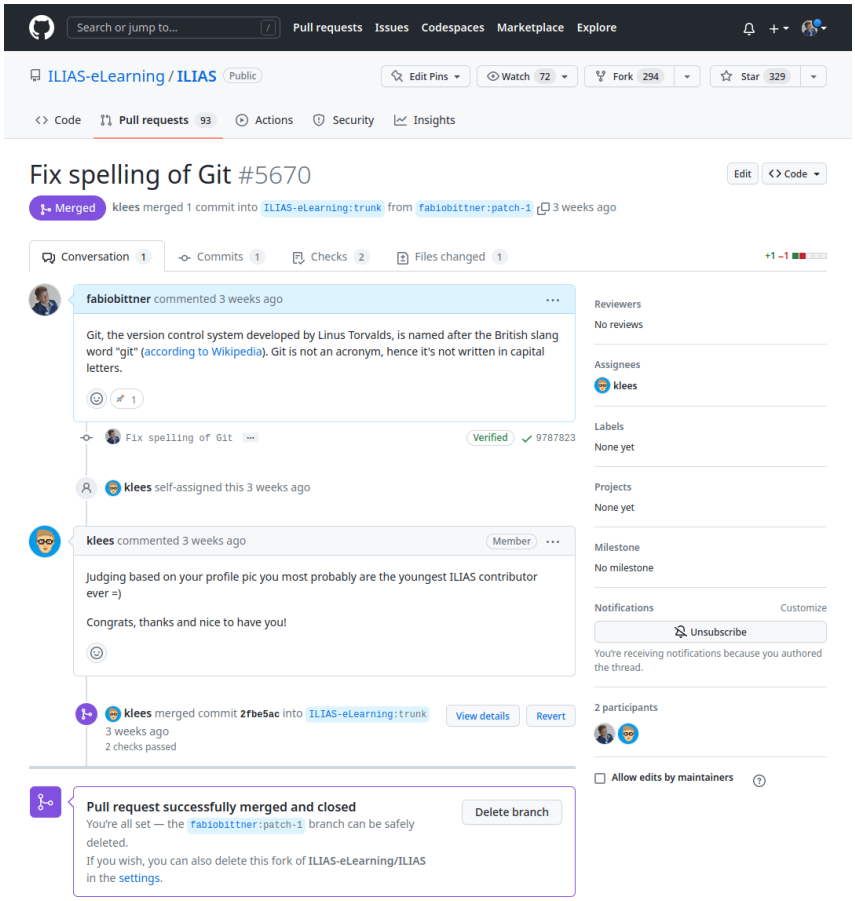

Abbildung 6.5: GitHub Pull Request

6.5.1 Fazit

Dieser Beitrag zum *Open Source*-Projekt ILIAS ist ein Beispiel einer Änderung, die im Grunde jede:r machen kann. Ich bin über den ‚Rechtschreibfehler' zufällig gestossen, als ich das Projekt analysiert habe, um darüber zu schreiben. Man braucht für diesen Beitrag zum Projekt keinerlei

[10] `https://github.com/ILIAS-eLearning/ILIAS/pull/5670`

Programmierkenntnisse, es reicht den Text mit der *GitHub-Benutzeroberfläche* zu editieren. Dazu benötigt man nur auf *GitHub* ein Benutzerkonto und etwas Experimentierfreude, neue Dinge auszuprobieren.

6.6 Umfragesoftware selbst entwickelt

Für meine Maturaarbeit wollte ich eine Umfrage zum Wissensstand zu und der Verbreitung von Freier Software in Schweizer Schulen durchführen. Dabei stellte sich heraus, dass praktisch das gesamte gängige Angebot zum Durchführen einer Umfrage *Freemium* Services sind, also in der Regel kostenlos nutzbar, aber unfreie Software.

Für ein modernes Umfrage-Service benötigt man einen Server, der ständig online ist, und eine Website oder App-Inhalte ausliefert. Das ist zwar keine Raketenwissenschaft, aber trotzdem nicht für jedermann so einfach umzusetzen. Deshalb gibt es viele solcher Services online.

6.6.1 Anforderungen

Um selbst ein Umfrageprogramm anzubieten, benötigt man z.B.

- Einen Computer (*Host*), auf dem ein Webserver laufen kann.
- (*Open Source*-) Software, die die Funktionalität von Umfragen anbietet.
- Das Programm wird eine Datenbank brauchen, um die Umfragen zu speichern.

Ich gehe davon aus, dass unsere Schule Serverinfrastruktur hat, um ein kleines Programm zu hosten. Die Software, wenn es keine fertige Lösung gibt, müsste man selbst entwickeln, dabei kann man aber, wenn möglich, schon bestehende Projekte wiederverwenden.

6.6.2 Recherche nach einer FLOSS-Lösung

Wir brauchen ein Programm, das am besten direkt einsetzbar ist, ohne dass wir selbst entwickeln müssen. Wenn es das nicht gibt, dann ein Programm in einer Programmiersprache, die wir schon verstehen oder leicht lernen können, und das fast schon alles kann, was wir brauchen. Damit wir fehlende oder nicht ganz passende Features leicht ergänzen oder anpassen können.

In der Schule haben wir die Programmiersprache *Python* kennengelernt und *Django* ist ein bekanntes *Framework* zum Entwickeln von Websites. Eine Suche im Web nach „python django survey open source“ erweist sich schon als hilfreich (Abbildung 6.6).

Der erste Link führt uns auf das Projekt *django-survey* von Matt Segal auf *GitHub*.[11] Das README erklärt das Programm gut, der erste Screenshot ist schon vielversprechend. Leider hat das *Source Code Repository* keine Lizenz, der Quellcode ist damit automatisch von Copyright geschützt, also unfrei. Wir können aber zumindest den Blogbeitrag lesen, den Matt im README verlinkt hat. Er beschreibt dort ausführlich die Entwicklung einer Survey-Applikation mit Django. Das hilft auch weiter. (Danke, Matt!)

Der zweite Link zeigt auf *django-survey-and-report*, eine „django survey app“, wie das README und der Projektname verraten.[12] Auch hier gibt es eine ausführliche Beschreibung, die

[11] `https://github.com/MattSegal/django-survey`
[12] `https://pypi.org/project/django-survey-and-report`

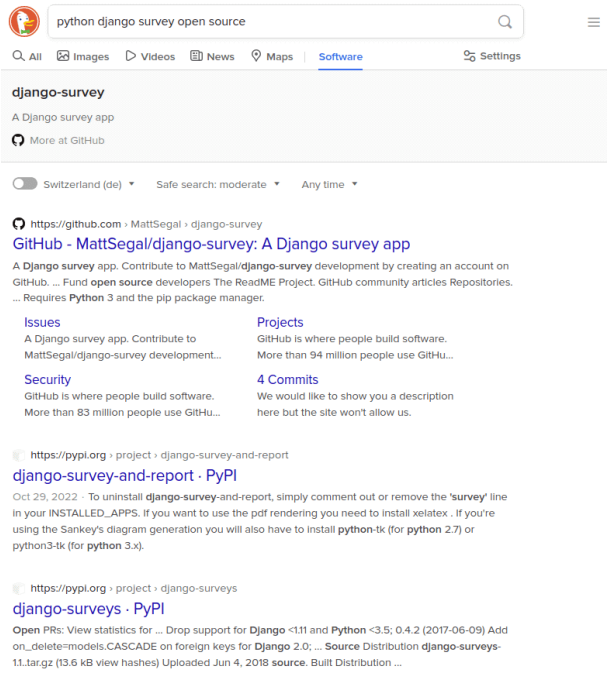

Abbildung 6.6: Recherche nach einer Umfrage-Applikation

Screenshots werden auf der Website aber leider nicht angezeigt. Über den „Homepage“-Link gelangt man aber zum *Source Code Repository*, wo die Bilder im README richtig dargestellt werden. Auch dieses Projekt sieht vielversprechend aus. Es steht unter der *GNU Affero General Public License v3 (AGPL-3.0)* Lizenz, d.h. wir dürfen den Quellcode frei verwenden.

6.6.3 Fazit

Wenn man sich etwas für Software interessiert, findet man im Internet für viele Problemstellungen Lösungen, die – Dank freier Lizenz – frei verwendbar sind.

Wie schwierig oder einfach die Inbetriebnahme einer solchen Software ist, muss sich noch zeigen. In unserem Fall ist, was zu tun ist, aber Schritt für Schritt beschrieben und wirkt einfach umzusetzen. Es sollte daher möglich sein, zumindest einen einfachen Prototypen rasch zum Laufen zu bringen.

Denkbar wäre, in einem Schüler:innen-Projekt, daraus eine vollwertige Lösung zu bauen, die man dann als *Open Source*-Lösung auch anderen Schulen zur Verfügung stellt.

Kapitel 7

Software im Einsatz an Schulen

Wie arbeiten wir heute an unseren Schulen? Wir verwenden vorwiegend Software von *Microsoft*. Das beginnt bei den privaten Laptops und Tablets, die jede:r Schüler:in zur Schule mitbringen muss. Darauf läuft in der Regel *Microsoft Office 365* (Word, PowerPoint, Excel) und für Schul E-Mails kommt das *Outlook* Webmail zum Einsetz. Als *Betriebssystem* fungiert meistens *Windows* oder *macOS*, auf den Tablets *iOS*.

Zur Unterstützung des Lehrbetriebs gibt es in der Regel Lernmanagementsoftware und *Microsoft SharePoint* zur Dokumentenablage und -verwaltung. Gemeinsam verwendete Software ist auf virtualisierten Servern verfügbar (z.B. *Citrix*). Die Schulwebsite wird oft extern gemanaged. Für Schülerprojekte (Schülerzeitung) verwenden die Schüler:innen häufig externe Services, wie z.B. wordpress.com.

Es gibt aber auch Initiativen für mehr Freiheit und Nachhaltigkeit an einzelnen Schulen. In diesem Kapitel sehen wir uns an, wie unterschiedliche Schulen Software einsetzen.

7.1 Pädagogische Maturitätsschule Kreuzlingen

Gemäss eigenen Recherchen hat die PMS Kreuzlingen folgende Software im Einsatz.

- *ILIAS*, *Open Source* Learning Management System
- Klassenbuch (Stundenplan, Absenzen, Hausaufgaben, Prüfungen), Eigenentwicklung
- Intranet, *Microsoft SharePoint*
- *Microsoft Office 365* (Word, PowerPoint, Excel)
- *Microsoft Outlook* (Webmail)
- *Citrix*, *Windows* Software auf Terminal Server
- *TigerJython* (Lernsoftware für den Informatikunterricht)
- Schulhomepage, extern verwaltet (Kontrolle: Kanton, Technologie: *GOViS* CMS)

Grösstenteils wird damit nicht nur *proprietäre Software*, sondern vor allem solche von *Microsoft*, eingesetzt. Für Lehrpersonen, Schüler:innen und Mitarbeiter:innen der Schule besteht die Möglichkeit, *Microsoft Office 365* „kostenlos“ am privaten PC für schulische Zwecke zu nutzen (Abbildung 7.1).

Abbildung 7.1: Alles gratis! Microsoft ist spendabel

7.2 Schulen in Grossbritannien

In Grossbritannien, berichtet ein Blogbeitrag auf KDE.org, hat die Regierung im Jahr 2012 den alten Informatik-Lehrplan über Bord geworfen, der grösstenteils das Schulen von *Microsoft Office 365*-Produkten vorgesehen hatte. Das hat motivierten Schulen erlaubt, eigene Wege zu gehen und Ersatz für *Microsoft* Produkte zu suchen, die eine nachhaltigere Vermittlung von Informatikthemen erlauben.

> *The motivation was initially both cost and philosophical, in that, even in an outstanding school, funds are always going to be limited (politicians don't seem to get the saying 'If you think education is expensive, try ignorance'). The cost of using Windows is high but not always obvious, Windows carries a lot of baggage that bumps the cost up considerably over a Linux environment. The philosophical angle was probably the philosophy of pragmatism. We wanted to offer the best IT systems and education possible with the funds available. Money spent on essentially promoting Microsoft Windows and Office to students can be better spent on old-fashioned things like teachers and actual education. Subsequently and fortuitously, the UK government threw out the old ICT syllabus, which was based largely on teaching students how to use Microsoft Office, and told schools to go for a more computer studies-based syllabus, which meant that we were in a position to hit the ground running so to speak.*[1]

7.2.1 Linux wird als Investition in die Zukunft gesehen

Sogar der Widerstand gegen die Umstellung der IT-Landschaft, hielt sich in Grenzen, wie der Verantwortliche für den IT-Support an der Westcliff High School zu Protokoll gibt.

> *Surprisingly, very little* [resistance]. *The Senior Leadership Team grilled me in two long meetings which was fun! Once you actually take a step back from the misconception that computers = Windows and actually seriously think about it, the pros clearly outweigh the cons. The world is changing very quickly. There is a survey that reports in 2000, 97% of computing devices had Windows installed, but now with tablets and phones, etc., Windows is only on 20% of computing devices, and in the world of big iron, Linux reigns supreme. We specialize in science and engineering and want our*

[1] Stu (KDE), *A Year of the Linux Desktop*.

students to go on to do great things like start the next Google or collapse the universe at CERN. In those environments, they will certainly need to know Linux.

Laut dem aktuellen Stand der Website der Westcliff High School ist das beschriebene System weiterhin so im Einsatz.[2] Das Angebot der Freien Software wurde allerdings in den letzten Jahren offenbar durch Cloud-basierte Angebote von *Microsoft* und *Google* ergänzt.

7.3 Freie Software am Georg-Büchner-Gymnasium (Hannover)

In einem Interview mit Netzpolitik.org aus dem Juni 2020 gibt Felix Schoppe, Lehrperson für Physik, Französisch und Informatik, Einblick in den Einsatz von Freier Software am Georg-Büchner-Gymnasium in Seelze bei Hannover.[3]

Schoppe kümmert sich um 3 Computerräume, mehrere Einzelrechner und ca. 150 Laptop-PCs für 1'400 Schüler:innen. Sein Gymnasium setzt neben Freier Software auch auf Gebraucht-Hardware, um Umweltschutz nachhaltig zu leben. Es gibt dazu einen breiten Konsens, sowohl beim Lehrkörper als auch bei der Schülervertretung. Das habe die Umstellung von *Windows* auf *Linux* einfach gemacht und es gab keinen nennenswerten Widerstand.

7.3.1 Schüler:innen Arbeitsgruppe für Hard- und Software

Schüler:innen engagieren sich in Form einer Arbeitsgruppe darum, alte Notebooks mit Linux auszustatten und die Hardware, wenn kaputt, sogar zu reparieren. Die Schule hat auch beschlossen, Hardware, die nicht selbst repariert werden kann, nicht mehr zu kaufen, sagt Schoppe im Interview. Diese Werte den Schüler:innen zu vermitteln, erachtet der Lehrer als wichtig.

> [...] *man muss sich auch fragen: Was für Werte stecken dahinter und was wollen wir den Schüler:innen vermitteln? Da finde ich es gerade in Zeiten des Klimawandels extrem wichtig zu zeigen: Man kann auch anders agieren. Man kann sich Geräte anschaffen, die man reparieren kann, die man sehr lange nutzen kann.*

Bei der Software steht für Schoppe das Erleben der Gemeinschaft im Vordergrund.

> *Bei Linux ist es auch so, dass* Freie Software *allgemein ein Gemeinschaftsprojekt ist. Das habe ich vielfach selbst erlebt.* [...] *wenn es Probleme gibt, dann wird das in der Entwicklungs-Community diskutiert und man arbeitet über Grenzen hinweg zusammen.* [...] *So entsteht eine enge Zusammenarbeit, obwohl man sich nie persönlich irgendwo gesehen hat. Das ist etwas, das man mit iPads nicht vermittelt bekommt.*

7.3.2 Dienstleister für IT-Support und Kompromisse mit Apps

Zum Betrieb der IT, erklärt Schoppe, hat die Schule zwei IT-Administratoren, die sich aber hauptsächlich um das Netzwerk inkl. Wi-Fi kümmern. Für die Schulserver habe die Schule einen Dienstleister, der die ständige Verfügbarkeit sicherstellt. Um die Endgeräte kümmere die Schule sich aber selbst – soll heissen, die Arbeitsgruppe und er als Lehrer.

Dass es ab und zu Applikationen gibt, die nicht auf den selbst verwalteten Rechnern laufen, z.B. „iPad-App[s]“, gehöre zum Lernprozess. Wenn man eine kompetente Informatik-Fachgruppe hat, helfe das aber, Lösungen auch für solche Probleme zu finden.

[2] Vgl. Westcliff High Scool for Girls, *IT & Software*.
[3] Vgl. Barthel, *Freie Software in der Schule: Mit Linux-Rechnern zur digitalen Nachhaltigkeit*.

7.4 Free Upgrade for a digital Sustainable School (FUSS)

FUSS ist ein Projekt, das im Jahr 2005 gegründet wurde, um *Freie Software* an die Schulen des Südtirolischen Bozen zu bringen. Es ist ein Nachhaltigkeitsprojekt für Schulen und gleichzeitig ein *Freie Software*-Produkt, um ein 360-Grad-Management eines Schulnetzwerks auf Basis von Freier Software zu ermöglichen. Das bedeutet, es wird auf Kompromisse verzichtet und ganz bewusst versucht, alle Bedürfnisse des Schulbetriebs mit Freier Software abzudecken. Das ist bis dato einzigartig in Europa.

Die Initialzündung fand 2004 statt, als es der lokalen Linux User Group Bozen gelang, sich mit politischen Entscheidern darauf zu verständigen, dass für gewisse Punkte im Aktionsplan zur Entwicklung der Informationsgesellschaft in der Provinz Bozen, der Einsatz von Freier Software insbesondere in Schulen, der öffentlichen Verwaltung und zur Innovationsförderung bei lokalen Unternehmen notwendig sei.[4]

Abbildung 7.2: Mission des *FUSS*-Projekts

7.4.1 Gleiche Möglichkeiten auch zu Hause

Grundmotivation hinter dem Projekt FUSS war die Idee, dass Lehrpersonen und Schüler:innen zu Hause die gleiche Software verwenden können sollen wie in der Schule, ohne dass eine digitale Kluft entsteht, denn nicht jede Familie könne sich jährlich Software-Lizenzen leisten, erklärt ein Projektverantwortlicher in einem Online-Meeting im März 2023.[5]

Aus heutiger Sicht und unabhängig vom FUSS-Projekt hat sich diese Problematik etwas relativiert. Software wird heute vielfach in der Cloud zur Verwendung angeboten. Nach *Google* hat auch *Microsoft* begonnen, sein Angebot über einen Online-Account zugänglich zu machen. Schulen bekommen Lizenzen angeboten, die es erlauben, die Software auch zu Hause zu nutzen (vgl. Abbildung 7.1).

Das hat dazu geführt, dass es eine leichte Möglichkeit gibt, diese digitale Kluft zu überwinden. Und sogar an Schulen mit Initiativen für *Freie Software* wurde dieses Angebot genutzt, wie unser Beispiel aus Grossbritannien zeigt (Abschnitt 7.2). Dass dies nur über Finanzierung von Steuergeldern möglich ist, ist aber nicht jedem bewusst. FUSS hat also weiterhin Bedeutung.

[4] Vgl. FUSS, *Incontro Emilia Romagna (23.03.2023 mattina)*, 2. und 3. Folie.
[5] Vgl. ebd., 3. Folie.

7.4.2 Kennzahlen

Paolo Dongilli, Koordinator des FUSS-Projekts präsentierte in einem Online-Meeting folgende Zahlen zum Projekt. Diese beziehen sich auf öffentliche, italienischsprachige Schulen in der Provinz Bozen.[6]

- 17'000 Studenten
- 1'900 Lehrpersonen
- 74 Schuleinrichtungen
- 4'500 PCs und Notebooks
- 64 Server
- ca. 200 didaktische Applikationen
- 3 Projekt-Koordinatoren
- 8 IT-Supporter (6 davon Vollzeitangestellte)
- ca. 74 Lehrpersonen als „IT-Champions“ in Schuleinrichtungen

[6] Vgl. FUSS, *Incontro Emilia Romagna (23.03.2023 mattina)*, 13. und 14. Folie.

Kapitel 8

Umfrage an Schweizer Schulen

Um den aktuellen Stand zu Kenntnis und Einsatz von FLOSS an Schweizer Gymnasien besser einschätzen zu können, habe ich Mitte März 2023 eine Umfrage an 86 Schulen geschickt.

Die Umfrage wurde aus Zeitgründen mit *Google Forms* umgesetzt. Eine freie Alternative wurde recherchiert (siehe Abschnitt 6.6).

8.1 Kontaktaufnahme

Umfrage Maturaarbeit „Freie Software, freie Gesellschaft“

Sehr geehrte Rektor:innen

Mein Name ist Fabio Bittner. Ich bin Schüler an der Pädagogischen Maturitätsschule Kreuzlingen und beschäftige mich im Rahmen meiner Maturaarbeit mit dem Thema „Freie Software, freie Gesellschaft“.

Für ein besseres Verständnis der aktuellen Situation im Einsatz von Software an weiterführenden Schulen benötige ich Informationen aus erster Hand. Dazu habe ich eine Umfrage zusammengestellt, die ich an ausgewählte Schulen in den Kantonen der Deutschschweiz und Graubünden sende.

Haben Sie ein paar Minuten Zeit, um mir dabei zu helfen? Die Beantwortung der Fragen dauert ca. 7 Minuten. Ich wäre Ihnen sehr dankbar, wenn Sie mir diese Umfrage bis Montag, 03.04.2023 beantworten könnten.

`https://forms.gle/ShB1buSN3vx3ANKt6`

Als Dankeschön stelle ich Ihnen auf Wunsch ein physisches Exemplar meiner Maturaarbeit für die Schulbibliothek zur Verfügung. Für Ihre wertvolle Unterstützung bedanke ich mich schon jetzt recht herzlich.

Wenn Sie Fragen zur Umfrage haben, erreichen Sie mich gerne unter Telefon 076 725 **** oder antworten Sie auf diese E-Mail.

Freundliche Grüsse
Fabio Bittner

Von dieser Umfrage sind 26 Antworten eingegangen. Im Folgenden sind die Fragen und Auswertung der Antworten angeführt.

❝ *Ich wäre froh, wenn offene Software zentral vom Kanton evaluiert und auch unterstützt würde, anstelle sich Microsoft hinzugeben - und dies auch vorzugeben.*

Abbildung 8.1: Abschluss-Statement einer Teilnehmer:in

8.2 Die Umfrage

8.2.1 Einleitung

Vielen Dank, dass Sie sich Zeit nehmen, diesen Fragebogen zu beantworten!

Mein Name ist Fabio Bittner. Ich bin Schüler an der Pädagogischen Maturitätsschule Kreuzlingen und beschäftige mich im Rahmen meiner Maturaarbeit mit dem Thema „Freie Software, freie Gesellschaft“.

Für ein besseres Verständnis der aktuellen Situation im Einsatz von Software an weiterführenden Schulen benötige ich Informationen aus erster Hand. Diese Umfrage geht an ausgewählte Schulen in den Kantonen der Deutschschweiz und Graubünden.

Hintergrundinformationen: Freie Software (Video) - `https://fsfe.org/freesoftware`

Name, Institution, Kontaktdaten

- Arno Germann, Kantonsschule Im Lee Winterthur
- Tobias Weber, Realgymnasium Rämibühl Zürich
- Chris Nielsen, Kantonsschule Hohe Promenade Zürich
- Gymnasium GR
- Brigitte Pallmann, Pädagogische Maturitätsschule Kreuzlingen
- Kantonsschule Frauenfeld
- Aleksandar Popov, KZO
- Martin Huber, Gymnasium Kloster Disentis
- Ruedi Herzog, KS Romanshorn
- Lea Castiglioni, Kantonsschule Freudenberg
- Marcello Indino, Kantonsschule Kreuzlingen
- Thomas Rätz, Gymnasium Liestal
- Andreas Gräub, Campus Muristalden
- Matthias Küng, Gymnasium Neufeld
- PHTG
- Gymnasium Oberaargau
- Giuliani Debora, KRW
- Gymnasium Kirchenfeld
- Martin Klee, KME, Mühlebachstrasse 112, 8008 Zürich
- Roland Lüthi, Kantonsschule Zürcher Unterland
- Niklaus Streit, Gymnasium Hofwil
- Raphael Barengo, KUEZH
- Lukas Strub, Gymnasium Unterstrass

- KRW
- Jürgen Mischke, FG Basel
- Daniel Burri, Neue Schule Zürich

Welche dieser Begriffe sind Ihnen bekannt?

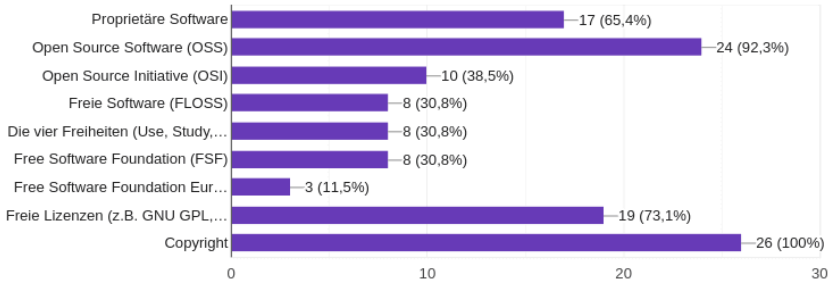

Abbildung 8.2: Frage 2, Bekannte Begriffe

8.2.2 Freie Software und kollaborative Prozesse

Kommt in Ihrer Bildungsinstitution Freie Software zum Einsatz? Software, die Sie für beliebige Zwecke nutzen dürfen. (Freedom to use)

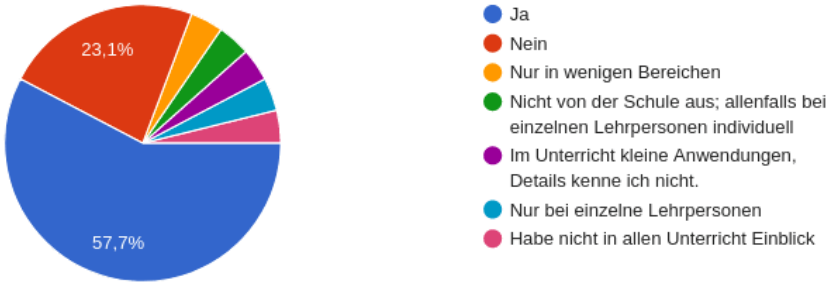

Abbildung 8.3: Frage 3, Freedom to use

Wenn ja, welche?

- Linux
- Steuerungssoftware im Labor (Logger Pro), evt. auch im EF Informatik, da sehe ich zu wenig hinein
- Bildbearbeitung (GIMP, paint.net), Online-Speicher (balloon)
- Moodle, Zammad, Guacamole
- z.B. Geogebra, Mozilla Firefox, ...
- gimp, netbeans, tiger-jython, virtualbox, blender,
- Tunnelblick
- LibreOfficer, Gimp, Linux
- sehr viele
- Webserver, MailClients
- Smartlearn

- Linux auf allen Servern der Basisinfrastruktur (DNS, DHCP, NTP, PXE, VPN, Radius, ACME, SSH, Firewall), Linux auf weiteren Servern (diverse Webseiten, Git, Wetterstation mit weewx), Linux auf den Storage-Servern (Ceph), Linux auf allen Virtualisierungsservern (Linux Container, Linux KVM), Windows-Client-Deployment mit Ansible, Client-Software (Firefox, VLC, Audacity, Geogebra, GIMP, Inkscape, Thonny, Filius) und z.T. Linux als Client-OS
- Raspberry Pi OS (Linux), Thonny (nicht 100% frei), Moodle, Latex
- Firefox
- z.B. Python
- Notepad++, Audacity, Gimp, Blender, Krita, LightZone, OpenShot, Pivot Animator, Dia, GEONext, Z. u. L., GraphCalc, VLC, MuseScore, Stellarium

Wenn Freie Software genutzt wird, wird diese verwendet um ihre Funktionsweise, ihren Quellcode zu studieren? (Freedom to study)

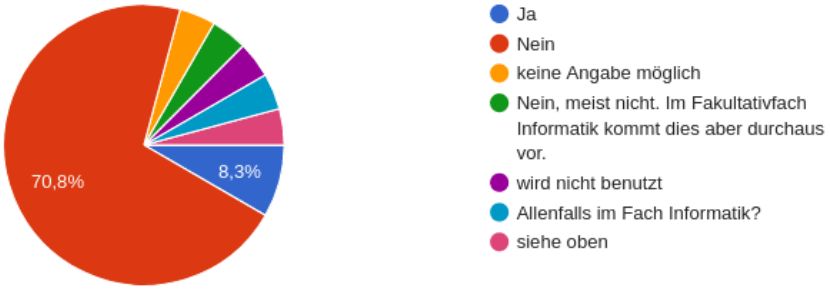

Abbildung 8.4: Frage 4, Freedom to study

Wenn ja, geben Sie Beispiele.

- opensource-Projekte
- Im Rahmen des Informatikunterrichts oder in entsprechenden Freikursen.
- Im Fakultativfach Informatik arbeiten wir viel mit Linux, wir schauen z.B. Debian-Pakete an. In den Themenbereichen Reverse-Engineering und Hacking analysieren wir durchaus fremden Code. Wir verwenden auch Git und klonen somit auch OSS-Repos.
- s. oben

Wenn Freie Software genutzt wird, haben Schüler:innen die Möglichkeit, sie (bei Bedarf auch den Quellcode) auf ihre Wünsche und Bedürfnisse anzupassen? (Freedom to improve)

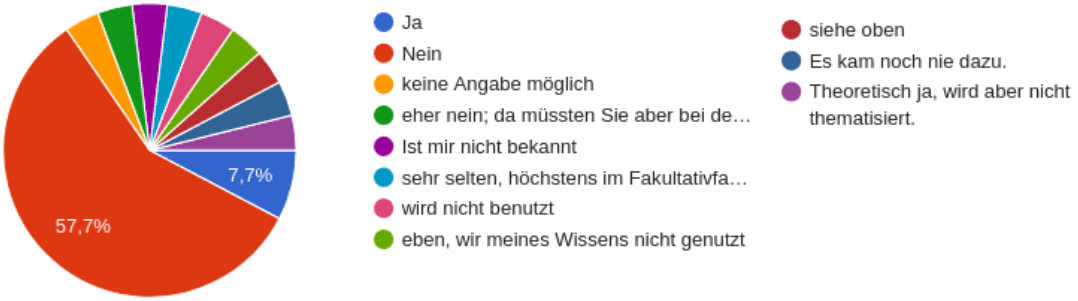

Abbildung 8.5: Frage 5, Freedom to improve

Wenn ja, geben Sie Beispiele.

- Die von der Schule angebotenen Programme können nicht direkt angepasst werden. Aber die SCH haben eigene Geräte und dort die volle Freiheit, opensource-Programme nach eigenem Gutdünken anzupassen.
- Im Umfeld von Projekten der Informatikmittelschule an der Kanti FF
- siehe oben
- Wir verwenden im Fakultativfach Informatik Git und verwenden auch fremden Code, den die Schüler*innen somit (zumindest lokal) auch anpassen könnten.

Wenn Freie Software genutzt und verbessert wird, haben die Schüler:innen die Möglichkeit den Quellcode frei mit anderen zu teilen? (Freedom to share)

Abbildung 8.6: Frage 6, Freedom to share

Wenn ja, geben Sie Beispiele.

- siehe letzte Frage.
- Im Umfeld von Projekten der Informatikmittelschule an der Kanti FF
- siehe oben
- Da die Schüler*innen Git-Repos auf unserer eigenen Gitlab-Plattform beantragen dürfen, können sie so ihre eigenen Projekte oder geteilte Projekte auch zusammen entwickeln und mit anderen teilen.

Haben Schüler:innen die Möglichkeit am Wartungs-, Anpassungs- und Verbesserungsprozess der in Einsatz befindlichen Software zu partizipieren? (z.B. Bug Reports, Feature Requests)

Abbildung 8.7: Frage 7, Bug Reports, Feature Requests

Wenn ja, geben Sie Beispiele.

- Z.B. im Moment läuft eine grosse IT-Umstellung in O365 (zugegeben ein sehr schlechtes Bsp. für freie Software): Da sind auch SuS in die Testphase eingebunden.
- Im Umfeld von Projekten der Informatikmittelschule an der Kanti FF
- siehe oben
- Klassenbuch (Intern-Webseite), hier erhalten wir viel Feedback von Schüler*innen

8.2.3 Software und Allgemeinbildung

Werden an ihrer Schule neutrale Technologiebegriffe verwendet? (Konzeptbegriffe statt Produktnamen, Beispiel: Textverarbeitung statt Word)

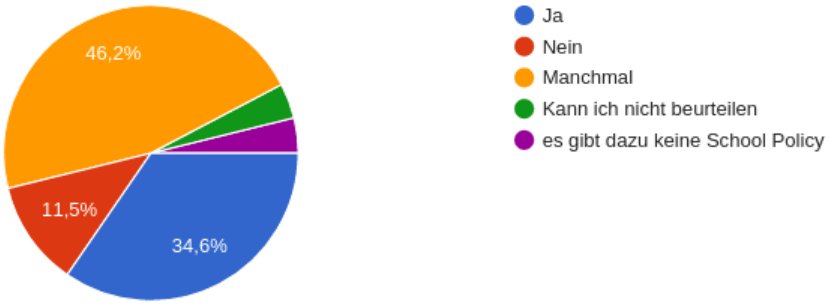

Abbildung 8.8: Frage 8, Neutrale Technologiebegriffe

Wird das Thema „Software“ an der Schule auch ausserhalb des Informatikunterrichts mit allen Schüler:innen behandelt? (z.B. Praktika, politische Bildung, Wirtschaft und Recht)

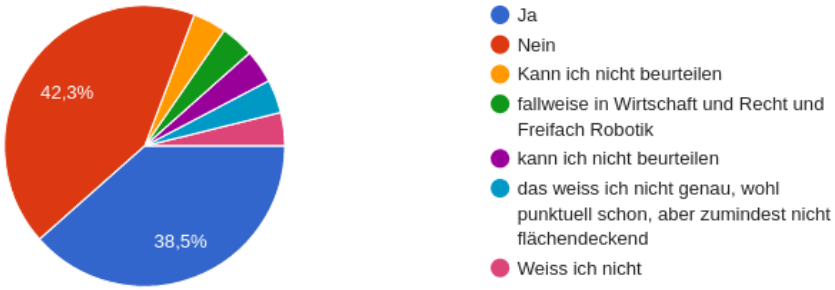

Abbildung 8.9: Frage 9, Ist Software Thema im Unterricht

Bitte erklären Sie, wenn nötig.

- Behandelt wird es ausserhalb des Unterrichts zwar nicht explizit - Die Hard- und Softwarefreiheit wird aber zumindest durch die freie Wahl für ein eigenes Gerät gefördert. Wir haben durchaus auch SCH, die sehr bewusst so konsequent wie möglich mit OpenSource-Produkten arbeiten.
- In Projektwochen zur Programmierung sowie Video-/Bildbearbeitung.
- Im Rahmen von Sonderwochen.
- In einer Reihe „interdisziplinäre Projekte“, in Wirtschaft und Recht, in Studienwochen

Haben Schüler:innen nach dem Abschluss der Schule (z.B. Matura) einen Überblick über Möglichkeiten (Einsatz und Nutzung) von Freier Software und proprietärer Software? (Auftrag der Allgemeinbildung erfüllt?)

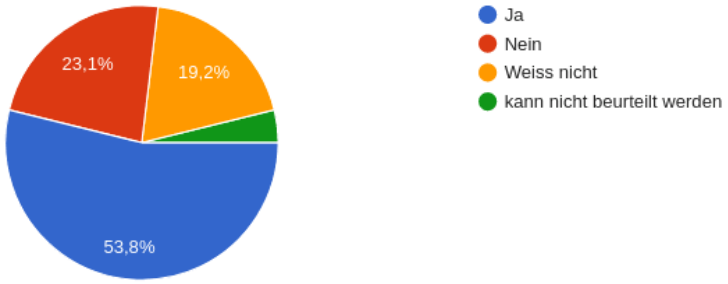

Abbildung 8.10: Frage 10, Auftrag der Allgemeinbildung

Bitte erklären Sie, wenn nötig.

- Letztlich sind wir auch Zwangskunden des Kantons, dies erschwert auch die Bewusstseinsbildung bei den Lehrpersonen. Und wenn die LP nicht mitziehen, dann erreichen wir die SuS nicht. In meiner vorherigen Schule (KS Kreuzlingen) hatten wir vor ca. 15 Jahren den Versuch gestarten, komplett auf freie Software zu setzen, sind aber am geballten Widerstand von LP und SuS gescheitert.
- Das ist explizit Thema im Informatikunterricht.
- Obligatorisches Fach Informatik am GYM stellt dies sicher.
- Die Begriff „Auftrag der Allgemeinbildung erfüllt“ irritiert mich ;) Ich habe mich gerade gefragt, wo wir dies in unseren Lehrplänen haben. Das ist noch nicht konkret verankert bei uns.
- Das Thema wird im Rahmen des obligatorischen Informatikunterrichts behandelt.
- Da Informatik inzwischen am Gymnasium obligatorisch ist, und wir im Informatikunterricht über freie Software sprechen, sind die Grundgedanken bekannt.
- Diese Fragen werden im Informatikunterricht thematisiert.
- Dies wird neu im Fach Informatik thematisiert

Haben Sie Kenntnis davon, dass Absolventen Ihrer Institution später Start-ups gegründet haben oder als Entscheidungsträger (für den Einsatz von IT-Systemen) tätig sind?

Abbildung 8.11: Frage 11, Werden Schüler Entscheidungsträger

Bitte erklären Sie, wenn nötig.

- diverse IMS Absolventen

8.2.4 Einführung, Wartung und Ausgaben für Software

Haben Sie in der Vergangenheit schon Versuche unternommen, Freie Software an Ihrer Institution einzuführen, die gescheitert sind? Warum sind diese gescheitert?

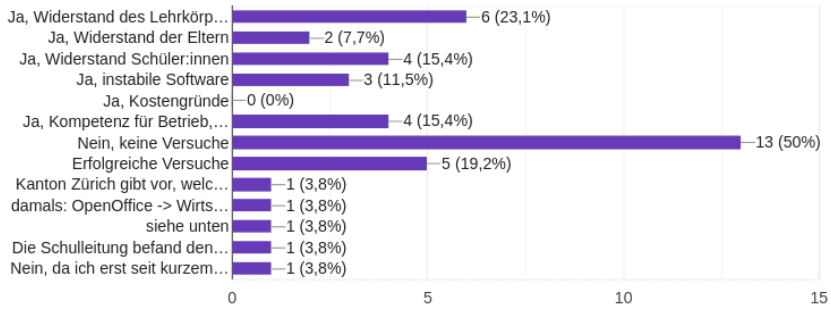

Abbildung 8.12: Frage 12, Freie Software einführen

Bitte erklären Sie, wenn nötig.

- Siehe vorher: Wir hatten im Serverbereich auf Linux gesetzt (für den Enduser trauten wir uns nicht) und in den Office-Produkten auf OpenOffice. Für die Bildbearbeitung GIMP. Das Problem war: Die Akzeptanz. Es kamen immer wieder die 2 Argumente: Später am Arbeitsplatz wird auch mit den kommerziellen Produkten gearbeitet und zu Hause haben wir auch dies.
- vgl. oben
- Gerade bei Einführung der BYODs haben wir an der Schule viel über die verschiedenen Möglichkeiten von Softwareeinsatz gesprochen und wollten dabei möglichst freie Wahl lassen. SCH und LP, die pro „proprietäre Software“ waren, begründeten ihre Haltung meistens mit Performance- und Kompatibilitätsproblemen der OpenSource-Produkte. Es war immer nur ein kleiner Prozentsatz an SCH und LP, die sich für Open-Source-Produkte interessierten und sich auch an diese heranwagten (meistens handelte es sich um „computerversierte“ User, die sich für OpenSource entschieden haben). Für die IT-Abteilung bedeutet die Verwendung von OpenSource-Produkten in der Tat auch viel erforderliches Knowhow, das nicht immer per se vorhanden ist.
- Konkret nicht unternommen, aber Gespräche dazu geführt. Da besteht eine grosse Angst vor dem Neuen und das Problem, dass viele Abhängigkeiten (z.B. schon vorbereitetes Material in proprietären Formaten) bestehen.
- Die Einführung freier Software ist Teil unserer IT-Strategie und befindet sich seit zwei Jahren in Arbeit. Die Lehrpersonen werden aber nicht gezwungen sein diese Software zu nutzen (etwa als Ersatz zu Windows-Lösungen), sondern es werden ihnen lediglich Alternativen vorgestellt und angeboten. Jede und jeder kann für sich selbst entscheiden, welche Produkte gewählt werden.
- Open Office ... Kompatibilität und Handhabung aber nicht akzeptiert worden
- Wir haben fast keine Windows-Server mehr. Das allermeiste realisieren wir mit Linux-Servern.

  Wir haben mehrere Jahre lang (nach der Kostenexplosion bei Adobe) im bildnerischen Gestalten nur OSS verwendet (GIMP, Scribus, Inkscape). Allerdings

haben sich - trotz intensiver Schulung - viele Lehrpersonen damit nicht wohl gefühlt und eine „einheitliche“ Darstellung und eine Integration dieser verschiedenen Produkte vermisst. Die Fachschaft hat dann beschlossen, Affinity-Produkte einzusetzen.

Wir haben früher (vor der Einführung des obligatorischen Fachs Informatik) im ICT-Unterricht und in den entsprechenden Unterrichtsunterlagen bei Textverarbeitung/Tabellenkalkulation/Präsentation sowohl MS Office wie auch LibreOffice berücksichtigt. Mit der Einführung von Office 365 und der Cloud, ist LibreOffice leider in den Hintergrund gerückt.

- Wie möchten gerne Versuche starten. Der Kanton wird kaum erfreut sein, da dieser Produkte der Microsoft Plattform pusht.

Von wem werden die Ausgaben für IT-Systeme an Ihrer Institution gezahlt? (Tools, Lizenzen, Wartungen, etc.)

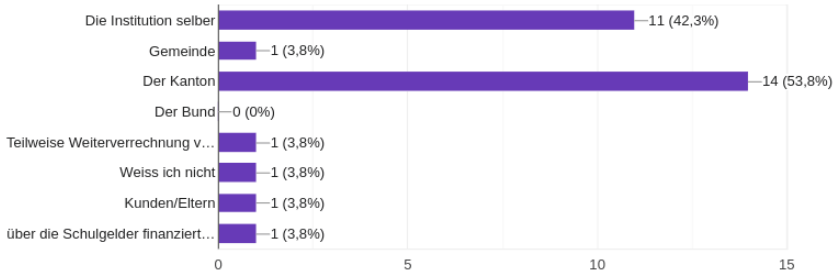

Abbildung 8.13: Frage 13, Wer zahlt die IT-Ausgaben

Welche Ausgaben hat Ihre Institution jährlich für IT-Systeme? (Tools, Lizenzen, Wartungen, etc., ohne Personalkosten)

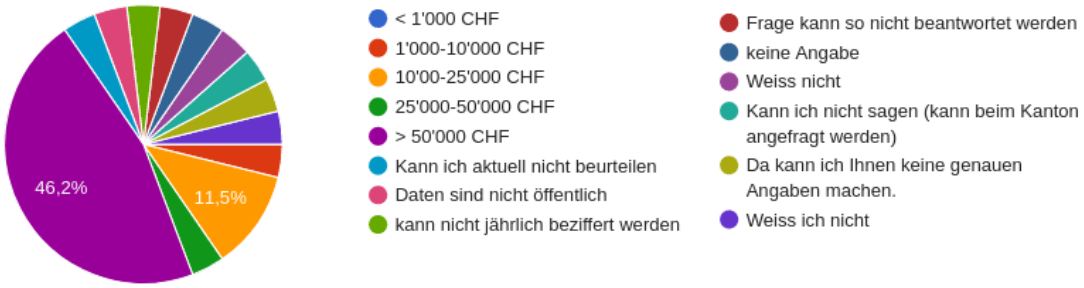

Abbildung 8.14: Frage 14, Summe der IT-Ausgaben

Bitte schlüsseln Sie auf oder erklären Sie (für beide obigen Fragen).

- Wir bezahlen eine Pauschale an den Kanton für die Netzwerkverbindung und für O365. Zusätzlich zahlen wir für Adobe (Photoshop)
- Die IT-Kosten gehen direkt auf das Budget der Institution. Eine Aufschlüsselung ist nicht immer eindeutig, weil wir Synergien nutzen auf dem Campus und viele Programme campusweit genutzt - und somit auch bezahlt - werden.
- Die Schule ist ja keine eigenständige Institution und ein Teil der Kosten ist durch die Eltern zu tragen.
- Die PHTG hat ein spezifisches Budget Pro Jahr für Software und kann dieses selbstständig auch verwalten

- Lizenzkosten beinhalten vor allem Microsoft (knapp 20'000.-), Netbiblio (3000.-), Affinity (3000.-), Untis (500.-) plus einige Zertifikate und andere kleine Lizenzen

  Schnelle Internetverbindung und Netzwerkverbindungen zwischen den Schulstandorten (knapp 20'000.-)

  Hardwarekosten je nach Jahr und Anschaffungsbedarf resp. Lifecycle (Server, Switches, Access Points, USV, Clients, Spezialhardware wie Wetterstation, Hausleitsystem).
- Kann ich nicht sagen (kann beim Kanton angefragt werden)

Wird Installation und Wartung (z.B. Updates) ihrer Software von regional ansässigen Unternehmen (oder ihnen selbst) durchgeführt?

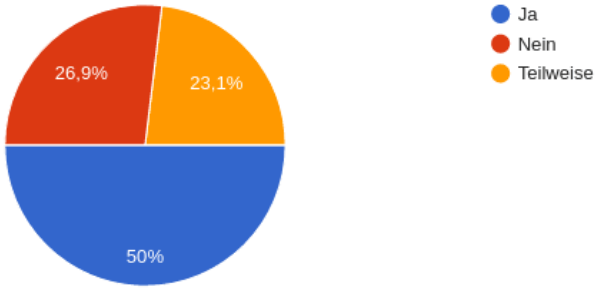

Abbildung 8.15: Frage 15, Regionale Support-Partner

Bitte erklären Sie zum besseren Verständnis.

- Entweder durch uns selbst oder durch den Kanton
- Die Aufgaben sind verteilt, zum Teil werden Wartungen von uns selbst oder von lokalen Unternehmen durchgeführt.
- Sourcing-Provider
- IT Abteilung inhouse
- Je nach Komplexitätsgrad in-house oder extern
- Regional ansässiges Unternehmen
- eigene IT-Abteilung
- Updates werden von uns selbst durchgeführt.
- Wir lösen möglichst viel selbst, verfügen aber auch über Wartungsverträge.
- Wartungsarbeiten, worunter auch Updates /Upgrades fallen werden selbstständig durchgeführt
- Support durch kantonale Stelle
- Einige Software-Updates werden intern gemacht. Andere werden von externen Firmen durchgeführt.
- Wir machen fast alles selber (Server, Netzwerk, Deployment, Beamermontage, ...).

  Benutzer- und Kurs-Team-Sync in die MS-Cloud macht eine Schweizer Firma.

  Drucksystem im Moment Canon Schweiz.

  Prüfungssystem eine Berner Firma. Das Prüfungssystem wir dieses Jahr in einen Verein überführt und als Open Source Software freigegeben (smartlearn).

- IT-Support der Schule und des Amtes
- BYOD, SuS sind selbst für Updates verantwortlich
- Hauseigene IT mit Mandat einer externen IT-Firma
- Wartung wird intern geleistet.
- sofern Schaffhausen von Zürich aus auch noch als „regional“ gilt.

Welche Tools stehen an Ihrer Institution für den Schulbetrieb zur Verfügung? Mit welcher Software bzw. Technologie sind sie realisiert?

- Moodle und O365 (OneNote) als Lernplattformen
  Klassenbuch und Intranet (Eigenentwicklung des Kantons Zürich)
  Website über eine Firma
- Klassenbuch (intern vom Kanton entwickelt), MS 365
- Microsoft Office 365, ProPlan, Intranet
- Moodle, Escola, Office inkl. Teams, One Note, Exchange, diverse Lernsoftware
- ILIAS, Klassenbuch, Intranet, MO365, Webpage.
- Webuntis, Intranet und Lernportal mit Sharepoint, Unterricht mit Moodle etc.
- Stundenplan, Absenzen, Klassenbuch Hausaufgaben, Prüfungen: Kanton
  Intranet: eigene Entwicklung
  Unterricht: Office Software, Microsoft Office 365, Microsoft SharePoint
  Schulhomepage: eigene Entwicklung, extern verwaltet (Kanton)
- Für Kommunikation & Unterricht: Microsoft Office 365 & SharePoint; Nur für Verwaltung (Dossiers, Absenzen, Noten): Schulsoftware eScola
- ISY (intern entwickelt, für Stundenplan, Schulkalender, interne Kommunikation etc.)
  MS Office 365
  Schulhomepage (via Kanton, Inhalte und Detailgestaltung aber in-house)
- Intranet (extern verwaltet, Kanton), O365, Homepage (externe Firma)
- Ilias, WebUntis (Stundenplan, Klassenbuch, Absenzenwesen), Intra- und Internet, MS 365, ecoDMS, BBB, sowie die bereits genannten Open Source-Produkte
- SharePoint, Office 365, Schulnetz
- M365, Odoo, Schulwebsite
- ILIAS, Klassenbuch, Intranet, Office 365 etc.
- Ilias, Sharepoint, M365, Homepage Typo
- Smartlearn. Entwickelt von einer externen Firma. Bezahlt durch Kanton. Software ist Open Source.
- Intranet, Microsoft SharePoint, Office Software, Microsoft Office 365, Schulhomepage, extern verwaltet (Kanton), Adobe Collection, Webuntis, Thonny, TigerJyhton
- Klassenbuch (Stundenplan, Absenzen, Hausaufgaben/Prüfungen, Raumreservationen, Anmeldungen für Kursangebote, Erfassung Qualitätsentwicklung, Maturarbeitsverwaltung, Portfolios), intern entwickelt (Java und übliche Webtechnologien, inkl. JSON-API)

Schulverwaltung (Abgleich von Evento in unsere DB, Zuweisung von Stellvertretern zu Kursen, Onboarding neue SuS, diverse Berichte, Steuerung von Kursangeboten und Einschreibeperioden, Verwaltung Terminliste, ...), intern entwickelt (Java und übliche Webtechnologien, inkl. JSON-API)

Pensenverwaltung, intern entwickelt (Java und übliche Webtechnologien, inkl. JSON-API), an mehreren Schulen in Betrieb

Schulwebseite (Wordpress selbst gehostet)

- Office 365, Schulhomepage, Moodle, Microsoft Share Point, WebUntis, Classtime
- Adobe und MS Office Packages; Intranet (Kantonslösung); Schulverwaltungssoftware (Kantonslösung)
- Office 365, Google for Education, Evento, Homepage, Intranet, schuleigene Server
- Microsoft Office 365, Teams, Intranetlösung (inkl. Klassenbuch, Absenzenwesen (noch) von gyselroth, Moodle (wird private gehostet, Kanton bietet aber eine Paket an)
- SchulNetz (centerboard) für Stundenplan, Absenzen, Prüfungsplan, Raumbuchungen, Terminplanung, Noten etc.

  Intranet als office365-SharePoint

  Office365-Paket für den Unterricht

  Selbstverwaltete Homepage
- Intranet, Microsoft SharePoint

  Office Software, Microsoft Office 365
- schulNetz, MS SharePoint, MS Office 365, WebUntis
- Office 365 mit TEAMS, interner Bereich der eigenen Webseite (selber verwaltet, extern gehostet), kommerzielle Lösung für die Schulverwaltung im LAN, kommerzielle Online-Lösung für Absenzen etc., Online-Dienste der Schulbuchverlage usw., lokal installierte Software auf den Schülergeräten (1:1-Ausrüstung, ausser Microsoft Office sind die meisten Programme die oben genannten freien Programme)

Das wollte ich noch sagen:

- Ich bin LP Informatik und InnoScout der Schule. Die Schulleitung hat den Auftrag für diese Umfrage an mich weitergeleitet.
- Ich (Leiter IT) bin Debian Maintainer und erstelle auch für unseren Serverbereich eigene Debian-Pakete.
- Ich bin sicher, dass irgendjemand an unserer Schule freie Software einsetzt. Dies geschieht aber nicht flächendeckend.
- Ich wäre froh, wenn offen Software zentral vom Kanton evaluiert und auch unterstützt würden, anstelle sich (bei uns) Microsoft hinzugeben - und dies auch vorzugeben.

Teil III

Analyse

Kapitel 9

Wozu Bildung dient

In diesem Kapitel betrachten wir das Thema „Software“ im Rahmen der Allgemeinbildung. Ist es klug, den Vorgaben, Vorschlägen oder Wünschen der lokalen Wirtschaft zu folgen oder gibt es einen „höheren Auftrag“? Was soll die Schule einer Maturandin und einem Maturanden ins Leben mitgeben?

Wir wissen aus der täglichen schulischen Realität, dass ein grosser Teil der Software, die wir an Schulen einsetzen, von bekannten international tätigen Firmen stammt (z.B. *Microsoft*, *Google*, *Apple*). Das bestätigen praktisch alle in der Schweiz befragten Schulen (siehe Kapitel 8). Zum Teil lehren unsere Schulen sogar bewusst ganz bestimmte Produkte (z.B. *Microsoft Word*, *Excel*, *PowerPoint*) und meiden Alternativen, gestützt auf das Argument, die Wirtschaft wolle das so, oder weil der Lehrkörper, Schüler:innen oder Eltern sich wünschen, was sie selbst „zu Hause haben“.

Das Einführen und Nutzen von Freier Software ist unpopulär. Das alleine schon macht die Verwendung schwierig, denn es besteht ein gewisser Gruppenzwang. Alternativen sind grösstenteils unbekannt, gelten als Terrain für Nerds und Programmierer. Darüber hinaus ist, wenn man einen Computer im Geschäft kauft, Software in der Regel schon vorinstalliert. Bei *Apple*-Hardware (z.B. *MacBooks*) aus technischen Gründen *Apple*-Software, bei anderen Marken (z.B. *Dell*, *Lenovo*, *HP*, *Acer*) ist es *Microsoft Windows*, das gebündelt mit der Hardware verkauft wird. Man kann als Käufer auch nicht selbst wählen, ob man die Software dazukaufen möchte oder nicht (im Fachjargon spricht man von der „Windows tax“, einer steuerähnlichen Abgabe beim Kauf).[1] Das Deinstallieren der Software und Rückfordern der Kosten ist rechtlich möglich, in der Umsetzung aber schwierig, wie ausführlich in mehreren Fällen dokumentiert.[2]

Warum man sich grundsätzlich für *Freie Software* entscheiden soll, haben wir im Kapitel 4 erörtert. Warum soll man die Mühen, die damit verbunden sind, aber auf sich nehmen? Welchen „höheren Auftrag“, falls es einen solchen gibt, definiert hierbei die Idee der Allgemeinbildung?

9.1 Definition

Bildung ist ein Begriff, der vielseitig definiert wird.

> *Bildung kann im Kern als Maß für die Übereinstimmung des persönlichen Wissens und Weltbildes eines Menschen mit der Wirklichkeit verstanden werden. Je höher die Bildung ist, desto größer wird die Fähigkeit, Verständnis für Zusammenhänge zu*

[1] Vgl. Wikipedia, *Bundling of Microsoft Windows*.
[2] Vgl. FSFE, *Activities/Windows Tax Refund - FSFE Wiki*.

entwickeln und wahre Erkenntnisse zu gewinnen.[3]

9.1.1 Ausbildung

Unter *Ausbildung* versteht man den Erwerb von Fachwissen, das Erlernen des Beherrschens eines gewissen Fachs, einer Maschine, eines Programms, einer konkreten Tätigkeit. Eine Fähigkeit, Zusammenhänge zu erkennen und Erkenntnisse zu entwickeln, ist dafür nicht notwendig.

Ausbildung steht im Kontrast zu *Bildung*. Ausbildung ist *Training*. Bildung kann man nicht trainieren. Bildung braucht Zeit, Zeit zur Reflexion.

9.2 Geschichtliche Vorbilder

9.2.1 Immanuel Kant

Der vielleicht einflussreichste deutsche Philosoph der Aufklärung, Immanuel Kant (1724-1804), erklärt den Erziehungsbegriff im Sinn der Aufklärung. In einem 1784 veröffentlichten Artikel in der Berlinischen Monatsschrift spricht er von der Bequemlichkeit der Unmündigkeit.[4]

> *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.*

Man könnte davon das Selbstverständnis der Lehrpersonen ableiten, das Kant von aufgeklärten Menschen verlangt. Sie sollen durchdringen, was ihr Leben und das ihrer Schüler:innen bestimmt, heute insbesondere die Informationstechnologie. Raus aus der „Comfort Zone“! Nicht auf das Urteil von Experten sollen sie sich verlassen, sondern sich selbst ein Urteil bilden und Wege suchen, mitbestimmen und mitgestalten zu können.

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

Die Aufklärung verlangt die Einbeziehung der eigenen Bemühung in Bildung, Religion und den Wissenschaften. Der geistige Wandel stellt damit auch die damals vorherrschenden autoritären Verhältnisse infrage, was wenig später in den Ausbruch der Französischen Revolution mündet. Es war der Aufbruch in eine demokratische Welt. Kant möchte, dass man lernt, zu hinterfragen, um Verantwortung für seine Entscheidungen übernehmen zu können. Anstatt sich bequem auf fertige Lösungen zu verlassen, soll man selbst entscheiden. Das bedeutet aber auch, dass man

[3] Böhm, *Nicht Vielwissen sättigt die Seele*, S. 25.
[4] Kant, „Beantwortung der Frage: Was ist Aufklärung?“

sich die Mühe machen muss, seine Wahlmöglichkeiten durchzudenken und sich anschliessend der Konsequenzen seiner Wahl im Klaren zu sein.

> *Es ist also für jeden einzelnen Menschen schwer, sich aus der ihm beinahe zur Natur gewordenen Unmündigkeit herauszuarbeiten. Er hat sie sogar lieb gewonnen und ist vor der Hand wirklich unfähig, sich seines eigenen Verstandes zu bedienen, weil man ihn niemals den Versuch davon machen ließ. Satzungen und Formeln, diese mechanischen Werkzeuge eines vernünftigen Gebrauchs oder vielmehr Mißbrauchs seiner Naturgaben, sind die Fußschellen einer immerwährenden Unmündigkeit. Wer sie auch abwürfe, würde dennoch auch über den schmalsten Graben einen nur unsicheren Sprung tun, weil er zu dergleichen freier Bewegung nicht gewöhnt ist. Daher gibt es nur Wenige, denen es gelungen ist, durch eigene Bearbeitung ihres Geistes sich aus der Unmündigkeit heraus zu wickeln und dennoch einen sicheren Gang zu tun.*

Kant sagt, man muss die Menschen daran gewöhnen, die Mündigkeit zu praktizieren. Auf die uns im heutigen Alltag umgebenden IT-Systeme gemünzt, kann dies bedeuten, dass wir die *Vier Freiheiten von Software* praktizieren lernen. Das Nutzenlernen von Programmen alleine ist im Kant'schen Verständnis zu wenig. Dadurch dass uns die innere Funktionsweise der Software nicht bekannt ist (*Freedom to study*), können wir die Konsequenzen nicht abschätzen. Ohne am Prozess der Verbesserung von Software teilzunehmen (*Freedom to improve*), bemühen wir uns zu wenig und bleiben unmündige „User" unserer Software.

All dies klingt überzogen für den Fall, dass uns nur *proprietäre Software* zur Verfügung steht. Heute ist die Realität aber eine andere: Es gibt für wahrscheinlich jede populäre *Closed Source*-Software eine brauchbare freie, *Open Source*-Alternative. Das bedeutet, wir *können* uns bemühen. Frei nach Kant hat die Schule damit die Aufgabe, die Schüler:innen diese Mündigkeit üben und festigen zu lassen, also den Umgang mit den *Vier Freiheiten* praktisch zu lehren.

9.2.2 Wilhelm von Humboldt

Ein weiterer einflussreicher Denker der Aufklärung ist der deutsche Gelehrte Wilhelm von Humboldt (1767-1835). Er hat im damaligen Preussen mit der Berliner Universität sein Bildungsideal realisiert. Es stellt die geistige Reifung über das rein zweckorientierte Lernen, also die Bildung über die Ausbildung.[5] Dazu schreibt Humboldt dem preussischen König:[6]

> *Es gibt schlechterdings gewisse Kenntnisse, die allgemein sein müssen, und noch mehr eine gewisse Bildung der Gesinnungen und des Charakters, die keinem fehlen darf. Jeder ist offenbar nur dann ein guter Handwerker, Kaufmann, Soldat und Geschäftsmann, wenn er an sich und ohne Hinsicht auf seinen besonderen Beruf ein guter, anständiger, seinem Stande nach aufgeklärter Mensch und Bürger ist. Gibt ihm der Schulunterricht, was hierfür erforderlich ist, so erwirbt er die besondere Fähigkeit seines Berufs nachher so leicht und behält immer die Freiheit, wie im Leben so oft geschieht, von einem zum andern überzugehen.*

Im Geiste Wilhelm von Humboldts gesprochen, ist das aktive Auseinandersetzen mit Freier Software keine Aufgabe allein des Informatikunterrichts oder gar spezialisierter Mittel- und Hochschulen: Es ist eine Aufgabe der Allgemeinbildung.

[5] Vgl. Lohrmann, *Abitur: Wilhelm von Humboldt - Lernen - Gesellschaft - Planet Wissen*.
[6] Berglar, *Wilhelm von Humboldt*, S. 87.

Die Kenntnis der Möglichkeiten von und der Einflussmöglichkeiten auf (Freie) Software muss Thema der Allgemeinbildung sein, gerade weil IT und IT-Services einen derart invasiven Charakter haben und unsere Gesellschaft dramatisch verändern.

9.2.3 Richard David Precht

Der deutsche Philosoph Richard David Precht kritisiert, dass unsere Gesellschaft die Digitalisierung nahezu ausschliesslich als ein technisches Problem der wirtschaftlichen Konkurrenzfähigkeit begreift.

Er spricht von einer gesamtgesellschaftlichen Herausforderung, in anderen Worten müssen die Entscheidungsträger von morgen diese Herausforderungen heute verstehen lernen, um morgen die richtigen Entscheidungen treffen zu können: Digitalisierungsunterricht als Allgemeinbildung an unseren Schulen.

> *Erbe der Aufklärung* [ist]*, sich die Zukunft von Menschen gestaltet zu denken, sie nicht in Gottes Hand oder in die Hand „einer eigengesetzlichen Evolution von Technologie“ zu legen.*[7]

Konkret heisst das, wir sollen uns nicht den Entwicklungen der IT-Branche hingeben und als „Technologietouristen“ an einer Reise teilnehmen, von der wir das Ziel nicht kennen. Es ist notwendig, dass wir Verantwortung übernehmen und zu verstehen beginnen, dass wir die Zukunft mitgestalten können.

[7] Wikipedia, *Richard David Precht*.

Kapitel 10

Massnahmen

Die in diesem Kapitel vorgeschlagenen Massnahmen sollen erreichen, dass Schüler:innen, der Lehrkörper und die Schuladministration durch praktisches Anwenden ein gutes Verständnis von Freier Software aufbauen und festigen können. Dieses praktische Arbeiten kann sowohl intern an *Open Source*-Software als auch extern an FLOSS-Projekten geschehen.

Im besten Fall wird die Schule somit selbst zu einem System, das kollaborativ, wie ein *Open Source*-Projekt, funktioniert. Die IT wird Teil des Lehrkörpers, die IT-Mitarbeiter konzentrieren sich auf die Weitergabe ihres Wissen – und lernen Neues vielleicht auch von den Schüler:innen.

Es ist empfehlenswert, diese Vorschläge *nicht* als zusätzliche Aufgaben zu sehen und sie auch *nicht* (nur) dem Informatikunterricht zuzuordnen. Zum Erlernen und Leben der Prozesse von Freier Software sollte Platz im Lehrplan gemacht werden und das Thema in allen Fächern präsent sein.

10.1 Schulhomepage selbst verwalten

Wie bei den Beispielen *GNOME* und *ILIAS* in Kapitel 6, soll es möglich sein, dass Schüler:innen *Issues* melden und Anpassungen vorschlagen. Diese Anpassungen (*Merge Requests*) können dann von bestimmten Maintainern (z.B. Rektor:in) angenommen oder zurückgewiesen werden.

10.1.1 Problem

Gemäss Schulverwaltung der PMS wird die Schulwebsite vom Kanton Thurgau mit dem Tool *GOViS* CMS[1] verwaltet. Die Schule hat dafür einen *GOViS*-Account zur Verfügung gestellt bekommen und kann damit inhaltliche Änderungen an der Kantonswebsite im Bereich der Schule vornehmen. Weil mit diesen Accounts direkt Änderungen durchgeführt werden, können nicht beliebige Personen an der Anpassung der Inhalte mitwirken.

10.1.2 Konkrete Vorgehensweise

Parallel zur bestehenden Schulwebsite wird eine Website auf Quellcode-Basis aufgebaut und unter *Versionskontrolle* verwaltet (z.B. auf *GitHub* oder *GitLab*). Eine einfache Variante wäre es, die Website direkt in HTML zu schreiben, die fortgeschrittene Variante, einen Static Site Generator zu verwenden. Beides ist mit *GitHub Pages* oder *GitLab Pages* einfach möglich.

[1] Vgl. GOViS, *GOViS - CMS für Gemeinden, Städte, Kantone, Schulen. Gemeindewebsite.*

Schüler:innen werden im Informatikunterricht eingewiesen, wie das Melden von Fehlern und Verbesserungsvorschlägen gemacht wird (*GitHub*/*GitLab Issues*) oder Anpassungen direkt vorgeschlagen werden (*Pull Requests*). Administratives Personal wird eingewiesen, wie man auf *Issues* und *Pull Requests* reagiert und diese zu einem positiven Abschluss bringt.

Schritt für Schritt werden die Seiten der bestehenden CMS-verwalteten Website migriert. Die aktuelle Website kann am Schluss als Landingpage bestehen bleiben. Diese Landingpage kann Inhalte darstellen, die sich nicht mehr ändern. Alle lebenden Inhalte werden am Schluss von den Schüler:innen, Lehrpersonen und anderen - auch externen - Personen selbständig gewartet und aktualisiert. Die Kontrolle behält die Schuladministration und engagierte Lehrpersonen.

10.1.3 Vorbilder

- FUSS Projektwebsite
  `https://gitlab.fuss.bz.it/fuss/fuss-website`
- Projektwebsite dieser Maturaarbeit
  `https://gitlab.com/fabio-totti-05/matura-freie-software/-/tree/main/website`

10.2 Fokuswoche „Freie Software“

Einmal im Jahr könnte in einer Fokuswoche intensiv das Thema *Freie Software* behandelt werden. Es würde sich anbieten, das parallel zur jährlichen FLOSS-Konferenz *FOSDEM* im Februar stattfinden zu lassen.

10.2.1 Konkrete Vorgehensweise

- Fokus in allen Unterrichtsfächern auf *Freie Software* legen.
- Informationsmaterial in der ganze Schule plazieren; eine kleine Messe mit Messeständen veranstalten, wie bei einer Konferenz.
- Demos und Kurzvorträge von Projekten, neuen Tools und Linux-Distributionen.
- Hands-on Sessions for Neulinge, Pair-Programming und Hacking-Sessions.
- Informationsmaterial von der FSFE bestellen und auflegen.
- Vortragende der Free Software Foundation Europe zu Themenvorträgen einladen.
- Lokale Unternehmen, die mit FLOSS arbeiten, einladen.
- Für Interessierte die (virtuelle) Teilnahme an der *FOSDEM*-Konferenz ermöglichen, erlauben oder sogar anregen.

10.3 Schüler:innen machen IT-Support

Schüler:innen lernen die Herausforderungen in der IT hands-on kennen.

Das IT-Fachpersonal kann sein Wissen weitergeben und kann – zumindest für manuelle Arbeiten – Entlastung durch Unterstützung engagierter Schüler:innen und/oder Lehrpersonen erfahren.

10.3.1 Problem

Die Schul-IT wird häufig von Administrationsmitarbeitern gemanaged. Wissen fliesst kaum oder nicht zwischen den Schüler:innen und dem Personal. In Organisationen, in denen dem IT-Personal Linux-Wissen fehlt, kann das Wissen von Schüler:innen nicht zum IT-Personal fliessen.

10.3.2 Konkrete Vorgehensweise

- Aufwändige manuelle Arbeiten oder Automatisierungsvorhaben (z.B. Installation von Linux-Clients oder alte Hardware mit Linux revitalisieren) im Rahmen von Projekten oder Praktika abarbeiten lassen.
- IT-Mitarbeiter kommen in Klassen und erklären aktuelle IT-Vorhaben unter Einsatz von *Open Source*-Software.
- Ämtli für IT-Support (z.B. Klassen wöchentlich im Wechsel).
- Sich gemeinsam mit IT-Personal und Schüler:innen Input von anderen Schulen holen.

10.3.3 Vorbilder

- Georg-Büchner-Gymnasium (Hannover)
  Siehe Abschnitt 7.3
- Penn High School (Pennsylvania)
  Siehe *Red Hat* Kurzfilm[2]

10.4 Lernstick als Schuldesktop

Schüler:innen lernen das *GNU/Linux-Betriebssystem* kennen, ohne etwas auf ihrem eigenen PC installieren zu müssen. Dazu kann beispielsweise der Berner Lernstick zum Einsatz kommen.

10.4.1 Problem

GNU/Linux als *Betriebssystem* ist weniger bekannt als *Windows* und *macOS*. Das macht die Einführung und Nutzung von Linux-basierter FLOSS-Software schwierig, denn die Akzeptanz fehlt.

10.4.2 Konkrete Vorgehensweise

- Crashkurse in einer Fokuswoche „*Freie Software*“.
- Bewusst Projektarbeiten planen, bei denen *Freie Software* zur Erledigung erforderlich ist, wo also der Stick benötigt wird.

10.5 Einführung von LaTeX für grössere schriftliche Arbeiten

Als Ziel festlegen, dass alle Schüler:innen befähigt sind, ihre Maturaarbeit mit LaTeX umsetzen zu können.

[2] Red Hat, *Penn Manor - Open Source Stories*.

10.5.1 Problem

LaTeX ist in akademischen Kreisen zwar bekannt, in den Schulen und im privaten Umfeld wird nur *Microsoft Word* verwendet.

10.5.2 Konkrete Vorgehensweise

- Unter dem Titel *Arbeiten mit akademischen Ansprüchen* eine Projektarbeit bewusst mit LaTeX machen lassen. Dabei bewusst die Hälfte des Zeitaufwandes oder mehr für das Lernen von LaTeX einplanen.
- Schüler:innen erstellen ein Projekt (Beispielprojekt), zusammen im Unterricht.
- Konzepte und Vorteile von bzw. Umgang mit *Versionskontrolle* (*Git*) lernen.

Kapitel 11

Schlusswort

Rückblickend ist es beeindruckend, welche Leistung die Grössen der Freien Software, Richard Stallman, Eric Raymond, Linus Torvalds, Ian Murdock, Mark Shuttleworth, Marco Börris, Peter Mattis und Spencer Kimball, u.v.m. erbracht haben. Sie haben alles daran gesetzt, eine *Better World* entstehen zu lassen. Eine Welt, in der Software nicht eine Bedrohung darstellt, sondern ein Nutzen ist für alle. Nicht nur für *Big Tech*.

Heute ist der Kampf der Guten immer noch im Gange. Das Böse ist noch nicht besiegt. Aber wir sind auf dem richtigen Weg. Ewig kann sich die dunkle Seite der Macht nicht halten. Die Bedeutung von Freier & *Open Source* Software nimmt immer weiter zu. Auch wenn die *Global Player* es nutzen und versuchen, uns mit ihren Services zu ersticken.

Die Strategie wahrscheinlich aller *Freie Software*-Projekte war immer, erfolgreiche Produkte zu „kopieren“, nachzuprogrammieren und frei zu machen. Das fühlt sich aus heutiger Sicht unehrlich an, aber wenn man darüber nachdenkt, was Richard Stallman und seine Freunde bis heute noch in Bewegung setzen, um den Zustand von damals wieder herzustellen, als alles noch geteilt werden durfte, wie unter Freunden, dann wirkt das doch sehr ritterlich.

Wenn sich dieser Trend fortsetzt, werden wir es schaffen, bald in einer Gesellschaft zu leben, in der *Freie Software* die Norm ist, zumindest in der öffentlichen Verwaltung, den Schulen und überall, wo die Bürger:innen sich Freiheit und Transparenz wünschen.

Und wenn unsere Schulen es tatsächlich schaffen, es allen Schüler:innen verständlich zu machen, dass jede:r (Freie) Software ändern, anpassen, kontrollieren kann, dann werden wir in einer besseren Welt leben. In der Welt der Freiheit, die sich Richard Stallman und seine Freunde gewünscht haben.

Teil IV

Appendix

Literatur

Barale, Stefano. *Che volto avrà la scuola del futuro? Buone leggi e buone pratiche (Stefano Barale, docente, membro CUB SUR)*. FUSS. 28. März 2023. URL: https://video.fuss.bz.it/w/qj6A3B3WpyGCdAgeGViXrd (besucht am 07. 04. 2023).

Barthel, Julia. *Freie Software in der Schule: Mit Linux-Rechnern zur digitalen Nachhaltigkeit*. netzpolitik.org e.V. 2. Juni 2020. URL: https://netzpolitik.org/2020/mit-linux-rechnern-zur-digitalen-nachhaltigkeit (besucht am 17. 04. 2023).

Berglar, Peter. *Wilhelm von Humboldt*. Rowohlt Verlag GmbH, Hamburg, 2015. ISBN: 9783499501616.

BigBlueButton. *BigBlueButton LMS Integrations | Designed for Schools | BigBlueButton*. BigBlueButton, Inc. URL: https://bigbluebutton.org/schools/integrations (besucht am 19. 04. 2023).

— *Development Design Guide | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/development/design (besucht am 19. 04. 2023).

— *Development Guide | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/development/guide (besucht am 19. 04. 2023).

— *FAQs | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/support/faq (besucht am 19. 04. 2023).

— *FAQs | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/support/faq#how-can-i-contribute (besucht am 19. 04. 2023).

— *Install BigBlueButton | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/administration/install (besucht am 19. 04. 2023).

— *new-features | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/new-features (besucht am 19. 04. 2023).

— *Privacy | BigBlueButton*. BigBlueButton, Inc. URL: https://docs.bigbluebutton.org/administration/privacy (besucht am 19. 04. 2023).

Bittner, Fabio. *GitLab - Fabio Bittner / Deutsch SLA*. URL: https://gitlab.com/fabio-totti-05/deutsch-sla (besucht am 14. 04. 2023).

Böhm, Winfried. *Nicht Vielwissen sättigt die Seele*. Hrsg. von Martin Lindauer. Ernst Klett Verlag GmbH, 1988.

Bugzilla. *Installation List - Bugzilla*. bugzilla.org. URL: https://www.bugzilla.org/about/installation-list (besucht am 20. 04. 2023).

Canonical. *launchpad tour*. Canonical. URL: https://launchpad.net/+tour/index (besucht am 10. 03. 2023).

— *Mark Shuttleworth - Canonical Vision*. Canonical. 4. Apr. 2021. URL: https://www.youtube.com/watch?v=PUiHNv_WgkQ (besucht am 10. 03. 2023).

Clark, Kendra. *Google's children's data privacy settlement explained*. The Drum, Carnyx Group Limited. 14. Dez. 2023. URL: https://www.thedrum.com/news/2021/12/14/google-s-children-s-data-privacy-settlement-explained (besucht am 08. 04. 2023).

Community, LibreOffice. *Change LibreOffice Default Look and Feel - libreofficehelp.com*. 2. Juli 2022. URL: https://www.libreofficehelp.com/change-libreoffice-default-look-and-feel (besucht am 20. 04. 2023).

Community, LibreOffice. *LibreOffice looks DATED and PIXELATED. Is this intentional? - English - Ask LibreOffice*. URL: https://ask.libreoffice.org/t/libreoffice-looks-dated-and-pixelated-is-this-intentional/39551 (besucht am 20. 04. 2023).

Creative Commons. *Share your work - Creative Commons*. Creative Commons. URL: https://creativecommons.org/share-your-work (besucht am 23. 04. 2023).

Debian Webmaster Team. *Debian – How to join Debian*. Debian Project. 10. Jan. 2023. URL: https://www.debian.org/devel/join (besucht am 09. 03. 2023).

Departement für Erziehung und Kultur Kanton Thurgau. *Lehrplan Volksschule Thurgau*. Departement für Erziehung und Kultur Kanton Thurgau. URL: https://tg.lehrplan.ch/index.php?code=b%7C10%7C0&la=yes (besucht am 22. 04. 2023).

Dörig, Vincent. *LaTeX.css — Make your website look like a LaTeX document*. Mai 2022. URL: https://latex.vercel.app (besucht am 14. 04. 2023).

Ewing, Larry. *Penguin Tutorial - Notes on creation*. URL: https://isc.tamu.edu/~lewing/linux/notes.html (besucht am 26. 03. 2023).

FSFE. *Activities/Windows Tax Refund - FSFE Wiki*. FSFE e.V. URL: https://wiki.fsfe.org/Activities/WindowsTaxRefund (besucht am 24. 03. 2023).

— *Lernen wie die Profis - Lösungen*. FSFE e.V. 14. Mai 2021. URL: https://lernenwiedieprofis.ch/loesungen/#sonstige-werkzeuge (besucht am 08. 04. 2023).

FUSS. *FUSS - Documentazione*. FUSS. URL: https://fuss.bz.it/page/doc (besucht am 07. 04. 2023).

— *Incontro Emilia Romagna (23.03.2023 mattina)*. FUSS. 29. März 2023. URL: https://intendenzabz.comeinclasse.it/playback/presentation/2.3/6de50259cf585ab4f3bbce557093403d410
1679985704179 (besucht am 07. 04. 2023).

Gerrit. *Gerrit's History | Gerrit Code Review*. Gerrit project. URL: https://www.gerritcodereview.com/about.html (besucht am 20. 04. 2023).

GNOME. *Introducing GNOME 3.24 'Portland'*. Youtube. 2017. URL: https://www.youtube.com/watch?v=_Z1PAXiyTB0 (besucht am 18. 02. 2023).

Google. *Produkte für Lehre und Lernen*. Google. URL: https://edu.google.com (besucht am 08. 04. 2023).

GOViS. *GOViS - CMS für Gemeinden, Städte, Kantone, Schulen. Gemeindewebsite*. backslash AG. URL: https://www.govis.ch (besucht am 04. 04. 2023).

Justin Mayer, et al. *Quickstart - Pelican 4.8.0*. URL: https://docs.getpelican.com/en/latest/quickstart.html (besucht am 14. 04. 2023).

— *Themes - Pelican 4.8.0*. URL: https://docs.getpelican.com/en/latest/themes.html (besucht am 14. 04. 2023).

Kambach, Nils. *Open-Source-Lizenzen Vergleich & Übersicht*. URL: https://www.berlios.de/open-source-lizenzen (besucht am 23. 01. 2023).

Kant, Immanuel. „Beantwortung der Frage: Was ist Aufklärung?“ In: *Berlinische Monatsschrift* (1784).

Klaus-Peter Kerbusk, DER SPIEGEL. *Brücke in die neue Welt - DER SPIEGEL*. 5. Sep. 1999. URL: https://www.spiegel.de/wirtschaft/bruecke-in-die-neue-welt-a-904d776d-0002-0001-0000-000014718402 (besucht am 20. 04. 2023).

Libera Chat contributors. *Guides | Libera Chat*. Libera.Chat. URL: https://libera.chat/guides (besucht am 20. 04. 2023).

LinuxForDevices. *The Linux Kernel – Explained in an Easy to Understand Way*. JournalDev IT Services Private Ltd. URL: https://www.linuxfordevices.com/tutorials/linux/linux-kernel (besucht am 02. 02. 2023).

Lohrmann, Julia. *Abitur: Wilhelm von Humboldt - Lernen - Gesellschaft - Planet Wissen*. WDR. URL: https://www.planet-wissen.de/gesellschaft/lernen/abitur_die_reifepruefung/pwiewilhelmvonhumboldteinmannfuermenschenbildung100.html (besucht am 25. 03. 2023).

Manfred Dworschak, DER SPIEGEL. *Kalkulierter Selbstmord - DER SPIEGEL*. 22. Nov. 1998. URL: https://www.spiegel.de/wissenschaft/kalkulierter-selbstmord-a-d9620bc8-0002-0001-0000-000008035571 (besucht am 20. 04. 2023).

Microsoft Education Team. *How data and AI are changing the world of education*. Microsoft. 7. Apr. 2022. URL: https://educationblog.microsoft.com/en-us/2022/04/how-data-and-ai-are-changing-the-world-of-education (besucht am 08. 04. 2023).

— *Leitfaden zum Education Insights Premium*. Microsoft. URL: https://support.microsoft.com/de-de/topic/leitfaden-zum-education-insights-premium-8738d1b1-4e1c-49bd-9e8d-b5292474c347 (besucht am 08. 04. 2023).

— *Unlock students' full potential with Education Insights in Microsoft Teams*. Microsoft. 15. Feb. 2023. URL: https://educationblog.microsoft.com/en-us/2023/02/unlock-students-full-potential-with-education-insights-in-microsoft-teams (besucht am 08. 04. 2023).

Moore, J.T.S. *Revolution OS*. Youtube. 2001. URL: https://www.youtube.com/watch?v=jw8K460vx1c (besucht am 19. 01. 2023).

Netlify. *Site Generators - A List of Static Site Generators for Jamstack Sites*. Netlify. URL: https://jamstack.org/generators (besucht am 14. 04. 2023).

Open Source Initiative. *Licenses – Open Source Initiative*. Open Source Initiative (OSI). URL: https://opensource.org/licenses (besucht am 23. 01. 2023).

— *The Open Source Definition – Open Source Initiative*. Open Source Initiative (OSI). 22. März 2007. URL: https://opensource.org/osd (besucht am 23. 01. 2023).

Pietschmann, Chris. *Linux is Most Used OS in Microsoft Azure – over 50 percent of VM cores | Build5Nines*. Build5Nines.com. 11. Mai 2020. URL: https://build5nines.com/linux-is-most-used-os-in-microsoft-azure-over-50-percent-fo-vm-cores (besucht am 23. 04. 2023).

Praveen Arimbrathodiyil, et al. *gitlab - Debian Wiki*. Debian Project. 10. Jan. 2023. URL: https://wiki.debian.org/gitlab (besucht am 30. 01. 2023).

Raymond, Eric Steven. *The Cathedral and the Bazaar*. Snowball Publishing, 2010. ISBN: 9781607962281.

Red Hat. *Penn Manor - Open Source Stories*. Red Hat, Inc. 24. Juni 2015. URL: https://www.redhat.com/en/open-source-stories/penn-manor (besucht am 20. 04. 2023).

— *What is the Linux kernel?* Red Hat, Inc. URL: https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel (besucht am 30. 01. 2023).

Reisinger, Charlie. *Enabling students in a digital age: Charlie Reisinger at TEDx Lancaster*. TEDx Talks. 30. Mai 2014. URL: https://www.youtube.com/watch?v=f8Co37G02Fc (besucht am 20. 04. 2023).

Sneddon, Joey. *LibreOffice Has a New Look in Ubuntu 20.10 - OMG! Ubuntu!* 11. Okt. 2020. URL: https://www.omgubuntu.co.uk/2020/10/libreoffice-yaru-icons-ubuntu (besucht am 20. 04. 2023).

Software in the Public Interest. *Debian – Reasons to use Debian*. Debian Project. 11. Feb. 2022. URL: https://www.debian.org/intro/why_debian (besucht am 30. 01. 2023).

sr.solutions. *01. Ausbildungsplanung mit ILIAS in der Industrie*. 15. Nov. 2015. URL: https://www.youtube.com/watch?v=UUiuOMqjois (besucht am 11. 03. 2023).

Stallman, Richard und Lawrence Lessing. *Free Software, Free Society*. Hrsg. von Joshua Gay. Free Software Foundation, Inc., 2002. ISBN: 1882114981.

Stu (KDE). *A Year of the Linux Desktop*. 4. Juli 2013. URL: https://dot.kde.org/2013/07/04/year-linux-desktop (besucht am 20. 01. 2023).

The Document Foundation. *Was ist LibreOffice? - LibreOffice*. The Document Foundation. URL: https://de.libreoffice.org/discover/libreoffice (besucht am 20. 04. 2023).

The GIMP Team. *GIMP - About GIMP*. The GIMP project. URL: https://www.gimp.org/about/ (besucht am 26. 03. 2023).

Torvalds, Linus und David Diamond. *Just for Fun*. HarperCollins, 2002. ISBN: 0066620732.

Ubuntu. *The story of Ubuntu*. Ubuntu. URL: https://ubuntu.com/about (besucht am 10. 03. 2023).

Wawrzyniak, Jessica. *Der DigitalPakt Schule | Digitalcourage*. Digitalcourage e.V. 18. März 2020. URL: https://digitalcourage.de/blog/2020/der-digitalpakt-schule (besucht am 22. 04. 2023).
— *Freie Software für Schulen | Digitalcourage*. Digitalcourage e.V. 27. März 2020. URL: https://digitalcourage.de/blog/2020/freie-software-fuer-schulen (besucht am 22. 04. 2023).
Westcliff High Scool for Girls. *IT & Software*. Westcliff High Scool for Girls. URL: https://www.whsg.info/400/it-software (besucht am 04. 04. 2023).
Wikimedia. *MediaWiki*. Wikimedia. URL: https://www.mediawiki.org/wiki/MediaWiki (besucht am 20. 04. 2023).
Wikipedia. *Apache OpenOffice*. Wikipedia. URL: https://de.wikipedia.org/wiki/Apache_OpenOffice (besucht am 20. 04. 2023).
— *BigBlueButton*. Wikipedia. URL: https://de.wikipedia.org/wiki/BigBlueButton (besucht am 19. 04. 2023).
— *BigBlueButton*. Wikipedia. URL: https://en.wikipedia.org/wiki/BigBlueButton (besucht am 19. 04. 2023).
— *Bundling of Microsoft Windows*. Wikipedia. URL: https://en.wikipedia.org/wiki/Bundling_of_Microsoft_Windows (besucht am 24. 03. 2023).
— *Freie Lizenz*. Wikipedia. URL: https://de.wikipedia.org/wiki/Freie_Lizenz (besucht am 23. 04. 2023).
— *Gnome*. Wikipedia. URL: https://de.wikipedia.org/wiki/Gnome (besucht am 18. 02. 2023).
— *ILIAS*. Wikipedia. URL: https://de.wikipedia.org/wiki/ILIAS (besucht am 11. 03. 2023).
— *LibreOffice*. Wikipedia. URL: https://de.wikipedia.org/wiki/LibreOffice (besucht am 20. 04. 2023).
— *List of Linux distributions*. Wikipedia. URL: https://en.wikipedia.org/wiki/List_of_Linux_distributions (besucht am 12. 04. 2023).
— *Marco Börries*. Wikipedia. URL: https://de.wikipedia.org/wiki/Marco_B%C3%B6rries (besucht am 20. 04. 2023).
— *Mark Shuttleworth*. Wikipedia. URL: https://en.wikipedia.org/wiki/Mark_Shuttleworth (besucht am 10. 03. 2023).
— *Peter Mattis*. Wikipedia. URL: https://en.wikipedia.org/wiki/Peter_Mattis (besucht am 26. 03. 2023).
— *Richard David Precht*. Wikipedia. URL: https://de.wikipedia.org/wiki/Richard_David_Precht (besucht am 03. 04. 2023).
— *Star Division*. Wikipedia. URL: https://de.wikipedia.org/wiki/Star_Division (besucht am 20. 04. 2023).
— *StarOffice*. Wikipedia. URL: https://de.wikipedia.org/wiki/StarOffice (besucht am 20. 04. 2023).
— *Ubuntu*. Wikipedia. URL: https://en.wikipedia.org/wiki/Ubuntu (besucht am 10. 03. 2023).
William Jon McCann, et al. *Getting in Touch*. GNOME. 2022. URL: https://wiki.gnome.org/GettingInTouch (besucht am 10. 03. 2023).

Abbildungsverzeichnis

1.1 Auszug Wegleitung Maturaarbeit, PMS Kreuzlingen 2022/23 3
1.2 Unsere Schüler:innen, unsere Mission 4
4.1 Das Gegenteil von *Copyright – all rights reserved* 11
4.2 Kombinationsmöglichkeiten für Creative Commons-Lizenzen 14
5.1 Linus Torvalds in comp.os.minix (Usenet), 25. August 1991 22
5.2 LibreOffice Online auf Nextcloud 31
5.3 Microsoft Office 365 31
5.4 Google Workspace 31
6.1 VSCodium mit *latex-workshop* Extension 36
6.2 MarkDown-Eingabedateien und die SSG-Konfiguration 37
6.3 Änderungen am Wiki 38
6.4 GitLab Repository Issue 38
6.5 GitHub Pull Request 39
6.6 Recherche nach einer Umfrage-Applikation 41
7.1 Alles gratis! Microsoft ist spendabel 43
7.2 Mission des *FUSS*-Projekts 45
8.1 Abschluss-Statement einer Teilnehmer:in 48
8.2 Frage 2, Bekannte Begriffe 49
8.3 Frage 3, Freedom to use 49
8.4 Frage 4, Freedom to study 50
8.5 Frage 5, Freedom to improve 50
8.6 Frage 6, Freedom to share 51
8.7 Frage 7, Bug Reports, Feature Requests 51
8.8 Frage 8, Neutrale Technologiebegriffe 52
8.9 Frage 9, Ist Software Thema im Unterricht 52
8.10 Frage 10, Auftrag der Allgemeinbildung 53

8.11 Frage 11, Werden Schüler Entscheidungsträger 53
8.12 Frage 12, Freie Software einführen 54
8.13 Frage 13, Wer zahlt die IT-Ausgaben 55
8.14 Frage 14, Summe der IT-Ausgaben 55
8.15 Frage 15, Regionale Support-Partner 56

Tabellenverzeichnis

5.1 Infrastruktur 17
5.2 Schulinformations- und Verwaltungssysteme 17
5.3 Kommunikation 18
5.4 Lernmanagementsysteme 18
5.5 Office 19
5.6 Grafik und Desktop Publishing 20
5.7 Multimedia 20

Glossar

Adaptive Learning Auf Deutsch, Adaptives Lernen, ist eine Methode, die vor allem in den Bereichen Pädagogik und Online-Marketing Anwendung findet. Beim Adaptive Learning werden Computeralgorithmen und künstliche Intelligenz eingesetzt, um die Interaktion mit den Lernenden zu steuern und massgeschneiderte Ressourcen und Lernaktivitäten bereitzustellen, die auf die individuellen Bedürfnisse von Lernenden eingehen. 7, 8

Benutzeroberfläche Die Benutzeroberfläche ist die Schnittstelle zwischen einer Maschine und deren Benutzer, die ihre gegenseitige Interaktion ermöglicht. 24, 25, 30, 34, 40

Betriebssystem Ein Computer bzw. Mobile Geräten bestehen aus Hardware, einem Betriebssystem und verschiedenen Programmen/Apps. Ein Betriebssystem kann man als ein „Vermittler“ zwischen Hardware und Software sehen.
In der Regel besteht ein Betriebssystem aus mehreren Softwarekomponenten: Kernel, Scheduler, Dateisystem, Speichermanager, Peripheriemanager, Benutzeroberfläche und Druckspooler. Zu den Betriebssystemen für Desktop-Computer gehören Microsoft Windows, macOS, Linux-Distributionen, Unix-ähnliche Systeme, BSD und Chrome OS, während es für mobile Geräte iOS, Android, HarmonyOS, Windows Phone, Sailfish OS, Symbian OS und KaiOS gibt. 11, 16, 21, 24, 34, 42, 66

Big Tech Big Tech bezieht sich auf die dominantesten und grössten Unternehmen in der Informationstechnologie und insbesondere auf die grossen US-Technologieunternehmen Alphabet (Google), Amazon, Apple, Meta (Facebook) und Microsoft. Diese Unternehmen werden auch als die „Big Five“ bezeichnet. 7, 8, 68

Breakout-Session Eine Funktion von Konferenz-Software, die es ermöglicht, Teilnehmer:innen eines Online-Meetings temporär in kleinere Gruppen aufzuteilen, z.B. um mehrere Themen parallel zu diskutieren, aus denen sie nach Ablauf einer bestimmten Zeit wieder in die gemeinsame Hauptsitzung zurückkehren. 28, 29

Bug Ein Fehler in einer Software bzw. seinem Programmcode. 22

Bug Report Formalisierte Beschreibung eines Fehlers in einem Bug Tracker oder Issue Tracker. 28, 31

Bug Tracker Software zur Protokollierung und Abarbeitung von Software Bugs oder Feature Requests. 27

Copyleft Der Name für die Methode der GNU Public Lizenz, um allen die 4 Freiheiten von Freier Software zu geben, umgesetzt mit der Copyright-Gesetzgebung. 11, 14, 15

Desktop-Umgebung Eine Desktop-Umgebung ist eine grafische Oberfläche eines Betriebssystems, die es dem Benutzer ermöglicht, den Computer oder die Software auf benutzerfreundliche Weise durch Interaktion mit grafischen Objekten zu nutzen.

Die Desktop-Umgebung ist entweder vom Hersteller des Betriebssystems vorgegeben – wie bei Windows oder macOS – oder kann vom Benutzer frei ausgewählt werden – wie bei Unix-, Linux, usw. 24

Digitalisierung Der aktiv betriebene Wandel einer Gesellschaft oder Organisation, mit dem Ziel manuelle oder analoge Vorgänge durch computergestützte Vorgänge und Artefakte zu ersetzen. 8

Distribution Ist eine Zusammenstellung eines Betriebssystems, das aus dem Linux-Kernel, einem GNU-Basissystem und mehreren anderen darauf abgestimmten Anwendungen, meist in Form von Software-Paketen, besteht. 16, 22–24, 35

Feature Request Formalisierte Anfrage für eine neue Funktionalität oder die Änderung des Verhaltens von Software. 22, 28

FOSDEM Kurz für *Free and Open Source Developer Meeting*, `https://fosdem.org`, ist ein zweitägiges Event in Brüssel und online, das von Freiwilligen organisiert wird, um die Nutzung von freier und quelloffener Software zu fördern. 65

Framework Eine Sammlung von Software Bibliotheken, die einem bestimmten Ziel in der Softwareentwicklung dient, z.B. Entwicklung einer Website oder einem API (Programmierschnittstelle). 40

Freemium Ein Kombinationswort aus „free“ und „premium“ und ein Geschäftsmodell für ein Produkt oder Service, mit dem eine Dienstleistung anfangs kostenlos zur Verfügung gestellt wird und die Verrechnung von Kosten erst nach einem Testzeitraum, meist über ein Abomodell, zu laufen beginnen. 40

Freie Software Software mit einer Lizenz, die dem Benutzer bestimmte Freiheiten gewährt: Die Freiheit das Programm auszuführen, die Freiheit das Programm zu untersuchen und anzupassen, die Freiheit das Programm weiterzugeben und die Freiheit das Programm zu verbessern und der Öffentlichkeit freizugeben; damit die ganze Gesellschaft davon profitieren kann. 1, 3, 5, 8, 10–12, 15, 25, 30, 34, 44, 45, 60, 65, 66, 68

GitHub Social Coding Plattform, wie GitLab. Seit 2018 in Besitz von Microsoft. 22, 27, 29, 39, 40, 64, 65

GitLab GitLab ist eine Webplattform für Softwareentwicklung und Versionskontrolle mit Git. Weitere Funktionen von GitLab sind die Versionskontrolle von Git, die Zugriffskontrolle, Fehlerverfolgung, Software-Feature-Anfragen, Aufgabenmanagement, kontinuierliche Integration und Wikis für jedes Projekt. 23–25, 28, 35, 36, 39, 64, 65

GNU/Linux Die Kombination des GNU-Betriebssystems mit dem Linux-Betriebssystemkern. 11, 66

Host Ein physikalischer oder virtueller Computer, der Services für andere Computer zur Verfügung stellt. Umgangsprachlich spricht man auch von *Server*, wobei dies streng genommen das Programm bezeichnet, das am Host läuft und das Service zur Verfügung stellt. 40

Instanz Eine konkrete Kopie oder lauffähige Installation einer Applikation, die ein Service anbietet (z.B. eine Bugzilla-Instanz). 22–25, 27, 28, 31, 38, 39

Issue Ein generischer Begriff für ein Problem in der Softwareentwicklung oder -wartung. Synonyme: Bug, Bug Report, Feature Request. 23, 24, 29, 64, 65

Issue Tracker Ein Issue Tracker ist eine Software, die es Entwicklern ermöglicht, Probleme und Fehler von Software (Issues/Bugs), Vorschläge für Funktionsverbesserungen, usw. zu organisieren, zu priorisieren und zu verfolgen, bis sie behoben sind.
Wenn Benutzer ein Programm nutzen und Probleme bemerken oder neue Funktionen oder Verbesserungen vorschlagen möchten, können den Issue Tracker nutzen, um Feedback zu hinterlassen. 22, 27

Lehrplan 21 Projekt der 21 deutsch- und mehrsprachigen Kantone der Schweiz zur Harmonisierung der Lehrpläne für die Volksschule. 8

LoCo Ubuntu Local Community (LoCo) Team Portal. 24

Mailingliste Der Nachrichtenaustausch per E-Mail innerhalb einer geschlossenen Gruppe von Menschen. 23, 24

Merge Request Ist eine Mitteilung eines Entwicklers/Entwicklungsteams in einer Versionsverwaltung wie GitLab, über eine Code-Änderungen in einem Softwareprojekt mit der Bitte diese Änderungen vor der Übernahme in in einem Entwicklungszweig (Branch) zu überprüfen. 24, 64

Open Source Open Source bezieht sich auf Open Source Software (OSS). Bei OSS ist der Quellcode öffentlich zugänglich. Das bedeutet dass jeder den Quellcode sehen, verändern und weitergeben kann, wie er es für richtig hält.
Gleichzeitig ist Open Source eine Bewegung und eine Arbeitsweise geworden, die auf die Werte des OSS baut. 3, 4, 10–15, 21, 24, 27, 28, 31, 38–42, 62, 64, 66, 68

proprietär Im Eigentum befindlich bzw., im juristischen Sinn, urheberrechtlich geschützt. 13

proprietäre Software Software, dessen Copyright bei einem Unternehmen oder einer Privatperson ist und dessen Quellcode weder frei noch offen ist. Bei proprietärer Software ist die Änderung, Weiterentwicklung sowie die Weitergabe der Software nicht nicht erlaubt. 10, 13, 14, 42, 62

Pull Request Ein anderer Name für *Merge Request*. 22, 27, 39, 65

Shell Eine Schnittstelle eines Computersytems für Menschen. Das kann eine grafische Oberfläche sein, in der Regel wird damit aber eine textbasierte Benutzeroberfläche (Terminal-Programm) beschrieben. 21

Software-Stack Eine Sammlung von Technologiekomponenten, die bei der Ausführung einer speziellen Anwendung helfen. Kann Komponenten wie Datenbank-Tools, Kodierungs-Frameworks und Client-Interface-Tools umfassen. 16

Source Code Repository Ein Repository ist eine Möglichkeit, Quellcode eines Programmes zu speichern. Bei Versionskontrollsoftware (VCS) werden darin Informationen über Dateien, ihre Grösse, das Erstellungsdatum und alle an ihnen vorgenommenen Änderungen (oft als „Commits“ bezeichnet) gespeichert. Git ist ein gängiges Beispiele für Software, die Repositories verwenden. 23, 24, 38, 40, 41

Versionskontrolle Ein Konzept bzw. Software in der Softwareentwicklung, mit dem Änderungen an Software kontrolliert (als Versionen) gespeichert werden. 23, 64, 67

Akronyme

CC Creative Commons. 14

CMS Content Management System. 42, 64, 65

CSS Cascading Style Sheets. 37

DOS Disk Operating System. 21

FAQ Frequently Asked Questions. 29

FLOSS Free/Libre Open Source Software. 13, 16, 47, 64–66

FOSS Free & Open Source Software. 13

FSF Free Software Foundation. 11, 12

FSFE Free Software Foundation Europe. 16, 65

FUSS Free Upgrade for a digital Sustainable School. 16, 45, 46

GIMP GNU Image Manipulation Program. 25

GNU GNU is Not Unix. 11, 12

GPL General Public License. 15, 24, 27

GTK GIMP-Toolkit. 24, 25

HTML HyperText Markup Language. 37, 64

IRC Internet Relay Chat. 23–26, 31

IT Information Technology. 5, 63–66

LGPL Lesser General Public License. 15, 24

LUG Linux User Group. 45

NDA Non-Disclosure Agreement. 11

OSI Open Source Initiative. 12

PDF Portable Document Format. 35–37

SSG Static Site Generator. 5, 37, 64

VCS Version Control System. 23