

Der Einsatz von proprietärer Software in der Rosa-Luxemburg-Stiftung ist historisch bedingt stark ausgeprägt. Hierbei spielt die Firma Microsoft in dem Umfeld der Betriebssysteme für lokalen Computer und zentralen Servern sowie im Bereich der Office-Applikationen die größte Rolle.

Gerade in diesen Einsatzbereichen gibt es eine starke Tendenz, bisher lokal operierende Informationsprozesse zunehmend in die Cloud zu verlagern oder stärker an die Unternehmen zu binden. So ist zusätzlich in den letzten Jahren ein starker Trend zu verzeichnen, sich von einmalig zu beschaffenden Lizenzen zu lösen und diese im Mietverfahren mit eingebauten Nutzungskontrollen nur noch anzubieten, wie es die Firma Adobe bereits fast vollständig für seine Produktpalette umgesetzt hat. Hierdurch treten relevante technische Fragen für die Nutzer*innen immer stärker in den Hintergrund und es kommt damit zu einer schleichenden Intransparenz wo die Daten der Nutzer*innen eigentlich gespeichert werden und wer alles darauf Zugriff hat. Die Abhängigkeit von den strategischen Ausrichtungen der Unternehmen stellt für die Stiftung damit zunehmend ein Risiko dar. Am Beispiel der Anpassung der Lizenzpolitik durch Microsoft im Fall CERN (European Council for Nuclear Research) wird dies sehr gut deutlich.¹

Aus politischen und organisatorischen Gründe ist eine Prüfung vorzunehmen, die eine strategische Fokussierung auf Lösungen durch Freie Software/Open-Source-Software (FOSS) bietet und zusätzlich aufzeigen soll, ob und wie eine deutliche Reduzierung der Abhängigkeiten zu den bisherigen Konzernen mit ihren proprietären Lösungen möglich ist.

Damit besteht für die Rosa-Luxemburg-Stiftung die Möglichkeit, sich vor den zwangsweisen Vorgaben möglicher Cloud-Migrationen oder Änderungen bei den Lizenzbestimmungen der Software-Hersteller zu schützen und damit den eigenen politischen Anspruch auch nach außen sichtbar zu machen.

Die Machbarkeitsstudie soll beleuchten welche nicht-technischen Implikationen eine mögliche Erweiterung oder Umstellung hin zu Freier Software haben könnte. Der Erfolg einer Umstellung und Nutzung von Software ist entscheidend von der Akzeptanz der Nutzer*innen abhängig. Die Studie soll aufzeigen, welche Systeme der aktuellen IT-Infrastruktur der Rosa-Luxemburg-Stiftung in Zukunft durch freie Software ergänzt oder ersetzt werden können und welche Auswirkungen das auf die Administration der IT der Stiftung sowie auf die Abläufe der Geschäftsprozesse und die tägliche PC-Nutzung haben würde. Auch der Kostenrahmen für so eine Umstellung soll aus der Studie hervorgehen.

Es wird erwartet, dass der Umsetzungszeitraum für die Studie auf ca. 6 Monate angesetzt werden sollte. Dabei wird mit einem Kostenrahmen von ca. 30T€ gerechnet. Hier wurde berücksichtigt, dass weitere Zuarbeiten und Unterstützungen durch die Kolleg*innen der Stiftungen erfolgen kann.

Nach Abschluss der Studie und Vorlage des Ergebnisses, müssen im nächsten Schritt die weiteren Handlungsoptionen geprüft werden. Diese sollten dann dem Vorstand zur weiteren Empfehlung vorgelegt werden.

Die Anlage ist das Ergebnis einer kleinen Vorstudie, die grundsätzlich die Idee der Umstellung auf freier Software beleuchtet.

¹ Siehe Anlage „Vorstudie zur Einführung einer Menüerweiterung durch Freie- / Open-Source-Produkte bei der Rosa-Luxemburg-Stiftung“

*Konsortium Datenkollektiv / UNI:CODE IT Solutions / Frank Hofmann
Juli 2019*

Vorstudie zur Einführung einer Menüerweiterung durch Freie- / Open-Source-Produkte bei der Rosa-Luxemburg-Stiftung

*Carsten Ungewitter
Pavel Dahncke
Niels Reich*

datenkollektiv.net



Eine Studie im Auftrag der Rosa-Luxemburg-Stiftung

Inhaltsverzeichnis

1	Technik ist politisch	3
2	Der Einsatz von Open-Source-Software in Unternehmen: Vorteile und Hindernisgründe	4
2.0.1	Einsatz von Open-Source-Software	4
2.0.2	Einsatz von Open-Source-Software speziell im Desktop Bereich . .	5
3	Anforderungen an eine Machbarkeitsstudie	7
4	Fazit	10
5	Anhang	11
5.1	Gesellschaftliche Aspekte von Produktion und Nutzung Freier Software .	11
5.1.1	Benutzer*innen, Software und Schnittstellen	11
5.1.2	Gesellschaftliche Rahmenbedingungen bei der Produktion Freier Software: Das Verhältnis von proprietären Diensten und freiem Code	13
5.2	Erfahrungen bei der Umstellung von Organisationen auf FOSS	14
5.2.1	Stadtverwaltung München – Linux	15
5.2.2	Cern - European Council for Nuclear Research	16
5.2.3	datenkollektiv.net: Nutzung von Linux-Desktops in NGOs am Beispiel Weiterdenken – Heinrich Böll Stiftung Sachsen	16
5.3	Detaillierte Pro-Contra-Übersicht Open-Source-Technologien und Desktop-Systeme	19
5.3.1	Vergleich zwischen Open-Source-Software (OSS) und proprietärer Software (PS)	19
5.3.2	Vergleich zwischen Linux- und Windows-Desktop	22
5.4	Aufwandsschätzung der Hauptstudie	25
5.5	Glossar	26
5.6	Autoren	28
5.6.1	UNI:CODE IT Solutions	28
5.6.2	datenkollektiv.net	29

1 Technik ist politisch

Wenn eine Organisation wie die Rosa-Luxemburg-Stiftung überlegt, ihre digitale Infrastruktur in Richtung Freie Software/Open-Source-Software (FOSS)¹ zu erweitern oder langfristig zu verschieben, kann und wird dies sowohl technische als auch nicht-technische Gründe haben.

Technik ist nicht neutral und nicht frei von Ideologie. Schon aus diesem Grund gibt es keine rein pragmatische Antwort auf die Frage nach einer „besten“ technischen Lösung für die konkreten Bedürfnisse. Bei der Frage einer „Menüerweiterung“² oder gar einer Migration der Software-Infrastruktur in Richtung Freie Software müssen daher zuerst die verschiedenen relevanten Entscheidungs-Dimensionen identifiziert werden.

Alle Erfahrungen mit der Einführung und Umstellung von Software zeigen: Die damit verbundenen Prozesse und Veränderungen sind keine Selbstläufer. Software und ihre Benutzeroberflächen greifen für viele Menschen tief in ihre Arbeitsgewohnheiten ein. Veränderungen sind nicht immer willkommen. Das trifft für Freie Software gleichermaßen zu wie für proprietäre³. Auch auf dieser Ebene gilt es zu überlegen: Was sind gute Rahmenbedingungen für eine Einführung, Erweiterung und Verschiebung der Software-Infrastruktur in Richtung FOSS und welche davon sind bei der Rosa-Luxemburg-Stiftung grundsätzlich gegeben?

Im Folgenden nähern wir uns einigen dieser Dimensionen und versuchen zu skizzieren, was eine Linux-Menüerweiterung für die Rosa-Luxemburg-Stiftung bedeuten würde, ob und unter welchen Umständen sie zu empfehlen wäre.

Zunächst werden Vor- und Nachteile von Linux im Vergleich mit proprietärer Software besprochen. Ein eigener Abschnitt widmet sich der Frage von Linux als Desktop-Betriebssystem (Abschnitt 2).

Schließlich wird ein Themenkatalog für die Erstellung einer detaillierten Machbarkeitsstudie vorgeschlagen, die die Grundlage für die Umsetzung einer Menüerweiterung darstellen würde (Abschnitt 3).

Eher grundsätzliche politische Überlegungen zur Entwicklung und Nutzung Freier Software und deren gesellschaftlichen Bedingungen finden sich im Anhang (Abschnitt 5.1).

Ebenfalls im Anhang findet sich die Darstellung von Erfahrungen anderer Organisationen bei der Einführung Freier Software (Abschnitt 5.2).

¹Zu den Begrifflichkeiten Open-Source / Freie Software siehe Glossar. FOSS: Free and Open Source Software.

²Wir nutzen hier den von Markus Euskirchen ins Spiel gebrachten Begriff der Menüerweiterung. Dieser meint in diesem Zusammenhang kein Computermenü - sondern die grundsätzliche Erweiterung des Angebot an nutzbaren Möglichkeiten. Im Gegensatz zum Begriff der „Migration“ (→Glossar) - also der Umstellung von einem Betriebssystem auf ein anderes - geht es hier um zusätzliche Angebote und ggf. um eine langsame Verschiebung in Richtung Freie Software. →Siehe auch Glossar

³Proprietäre Software als Gegensatz zu Freier Software: siehe Glossar.

2 Der Einsatz von Open-Source-Software in Unternehmen: Vorteile und Hinderungsgründe

2.0.1 Einsatz von Open-Source-Software

Aus einer sozialistischen Perspektive liegt auf der Hand, dass Kooperation Technologien weiter voranbringt als Konkurrenz. Entsprechend hat die auf Kooperation basierende Open-Source-Software (OSS¹) sich derart erfolgreich entwickelt, dass sie von den Herstellern proprietärer Software nicht mehr ignoriert werden kann.²

Im Anhang (Abschnitt 5.3) findet sich eine detaillierte Gegenüberstellung von Vor- und Nachteilen. Die dargelegten Punkte sprechen technisch klar für OSS. Dies zeigt sich auch in der Öffnung der großen Software-Konzerne in dieser Richtung, da sie sonst technologisch abgehängt würden. So integriert Windows 10 ein Linux-Subsystem und viele neue Projekte von Google und Microsoft wurden unter eine quelloffene Lizenz gestellt.³ Dies betrifft vor allem die zugrundeliegenden Technologien und Bibliotheken, während die den Nutzer*innen zugewandten Produkte proprietär bleiben.

Für die spezifische Perspektive der RLS dürften politische und organisatorische Gründe eine entscheidende Rolle bei der Wahl der zukünftigen Ausrichtung spielen. Als Akteur mit sozialistischem Selbstverständnis findet sich in der FOSS-Bewegung eine natürliche politische Partnerin, denn es geht der FOSS-Bewegung darum die "virtuellen Produktionsmittel" der Software-Branche unter gesellschaftliche Aufsicht zu stellen. Durch einen stärkeren Fokus auf Open-Source-Lösungen im eigenen Hause wird nicht nur der politische Anspruch nach außen sichtbar und unterstrichen, sondern die Organisation schützt sich auch vor einer möglichen Erpressbarkeit bei einer von den Herstellern forcierten Cloud-Migration oder Änderungen der Lizenzbestimmungen der Software-Hersteller.^{4 5}

Damit der Einsatz von Open-Source keine weitere Ausbreitung prekärer Arbeitsbedingungen in sich birgt, sollte der Fokus nicht in der Kostensenkung im Bereich IT liegen,

¹Siehe Glossar → Freie Software, → Open Source Software

²Hegemann, Lisa. 2018. „Übernahme von Github: Microsoft investiert ins Image“. Die Zeit. <https://www.zeit.de/digital/internet/2018-06/microsoft-github-software-entwickler-uebernahme/seite-2> (12. Juli 2019).

³Etwa sichtbar an den Beispielen Visual Studio Code, C#, golang oder auch dem Kauf der Github-Plattform durch Microsoft.

⁴Siehe die Lizenzanpassung durch Microsoft im Fall CERN beschrieben im Anhang

⁵Miedl, Wolfgang. 2018. „Microsofts Cloud-Zwang drückt die Anwender: mbuf fordert bessere Migrationsunterstützung und flexiblere Lizenzoptionen“. sharepoint 360. <https://sharepoint360.de/microsofts-cloud-zwang-drueckt-die-anwender-mbuf-fordert-bessere-migrationsunterstuetzung-und-flexiblere-lizenzoptionen/> (12. Juli 2019)

sondern die genutzten Projekte sollten durch Spenden und Supportverträge in ihrem Bestand gesichert werden.⁶

2.0.2 Einsatz von Open-Source-Software speziell im Desktop Bereich

Der ebenfalls im Anhang (Abschnitt 5.3) detailliert geführte Vergleich von Linux-OSS-Desktopumgebungen und dem Windows Desktop liefert ein komplexeres Bild. Zwar warten moderne Linux-Desktops mit einer eleganten, zeitgemäßen User-Experience auf, die sie nicht nur für IT-nahe Anwender*innen zu einer beliebten Arbeitsumgebung werden lassen; zwar passen sie sich viel eher den Workflows und persönlichen Vorlieben der Nutzerinnen an, behandeln Daten vertraulich und ihre Verarbeitung transparent. Dennoch bleibt ihre Einführung innerhalb einer Organisation eine Herausforderung. Mitarbeiter*innen bringen oft nur Qualifikationen für Microsoft Windows und Microsoft Office in das Unternehmen und benötigen ggf. Weiterbildung. Eingesetzte Software-Produkte sind unter Umständen nicht nativ auf den Linux-Systemen lauffähig oder werden von den Herstellern so nicht unterstützt. Verwendete Hardware muss auf ihre Eignung getestet werden. Die Server-Infrastruktur muss ggf. angepasst werden. Auch IT Sicherheit muß in vielen Punkten neu gedacht werden.⁷

Letztendlich kann das Roll-Out einer Linux-Desktop-Alternative für die RLS aber aus verschiedenen Gründen empfohlen werden:

- aus politischen und taktischen Beweggründen. Der Desktop macht im täglichen Arbeiten, aber auch bei öffentlichen Präsentationen den Einsatz von OSS deutlich und unterstreicht so die Ernsthaftigkeit des eigenen politischen Anspruchs. Er bildet ein Mosaiksteinchen einer gelebten und erstrittenen sozialistischen Kultur.
- Weiterhin wird mit einem Linux-Desktop eine Alternative geschaffen, um nicht den weiteren Entwicklungen des Unternehmens Microsoft ausgeliefert zu sein. In Anlage Cern - European Council for Nuclear Research (Abschnitt 5.2.2) wird dargelegt, wie eine Bildungseinrichtung ihre entsprechende Einstufung und das damit verbundene kostengünstige Lizenzmodell bei Microsoft verlieren kann.
- Im Hinblick auf Datenschutz und Datenhoheit stellt ein Linux-Desktop eine zukunftssichere Alternative dar. Im Hinblick auf die neuen Windows Betriebssysteme ab Windows 10 wurde wiederholt in Zweifel gezogen, ob diese überhaupt

⁶Scholz, Trebor. 2019. „Wovon sollen Open-Source-ProgrammiererInnen eigentlich leben?“. Green net project <https://greennetproject.org/2019/03/26/trebor-scholz-wovon-sollen-open-source-programmiererinnen-eigentlich-leben/> (12. Juli 2019)

⁷Die Steuerung und Kontrolle aller Clients über „Gruppenrichtlinien“ gibt es in der Linux-Welt so nicht. Dafür gibt es andere Möglichkeiten des zentralen Client-Managements.

datenschutzkonform betrieben werden können.^{8 9 10}

- Im Bereich der Windows-Desktops und anderer proprietärer Software sind zwei Tendenzen festzustellen: Zum Einen die zunehmend alternativlose Integration in die Cloud-Plattformen der Hersteller (z.B. Microsoft, Adobe), zum Anderen ein Vermarktungskonzept von Software das auf die Verwertung von Nutzer*innendaten abzielt.¹¹

⁸Matthes, Marie-Charlotte, und Ingo Dachwitz. 2018. „Politik zur Datenschleuder Windows 10: Aufsichtsbehörden müssen handeln“. netzpolitik.org. <https://netzpolitik.org/2018/politik-zur-datenschleuder-windows-10-aufsichtsbehoerden-muessen-handeln/> (12. Juli 2019).

⁹Baehring, Jill. 2018. „Datenschutz-Folgenabschätzung zeigt hohe Risiken bei Microsoft Office ProPlus Enterprise“. Privacy Company. <https://www.privacycompany.de/datenschutz-folgenabschätzung-zeigt-risiken-bei-microsoft-office-proplus-enterprise/> (12. Juli 2019).

¹⁰Schulzki-Haddouti, Christiane. 2018. „Entscheidung für Closed Source: Behörden ignorieren Sicherheitsbedenken gegenüber Windows 10“. c't – magazin für computertechnik 5/2018: 116–19. <https://www.heise.de/newsticker/meldung/Behoerden-ignorieren-Sicherheitsbedenken-gegenueber-Windows-10-3971133.html> (12. Juli 2019).

¹¹Bei der Nutzung von Windows-Desktop wird mit Office 365, Sharepoint und ähnlichem wird in naher Zukunft die Abgabe der eigenen Daten „in die Cloud“ unumgänglich. Auch hier sind mittel- und langfristig Lösungen mit freier Software eine sichere Alternative um die Hoheit über die eigenen Daten zu gewährleisten.

3 Anforderungen an eine Machbarkeitsstudie

Die Möglichkeit und die Bedingungen für eine Erweiterung bzw. einen sukzessiven Wechsel zu Linux-Desktops und Freier Software müssen innerhalb einer ausführlicheren Machbarkeitsstudie beantwortet werden. Im Folgenden wird ein Vorschlag für die Themenfelder einer solchen Machbarkeitsstudie unterbreitet. Datenschutz und Datensicherheit sollten dabei als Querschnittsthemen betrachtet werden, die in allen allen Bereichen analysiert werden müssen.

Form, Inhalt und Umfang der Studie hängen schließlich davon ab, inwieweit diese über die spezifische Situation der Rosa-Luxemburg-Stiftung hinaus generalisierbar sein soll – im Sinne eines Modells, an dem sich auch andere Organisationen orientieren können und das auch über die RLS hinaus Wirkung für den Einsatz von Freier Software entfalten kann.

Eine Studie zur Nutzung Freier Software in Organisationen sollte auch beleuchten welche nicht-technischen Implikationen eine Erweiterung/Umstellung hin zu Freier Software hat. Dazu gehören wie im Anhang (Abschnitt 5.1) kurz skizziert auch techniktheoretische und politische Überlegungen (Stichwort Commons).

Da der Erfolg bei der Umstellung und Nutzung von Software entscheidend von der Akzeptanz der Nutzer*innen abhängt, muss sich die Machbarkeitsstudie auch zentral diesem Aspekt widmen.

Die Studie soll Auskunft darüber geben, welche Komponenten der aktuellen IT-Landschaft der RLS in Zukunft durch freie Software ergänzt bzw. ersetzt werden können und welche Auswirkungen das auf Administration der Gesamt-IT sowie auf die Abläufe der Geschäftsprozesse und die tägliche PC-Nutzung haben würde.

Ergebnis der Studie sind unterschiedliche Szenarien mit einem unterschiedlichen Grad an Erweiterung bzw. Migration der IT-Landschaft auf freie Software.

Dabei werden die Kosten der Umsetzung sowie die langfristige Kostenentwicklung durch eine sukzessive Umstellung auf freie Software geschätzt (Abschnitt 5.4).

Arbeitsplatzanalyse und Anforderungen erfassen In einem ersten Schritt werden die bisherigen Arbeitsabläufe und Anforderungen der Nutzer*innen unter Windows und der aktuell genutzten Software analysiert. Dazu wird es sinnvoll sein, exemplarische Einsatzgebiete in unterschiedlichen Bereichen zu identifizieren: Verwaltung, Buchhaltung, wissenschaftliche Mitarbeiter*innen, Grafik/Design, etc.

Aufschlüsselung der Client-Applikationen Aus der Arbeitsplatzanalyse entsteht ein Katalog aus bisher genutzten Anwendungen und Arbeitsroutinen. Pro Anwendung

werden etwaige Open-Source Alternativen, vorhandene proprietäre Linux Versionen oder die notwendige Benutzung von virtuellen Umgebungen evaluiert. Terminalserver¹ sind als Alternative zur Benutzung von proprietärer nicht unter Linux lauffähiger Software zu betrachten.

Analyse der Server- und Backend-Systeme Die Studie muss eine Aufstellung der vorhandenen Software im Server- und Backendbereich beinhalten. Für Serversysteme wird analysiert, welche Alternativen auf Basis freier Software grundsätzlich existieren. Für jede Client-Anwendung muss evaluiert werden, inwieweit eine Zusammenarbeit mit dem vorhandenen Server Backend möglich ist und wo Anwendungen ersetzt oder zusätzlich aufgebaut werden müssen. Beispiele: Exchange Mailserver, ActiveDirectory Domaincontroller, Dateidienste (Fileserver)

Anpassung der organisatorischen Prozesse & Rechteverwaltung Die Studie muss gezielt darauf eingehen, wie und ob organisatorische Prozesse sich durch den Einsatz einer veränderten Basis an Software verändern. Im Speziellen, wie sich die Arbeit für die Administrator*innen verändert wenn neue Benutzer*innen angelegt werden müssen oder neue Hardware installiert werden muss. Das vorhandene Rechtekonzept muss an die neuen Clients angepasst werden, um das Einbinden in die vorhanden Struktur zu erleichtern.

Konzeption Linux Desktop Ziel der Studie sollte der Prototyp eines Referenz-Desktop sein. Zentral ist dabei die Auswahl einer geeigneten Linux-Distribution sowie einer (oder mehrerer) Desktop-Umgebung(en). Hier stellt sich auch die Frage, ob kostenpflichtige Enterprise-Distributionen inkl. Herstellersupport² ausgewählt werden sollen oder auf freien Community-Distributionen gesetzt wird.³

Schulungskonzept Es werden zwei Schulungskonzepte erarbeitet. Erstens müssen die Administrator*innen geschult werden, um den Betrieb sicherstellen zu können und zweitens die Anwender*innen, um ihre Tätigkeiten weiterhin ausführen zu können. Zusätzlich soll mitgedacht werden, dass die Erarbeitung der Unterlagen am Ende ein Nachschlagewerk erzeugt, das alle Mitarbeitenden dazu befähigt einfachere Probleme selbstständig zu lösen und ihren Client optimal als Werkzeug einzusetzen.

¹In Frage kommen Windows-Terminal-Server (siehe Glossar) mit Remote-Sitzungen oder auch lokale Virtualisierung eines Windows-Betriebssystems innerhalb der Linux-Umgebung.

²Freie Software heißt nicht immer kostenlos. Es gibt Enterprise-Linux-Distributionen mit unterschiedlichen Geschäftsmodellen: 1. Vom Quellcode her vollständig freie Distributionen, bei denen aber ein Support-Vertrag mit dem Hersteller geschlossen werden muss, der z.B. bestimmte Update-Zyklen garantiert. 2. Distributionen die in Teilen proprietären Code enthalten und deren Nutzung kostenpflichtig ist. Häufig gibt es von letzteren auch eine sogenannter Community-Variante ohne proprietäre Bestandteile und ohne Support oder für bestimmte Einsatzzwecke ist die Nutzung kostenlos.

³Eine "positive Nutzer*innen-Erfahrung" hängt nicht zuletzt davon ab, inwieweit die Nutzer*innen das Gefühl von Autonomie bei der Nutzung ihrer Geräte haben, damit so weit wie möglich wie mit einem eigenen Computer agieren zu können (Können sie zum Beispiel Einstellungen ändern oder Software installieren?). Hier muss eine Abwägung zwischen "Nutzer*innen-Autonomie" und Sicherheit getroffen werden. Diese Frage könnte ebenfalls Teil der Machbarkeitsstudie sein.

Zeitplan und Kostenschätzung Die Studie muss einen belastbaren Zeitplan für die Umsetzung des Entwicklungsvorhabens enthalten. Aus diesem Zeitplan lässt sich dann auch eine Kostenabschätzung für die Umsetzung ableiten.

Policy Checkliste für den Einsatz neuer Software Im Rahmen der Studie soll eine Checkliste erstellt werden, anhand derer sich Software vor Einführung nach technischen und politischen Punkten evaluieren lässt.

Einbindung Externer und “Bring your own device”⁴ Eine moderne IT kommt nicht mehr ohne Überlegungen zu “Bring your own device” aus. Dürfen Mitarbeiter*innen ihre privaten Geräte zu Arbeitszwecken benutzen? Wie lassen sich diese den Sicherheits- und Datenschutzrichtlinien entsprechend in das Firmennetzwerk einbinden? Diese Überlegungen müssen natürlich auch für Linux Clients angestellt werden.

Ein ähnlicher Punkt ist die Einbindung von Geräten etwaiger externer Mitarbeiter*innen, dies betrifft auch Büros, die nicht oder nicht vollständig in die IT Landschaft eingebunden sind.

Interne (sichere) Kommunikation bei der RLS Die interne Kommunikation vieler Unternehmen wird aus diversen Gründen immer seltener per Mail abgewickelt. Unter anderem geringe Verfügbarkeit, “Spaghetti-E-mails” und langsame Rückmeldungen lassen Mitarbeiter*innen zu privaten Messengern greifen, um miteinander über Arbeitsthemen zu kommunizieren. Zusätzlich lässt sich die Sicherheit bei E-mails oft nicht oder nur sehr umständlich umsetzen.

Es soll daher untersucht werden, ob es möglich ist, mithilfe von freier Software einen internen Kommunikationskanal für die RLS zu erstellen, der sicher und benutzungsfreundlich ist.

⁴Auch BYOD (siehe Glossar)

4 Fazit

Gegenüber einer Vollmigration ist eine „Menüerweiterung“ mit einem Linux-Desktop sicher der richtige Weg, um Freie Software innerhalb und über die Grenzen der Rosa-Luxemburg-Stiftung hinaus zu befördern. Da die Vorteile von Freier Software gegenüber proprietärer Software nicht alleine und – in Bezug auf die Endnutzer*innen betrachtet – auch nicht in erster Linie im technischen und Usability-Bereich liegen, ist es von Vorteil, wenn die Organisation die gesellschaftspolitischen Ideen Freier Software teilt und diese aktiv vertritt.

Von der technischen und administrativen Seite her ist eine Menüerweiterung und eine damit verbundene heterogene IT-Landschaft mit dem Parallelbetrieb von Windows- und Linux-Clients möglich. Gerade im Hinblick auf eine zunehmende Tendenz zu „bring your own device“ ist eine Serverarchitektur, die nicht auf einen speziellen Client fokussiert, ohnehin eine zeitgemäße Entwicklung.

Damit das „erweiterte Menü“ – also die Linux-Desktops – auch als Alternative wahrgenommen und genutzt werden, ist jedoch eine aktive Einführungsstrategie und ein entsprechendes Schulungs- und Supportkonzept zu entwickeln.

Es gibt zahlreiche Erfahrungen in der Nutzung von Freier Software bzw. einer „Linux-Migration“ in größeren Organisationen.¹ Der Erfolg hängt allerdings von vielen Faktoren ab und Erfahrungen lassen sich kaum eins zu eins auf andere Organisationen übertragen. Eine ausführlichere Machbarkeitsstudie sollte daher die konkreten Rahmenbedingungen bei der Rosa-Luxemburg-Stiftung untersuchen und ein Konzept zur verstärkten Nutzung Freier Software (Menüerweiterung) entwickeln. Wir würden empfehlen, die Machbarkeitsstudie so zu gestalten, dass über die konkrete Fragestellung für die Rosa-Luxemburg-Stiftung hinaus auch untersucht wird, unter welchen Bedingungen Freie Software ihre emanzipatorischen Potentiale entfalten kann und wie auch andere Organisationen von den Erkenntnissen profitieren können.

¹Einen der umfassendsten Migrationsleitfäden kommt übrigens von der Bundesstelle für Informationstechnik. Leider wurde die sehr ausführliche Studie seit 2012 nicht mehr weiter geführt. (Migrationsleitfaden Leitfaden für die Migration von Software. 2012. Berlin: Referat BIT A4 - Standards und Methoden, Kompetenzzentrum Open Source Software (CC OSS) in der Bundesstelle für Informationstechnik - Bundesverwaltungsamt. http://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und-Standards/migrationsleitfaden_4_0_download.pdf?__blob=publicationFile (14. Juli 2019).)

5 Anhang

5.1 Gesellschaftliche Aspekte von Produktion und Nutzung Freier Software

Software ist politisch. Obwohl es eine Binsenweisheit ist, dass mittlerweile fast alle Lebensbereiche mehr oder weniger stark von digitalen Technologien durchzogen sind, ist das Bewusstsein für die politischen Aspekte von Software relativ wenig verbreitet.

Die politischen Aspekte betreffen dabei unter anderem:

- die Schnittstellen zwischen Benutzer*innen und Technik, z.B. bei der Frage, welche Sicht die Software auf die Wirklichkeit zulässt, wie Nutzer*innen geleitet werden oder welche Art von Arbeitsorganisation nahegelegt wird.
- die private Verfügungsgewalt über relevante Kommunikationstechnologien. Dies betrifft einerseits Plattformen, die Quasi-Monopole geworden sind (Facebook, Twitter, google etc.) aber auch Anwendungen die durch ihre Vorherrschaft Standards definieren (Microsoft Office, Adobe PDF).
- die private Aneignung von Nutzer*innen-Daten mit oder ohne expliziter Einwilligung,
- staatliche Überwachung und Zensur
- Produktionsbedingungen der Software

Bei den Überlegungen zu einer Menüerweiterung und Verschiebung zu freier FOSS werden diese Aspekte vermutlich alle mit unterschiedlicher Gewichtung eine Rolle spielen.

Eine Behandlung aller Punkte würde hier den Rahmen sprengen. Daher soll exemplarisch der Aspekt der direkten Interaktion von Benutzer*innen mit den Schnittstellen (User-Interfaces) näher illustriert werden.

Außerdem verlaufen Entwicklung und Nutzung freier und quelloffener Software nicht widerspruchlos. Daher wollen wir auch das Verhältnis von Freier und proprietärer Software und die daraus resultierenden Perspektiven von FOSS andiskutieren.

5.1.1 Benutzer*innen, Software und Schnittstellen

Eine erste Dimension bei der Entscheidung für eine Software betrifft die direkte Interaktion der Benutzer*innen mit Benutzeroberflächen. Klar: Software muss funktionieren,

geeignet sein für den geplanten Zweck und gut bedienbar. Letztere Frage wird häufig unter dem Stichwort „Usability“ verhandelt.

Doch bereits in der Frage der Usability gibt es keine einfache Dichotomie zwischen besserer oder schlechterer Software¹. In der täglichen Praxis stellt sich z.B. schon bei einfachen Anwendungen die Frage ob der Schwerpunkt eher auf einer maximalen Kontrolle der Benutzer*in bei der Verwendung des Programms liegt (ggf. unter Inkaufnahme einer steileren Lernkurve) oder auf der leichten Bedienbarkeit nach wenigen vorgegebenen Schemata.

Gerade im Bereich moderner grafischer Benutzeroberflächen gibt es eine starke Tendenz, technische Vorgänge so weit wie möglich vor den Nutzer*innen zu verbergen. Dies hat aber auch die Kehrseite, dass relevante technische Fragen immer stärker in den Hintergrund treten und Nutzer*innen immer weniger eine Vorstellung z.B. davon haben, wo ihre Daten eigentlich gespeichert werden (wo auf dem Gerät?, in der Cloud?), wer ggf. Zugriff auf die Daten hat oder wie sicher ihre Daten überhaupt sind. Neal Stephenson² stellte schon vor über 15 Jahren die These auf, dass grafische Benutzeroberflächen dazu tendieren, die Nutzer*innen zu entmündigen. Vor dem Hintergrund moderner Oberflächen wie denen von Android, iOS aber auch den neueren Windows10-Oberflächen, ist diese Frage präsenter denn je.

Software ermöglicht Nutzer*innen aber nicht nur mehr oder weniger Steuerungsmöglichkeiten in einem bestimmten Anwendungsfall. Die in einer großen ERP-Lösung³ abgebildeten Geschäftsprozesse beispielsweise sind keine Naturnotwendigkeiten oder neutralen Tatsachen, sondern Ergebnisse einer bestimmten in Software gegossenen Sicht auf eine Organisation. Im täglichen Gebrauch sind sich Nutzer*innen dieser Erkenntnis wohl kaum immer bewusst: Software bestimmt unsere Sicht auf die Aufgabenstellung und die Organisation. Sie präformiert unsere Sicht auf die Realität.

Freie Software bedeutet nicht automatisch eine größere Nutzer*innen-Autonomie. Schließlich können auch hier Benutzeroberflächen programmiert werden, die den Nutzer*innen kaum Handlungsspielräume lassen, ERP-Systeme entwickelt werden, die sich an neoliberalen Unternehmensgrundsätzen orientieren. Allerdings bietet das Entwicklungsmodell freier Software, das nicht ausschließlich kapitalistische Marktlogik verfolgt, mit seinen anderen Prämissen und Voraussetzungen hier eine Chance: Die Verfügungsmöglichkeiten der Menschen über die Technik mit der sie umgehen, können Teil eines offenen Diskurses sein und muss nicht hinter kommerziellen Interessen zurück bleiben.

¹abgesehen natürlich von möglichst fehlerfreiem Code – also wenigen Bugs, die verhindern, dass das Programm überhaupt tut, was es vorgibt zu tun.

²Stephenson, Neal. 2002. Die Diktatur des schönen Scheins: wie grafische Oberflächen den Computernutzer entmündigen. Dt. Erstveröff. München: Goldmann.

³Enterprise-Resource-Planning Software integriert und zentralisiert die Ansicht auf unterschiedliche Bereiche der Organisation (z.B. Projektmanagement und Aufgabenplanung, Buchhaltung, Einkauf/-Verkauf, Lagerhaltung, Fort- und Weiterbildung, Personalmanagement, etc.) in einer gemeinsamen Oberfläche und ermöglicht Verknüpfungen zwischen diesen Feldern.

5.1.2 Gesellschaftliche Rahmenbedingungen bei der Produktion Freier Software: Das Verhältnis von proprietären Diensten und freiem Code

Freie Software kommt mit einer großen Forderung und einem großen Versprechen einher: Die - mittlerweile fast wie Schreibwerkzeuge, Schrift oder sogar Sprache zur grundlegenden menschlichen Kultur gehörenden - digitalen Werkzeuge sollen allen zur freien Verfügung stehen.

Aber obwohl es nicht zuletzt Aktivisten aus der FOSS-Szene waren, die den kometenhaften Aufstieg des Internets ermöglicht haben, und heute ein nicht unbedeutender Teil der Infrastruktur des Internets mit freier Software betrieben wird, hat sich das Versprechen nur sehr widersprüchlich eingelöst.

Mit der zweiten Welle, mit den großen Plattformen des Web 2.0 wie Youtube, Facebook und vielen anderen wurde realisiert, dass nicht nur die Software selbst Teil des Kampfes um eine freie Informationsgesellschaft sein muss. Während zwar ein überwiegender und stetig steigender Anteil der Server im Internet mit freier Software arbeitet, ist das Internet selbst keineswegs freier und egalitärer geworden. Im Gegenteil. Zunehmend relevant für die Kommodifizierung des Netzes wurde nicht die Software selbst – sondern die Kontrolle über die Daten der Nutzer*innen. Technisch ausgedrückt: Die Frage der Plattform löste die Frage der eingesetzten Protokolle (smtp, http, etc.) ab.

Freie Software (oder in der “schwächeren” Begrifflichkeit: Open Source) ist zu einem relevanten Geschäftsmodell geworden. Den größten Beitrag zu Open Source Projekten erbringen mittlerweile große Player wie Google oder auch Microsoft. Sie haben begriffen, dass zunehmend nicht mehr das Eigentum an Quellcode alleine einen Wettbewerbsvorteil darstellt, sondern vielmehr das Betreiben einer Plattform, die im besten Fall eine Quasi-Monopolstellung besitzt. Paradigmatisch steht hierfür das Android-Betriebssystem: Obwohl es überwiegend aus freier Software besteht (was den wenigsten Nutzer*innen bewusst sein dürfte), trägt es dazu bei, die Plattformen von Google als Quasi-Standard zu definieren und den Nutzer*innen wenig Verfügungsmöglichkeiten über ihre Geräte zu geben.

Wie so oft in der Geschichte gelingt es, emanzipatorische Errungenschaften in die Entwicklungsdynamik des Kapitalismus zu integrieren. Die Hoffnung bleibt aber: Der einmal entfesselte Geist emanzipatorischer Ideen lässt sich nur schwer wieder in der Flasche verkorken. Neoliberale Arbeitsverhältnisse bringen mit ihren dazugewonnenen Freiheiten sowohl die Instrumente der Selbstregulation und Selbstunterdrückung mit sich, als sie auch Potentiale für ihre eigene Überwindung in sich tragen. Analog bedeutet das für die kooperative Entwicklung freier Software u.a.: Die Erfahrung der Kooperation in Projekten freier Software überschreitet notwendigerweise längst die Logiken von Konkurrenz und Eigentum – auch dann, wenn die Beteiligten innerhalb von Firmen agieren, die miteinander im Wettbewerb stehen.

Freie Software verbreitet sich aktuell dort am stärksten, wo sie am wenigsten im Widerspruch zu einer kapitalistischen Marktlogik steht. Der Betrieb großer Cloud-Plattformen hat die Entwicklungsprinzipien freier Software geradezu zur Bedingung, weil die Entwicklung der komplexen Systeme mit den Prinzipien von Open Source leichter zu bewerkstelligen ist.

In den Desktop-Bereich dagegen konnte freie Software bislang nur wenig eindringen. Hier steht sie in tatsächlicher Konkurrenz zu den kommerziellen Modellen proprietärer Software, die entsprechend verteidigt werden: durch Monopolstellungen, Konsortien und Lobbypolitik. Microsoft ist allerdings bewusst, dass dies allein auf lange Sicht keine Garantie für Erfolg ist. Entsprechend wird die Lizenzpolitik angepasst und die Benutzer*innen werden mit Abo-Modellen und der Speicherung der Nutzerdaten in der Microsoft-Cloud langfristig und alternativlos an einen Anbieter gebunden.

Daraus kann gefolgert werden dass gerade im Desktop-Bereich die größten politischen und emanzipatorischen Potentiale von freier Software liegen. Hier ist sie die Voraussetzung für einen sicheren, datenschutzfreundlichen und egalitären Zugang zu Information und Wissen und sichert zugleich die Verfügungsgewalt über die eigenen Daten und die eigene Technik.

Die unmittelbare Sichtbarkeit von freier Software im Desktop-Bereich stärkt außerdem einen Diskurs darüber, wie Digitalisierung gestaltet werden soll, wer davon profitieren und wer daran verdienen kann und soll. Akteure aus öffentlicher Verwaltung, Wissenschaft, Bildung und Politik können über den Grundsatz „public money - public code“⁴ z.B. bei der Vergabe von Aufträgen dazu beitragen, dass freie Software und Open Source ihre Potentiale entfalten und damit der Kommodifizierung öffentlicher Güter etwas entgegensetzen.

5.2 Erfahrungen bei der Umstellung von Organisationen auf FOSS

Es gibt aus den letzten Jahren und mittlerweile schon Jahrzehnten eine Reihe von Beispielen für Organisationen, deren IT-Infrastruktur weitgehend mit Freier Software betrieben wird oder die beschlossen haben einen Umstellungsprozess zu Freier Software zu beginnen.

Einige populäre Beispiele werden wahlweise als Erfolg für freie Software (Extremadura, Spanien, München) oder als Mißerfolg (München) gewertet⁵.

Um tatsächlich Erfahrungen aus Erweiterungs- und Migrationsprozessen bewerten zu können, müssen die Organisationen und ihre Ausgangsbedingungen vergleichbar sein; und es müssen relativ detaillierte Informationen über die Art und das Ausmaß der Umstellung vorliegen. Erweiterungs- oder Umstellungsprozesse sind jedoch grundsätzlich nicht aus rein technischer Sicht zu bewerten: Der Erfolg hängt mindestens in gleichem Umfang von organisatorischen und sozialen Rahmenbedingungen ab und davon, wie gut die Mitarbeiter*innen in den Veränderungsprozess mit einbezogen werden.

Eine Liste ohne Anspruch auf Vollständigkeit mit einer Reihe von öffentlichen Einrichtungen weltweit, die auf Linux als Betriebssystem setzen, findet sich bei Wikipe-

⁴Federico, Heinz. „Öffentliche Verwaltung braucht freie Software“. Heinrich-Böll-Stiftung. <https://www.boell.de/de/demokratie/netz-freie-software-oeffentliche-verwaltung-15053.html> (12. Juli 2019).

⁵Euskirchen, Markus. 2017. „Öffentliches Geld für öffentliche Software. Der Kampf um Linux in München“. Rosa-Luxemburg-Stiftung Analysen 43.

dia⁶. Hunderte von Erfolgsgeschichten des Linux-Terminal-Server Projektes⁷ zeigen, dass Linux-Desktops überall auf der Welt genutzt werden.

Im folgenden sind einige Beispiele für den Einsatz von Freier Software im Desktop-Bereich knapp skizziert und eines etwas ausführlicher dargestellt: Weiterdenken, Heinrich Böll Stiftung Sachsen ist zwar verglichen mit der RLS eine kleine Organisation hat aber bezüglich der Aufgaben eine gewisse Ähnlichkeit mit der Rosa-Luxemburg-Stiftung.

5.2.1 Stadtverwaltung München – Linux

Prominentes und viel diskutiertes Beispiel einer gescheiterten Migration auf Linux-Desktop-Clients ist die Stadtverwaltung München. Die Stadt München beschloss im Jahr 2003 unter einer rot-grünen Mehrheit die Vollmigration der öffentlichen Verwaltung auf Linux. Das Projekt befand sich bis 2013 in der Umsetzung. Im Herbst 2017 entschied der Stadtrat, jetzt mit einer Regierungsmehrheit von CSU und SPD, für eine Rückmigration aller Clients auf das Betriebssystem Windows 10. Die Entscheidung wurde fachlich gestützt durch ein Gutachten des Microsoft Partners Accenture.⁸

Der Fall München wirft die Frage auf, ob die freie Linux Desktop Alternative technisch die Reife für den Einsatz in einer professionellen Umgebung wie einer Verwaltung besitzt. Tatsächlich eignet sich das Münchener Migrationsprojekt zur Untersuchung dieser Frage eher weniger. Zu sehr überlagerten hier andere Konfliktfelder die technische Dimension. Dies war einerseits die Politisierung des Ganzen als rot-grünes Prestigeprojekt und die damit verbundene Diskursverschiebung. Weiterhin wurde unter dem Label der Linux Migration eine größere Verwaltungsreform voran getrieben. Dadurch kristallisierte alle Kritik an Prozessveränderungen oder dem Verlust gewachsener Abteilungsstrukturen und Autonomie in einer Kritik am neuen Linux-System.⁹

Dennoch lassen sich Lehren aus dem Scheitern in München ziehen. Eine nachhaltige Hinwendung zu Open-Source-Software muss aus der Belegschaft getragen werden und geht mit einem veränderten Mindset einher. Der Horizont erweitert sich über das eigene Unternehmen und man beteiligt sich (auch innerhalb der Arbeitszeit) an Nutzer*innen- und Entwickler*innen-Communities. Weiterhin wird ein monolithisches Transformationsvorhaben inklusive Umstellung von Arbeitsprozessen und Rechtekonzepten den Realitäten moderner IT aufgrund ihrer Komplexität und rasanten Entwicklung nicht gerecht. Vielmehr sollten sich greifbare Etappenziele gesetzt und der erreichte Zustand kontinuierlich reflektiert werden. Dabei müssen nicht nur technische Änderungen erzielt, sondern auch die Nutzer*innen dauerhaft mitgenommen werden.¹⁰

⁶ „Open-Source-Software in öffentlichen Einrichtungen“. 2019. Wikipedia. https://de.wikipedia.org/w/index.php?title=Open-Source-Software_in_%C3%B6ffentlichen_Einrichtungen&oldid=190116680 (12. Juli 2019).

⁷<http://www.ltsp.org/stories/>

⁸heise online. 2017. „Endgültiges Aus für LiMux: Münchener Stadtrat setzt den Pinguin vor die Tür“. heise online. <https://www.heise.de/newsticker/meldung/Endgueltiges-Aus-fuer-LiMux-Muenchener-Stadtrat-setzt-den-Pinguin-vor-die-Tuer-3900439.html> (12. Juli 2019).

⁹Euskirchen, Markus. 2017. „Öffentliches Geld für Öffentliche software. Der Kampf um Linux in München“. Rosa-Luxemburg-Stiftung Analysen 43.

¹⁰Hier sei das Schlagwort der „Agilen Methoden“ genannt und auf folgenden Einführungsartikel verwie-

5.2.2 Cern - European Council for Nuclear Research

Als weiteres Beispiel ist das CERN zu nennen. Im Sommer diesen Jahres wird der erste Pilot des Projekts MAlt¹¹ (ein alternativer Email-Service und eine Alternative zu Skype for Business Clients) gestartet.

Das Cern hat im Jahr 2018 das Microsoft Alternatives Project (MAlt) ins Leben gerufen, um gezielt der immer stärkeren Bindung an kommerzielle Softwareanbieter entgegen zu wirken. In der Vergangenheit hatten diese dem CERN günstige Angebote unterbreitet und bei Erweiterung der Nutzung die Lizenzkosten stark angehoben.

Dort wird auch hervorgehoben, dass die kollaborative Nutzung der Ressourcen nicht in ein "normales" Geschäftsmodell, das meist auf einer Lizenzkostenermittlung pro Benutzer*in basiert, passen.

Es gibt hier noch keine Ergebnisse, aber die Willkür mit der Microsoft aktuell dem CERN den Status einer Bildungseinrichtung aberkannt hat, zeigt, dass es notwendig ist sich rechtzeitig mit Alternativen zu befassen.

5.2.3 datenkollektiv.net: Nutzung von Linux-Desktops in NGOs am Beispiel Weiterdenken – Heinrich Böll Stiftung Sachsen

Das datenkollektiv betreut die IT einer Reihe von Organisationen, deren Arbeitsplatz-Rechner ausschließlich oder weitgehend auf Linux basieren. Darunter einige Vereine, einen Verlag und zwei Rechtsanwaltskanzleien. Am Beispiel von Weiterdenken soll exemplarisch ein Umstellungsprozess skizziert werden. Im Jahr 2015 entschied sich die Heinrich Böll Landesstiftung Sachsen im Zuge eines Umzugs in ein neues Domizil für eine Umstellung der IT auf Freie Software.

Die Motivation waren neben einer grundsätzlichen Sympathie für Freie Software die Aspekte Datenschutz und Datensicherheit. Die Stiftung hatte – im Nachgang der Edward Snowden Enthüllungen – auch einen thematischen Schwerpunkt auf Datenschutz und Überwachung gesetzt und wollte als Organisation auch im Sinne eines Modellprojektes Verantwortung übernehmen.

Ziel war die möglichst weitgehende Umstellung auf Linux und freie Software sowohl im Bereich der Server als auch der Desktop-Umgebungen ("Vollmigration"). Zugleich sollten die digitalen Möglichkeiten ausgebaut und modernisiert werden. Dazu gehörte u.a. ein für alle Mitarbeiter*innen zentral nutzbares Adressmanagement, die Möglichkeit des Zugangs von außen per VPN und die Systematisierung der Bibliothek. Angestrebt wurde eine Infrastruktur, in der interne und externe Mitarbeiter*innen über Filesharing und Kommunikationssoftware optimal zusammen arbeiten können ohne auf externe Plattformen auszuweichen. Geplant waren etwa 15 feste Arbeitsplätze sowie einige Laptop-Arbeitsplätze.

sen: Schröder, René. 2019. „Agile Methoden: Einführung in die agile Softwareentwicklung“. entwickler.de. <https://entwickler.de/online/agile/agile-methoden-einfuehrung-209035.html> (12. Juli 2019).

¹¹<https://home.cern/news/news/computing/migrating-open-source-technologies>

Infrastruktur vor der Umstellung

- Fileserver: Windows Small Business 2003
- Exchange-Server (E-Mails werden beim Provider abgeholt und lokal gespeichert)
- E-Mail-Hosting bei externem ISP
- Desktop-Betriebssystem(e): Windows 7 auf Arbeitsplatz-Computern (ohne Domänenanmeldung)
- Anwendungssoftware:
 - Firefox als Browser
 - MS Office in verschiedenen Versionen (2007 – 2010)
 - Adressverwaltung auf der Basis einer Access-Datenbank
 - Outlook als Groupware u. E-Mail-Programm
 - Remote-Zugriff auf Desktop über Teamviewer (z.B. zum E-Mail-Abwurf)
 - Adobe InDesign, Photoshop, Acrobat
 - Audacity für Audio-Schnitt (Veranstaltungsmitschnitte)
- Nutzung von weiteren Internet-Services
 - Dropbox
 - google-Kalender

Planung und Umsetzung

Ca. eininhalb Jahre vor dem geplanten Umzug wurde das *datenkollektiv* mit der Planung und Umsetzung betraut.

Bereits ab diesem Zeitpunkt wurden neu eingerichtete PC-Arbeitsplätze als Linux-Desktops ausgeführt um Erfahrungen zu sammeln. Ab diesem Zeitpunkt wurden alle Entscheidungen im Hinblick auf die IT-Infrastruktur (Software/Hardware) darauf hin geprüft, inwieweit sie sich in die zukünftige Umgebung einpassen.

Insgesamt bestand die Umsetzung aus folgenden Komponenten:

- Einführung eines zentralen ID-Managements und Anbindung aller Services per LDAP
- Desktop: Linux Terminal-Server mit Thin-Clients basierend auf Debian Gnu Linux und LTSP.
- Virtualisierung einer Windows-Terminal Umgebung für einzelne Software für die es keine adäquaten freien und linux-kompatiblen Alternativen gab: Adobe CS und Adobe PDF. Zugriff per RDP.¹²

¹²RDP: Remote-Desktop-Protokoll, ein Protokoll um auf Windows-Terminal-Server-Umgebungen über das Netzwerk zuzugreifen. So muss kein Windows Betriebssystem auf dem eigenen Desktop-PC laufen.

- Umstellung des E-Mail-Hostings auf Kolab (mit datenkollektiv.net als E-Mail Provider)
- CiviCRM als Adressdatenbank
- Koha als Bibliothekssoftware
- Wiki für das interne Wissensmanagement
- Owncloud (mittlerweile Nextcloud), zwei Instanzen:
 - Intern: für den Zugriff auf alle Dokumente, per x509-Zertifikat abgesichert
 - Extern: für die Kooperation mit Externen und Download von Dokumenten für Dritte (Dokumentationen, Fotos, etc.)
- E-Mail Groupware¹³ auf Basis von Kolab¹⁴, extern gehostet von datenkollektiv.net. Authentifizierung über das zentrale ID-Management per Ldap.

Erfahrungen

- Office Anwendungssoftware
 - relativ wenig Schwierigkeiten bei der Anpassung bei der Nutzung von Libre-Office. Das kann unter anderem darauf zurück zu führen sein, dass einige Mitarbeiter*innen bereits gute Erfahrungen mit dem Programm hatten.
 - Probleme bereiteten erwartungsgemäß PDF Dokumente mit spezifischen Adobe-Erweiterungen (Adobe Life-Cycle DRM¹⁵), die nur mit dem Acrobat-Reader gelesen werden können. Hierbei handelt es sich meistens um Antrags- und Abrechnungsfomulare öffentlicher Fördergeldgeber. Diesem Problem wurde mit einer Windows-Instanz per RDP-Zugriff begegnet
- Durch gleichzeitige Umstellung und Erweiterung
 - einerseits hohe Komplexität, Hürden bei der Einarbeitung und Schulungsbedarf
 - andererseits: positives Nutzer*innenerlebnis durch die Erweiterungen der Möglichkeiten

¹³Groupware, siehe Begriffsklärung im Glossar.

¹⁴Kolab ist eine freie Groupware-Lösung, die seit 2003 entwickelt wird und ursprünglich vom Bundesamt für Sicherheit in der Informationstechnik in Auftrag gegeben wurde. („Kolab“. 2019. Wikipedia. <https://de.wikipedia.org/w/index.php?title=Kolab&oldid=189545673> (12. Juli 2019).)

¹⁵Dokumente mit digitaler Rechteverwaltung per Adobe-Lifecycle können bislang mit keinen freien Linux-Kompatiblen PDF-Readern verwendet werden. Adobe selbst hat die Entwicklung des Acrobat Readers für Linux mit der Version 9 eingestellt.

Fazit

Die Stiftung Weiterdenken Heinrich Böll Stiftung Sachsen ist einen ambitionierten Weg gegangen um fast die gesamte IT-Infrastruktur mit freier Software zu betreiben. Drei Jahre nach der erfolgten Umstellung kann die Migration als Erfolg bewertet werden. Die Desktop Systeme bewähren sich im Alltag und es gibt wenig Unzufriedenheit mit den genutzten Systemen.

Die Zusammenarbeit und der Datenaustausch mit Externen funktioniert im großen und ganzen relativ reibungslos. Kleinere Hürden bleiben: Insbesondere im Bereich PDF-Formulare und Grafik/Layout wird nach wie vor von einzelnen Mitarbeiter*innen auf Adobe Produkte zurück gegriffen.

Daraus lässt sich schließen, dass unter bestimmten Umständen und bei einer überschaubaren Größe der Organisation eine "Vollmigration" auf Linux-Desktops durchaus möglich ist.

Der Erfolg der Umstellung ist zu einem guten Teil der großen Aufgeschlossenheit gegenüber freier Software zu verdanken. Auch die Vorerfahrungen der Beteiligten mit privaten Linux-Desktops waren von Vorteil. Anfängliche Probleme durch Bugs und ungünstige Einstellungen wurden nicht "Linux" als solchem zugeschrieben sondern als das gesehen, was sie sind: weitgehend unvermeidbare Schwierigkeiten bei der Umstellung und Inbetriebnahme komplexerer Systeme.

Diese Erfahrungen sind aber nicht 1:1 auf andere (insbesondere größere) Organisationen zu übertragen.

5.3 Detaillierte Pro-Contra-Übersicht Open-Source-Technologien und Desktop-Systeme

5.3.1 Vergleich zwischen Open-Source-Software (OSS) und proprietärer Software (PS)

Es existieren verschiedene Open-Source-Lizenzen, die die Nutzung des Quellcodes unterschiedlich reglementieren. Dies reicht von sogenanntem "Look but don't touch" bis zu Lizenzen, die auch für die Nutzung in einem anderen Werk für dieses Quelloffenheit einfordern. Deshalb sind pauschale Aussagen zu Open-Source schwierig.

	OSS	PS
Vorteile	<ul style="list-style-type: none"> • vergesellschaftetes geistiges Eigentum (progressives Moment) • Akkumulation gesellschaftlichen, frei zugänglichen Wissens • Validierung des Werkes hinsichtlich Funktionalität und Sicherheit möglich • offene, transparente Datenverwaltung • Entwicklung von Schnittstellen möglich • individuelle Anpassungen möglich • aktive Nutzer*innen- und Entwickler*innen-Communities • freie Wahl der Supportpartner*in • geringere Abhängigkeit von marktwirtschaftlichen Akteur*innen • höheres Verständnis und damit verantwortungsvollere Nutzung von Technologien • modulare Systeme 	<ul style="list-style-type: none"> • bessere Verwertbarkeit und damit oft mehr Entwicklungskapital • „Security-By-Obscurity“ (Sicherheit durch unbekanntes Funktionsweise)^a • evtl. haftbare Inverkehrbringer*in des Produkts • evtl. größerer Fokus auf die Nutzer*in als Kund*in • eine komplexe Lösung aus einer Hand <p>^aOb dies tatsächlich als Vorteil existiert, wie gelegentlich behauptet, ist anzuzweifeln, vgl: Gierow, Hauke. „Reverse Engineering: Mehr Spaß mit Amazons Dash-Button - Golem.de“. https://www.golem.de/news/reverse-engineering-mehr-spas-mit-amazons-dash-button-1701-125359.html (12. Juli 2019).</p>

	OSS	PS
Nachteile	<ul style="list-style-type: none"> • allgemeine Probleme der Bewertung und damit des Bestandes innerhalb einer kapitalistischen Wirtschaft • Zukunft/ Entwicklung von Projekten evtl. schwer planbar: Finanzierung und schwankendes Interesse ehrenamtlicher Entwickler*innen • evtl. Unterlaufen von branchenüblichen Arbeitsstandards durch großes ehrenamtliches Segment • evtl. mehr Fokus auf die Entwickler*innen-Gemeinschaft als auf die Nutzer*innen 	<ul style="list-style-type: none"> • Akkumulation von privatem, unzugänglichem Wissen • Abhängigkeit vom Monopol des Hersteller-Supports • keine Synergien durch Communities und offenen weltweiten Austausch • intransparente Datenverwaltung und ggf. „Vendor-Lock-In“ • zunehmend problematisches Verhalten bzgl. Datenschutz durch umfassende Analyse von Nutzer*innendaten und Integration von Werbeangeboten • meist (außer evtl. Pluginarchitekturen) keine individuellen Anpassungen möglich • mittlerweile oft Zwang zu SAAS-Lösungen: Verlust der Kontrolle über eigenen Daten, Zwang zu Updates • evtl. Schwierigkeiten bei der Schaffung von Schnittstellen • umfangreiche, monolithische, wenig an den Bedarf angepasste Lösungen • fehlendes Verständnis der eingesetzten Technologien und deren Risiken • Zukunft/ Entwicklung von Projekten evtl. schwer planbar: Abhängigkeit von Änderungen der Lizenz-Politik

5.3.2 Vergleich zwischen Linux- und Windows-Desktop

Beim Vergleich der Desktopumgebungen muss beachtet werden, dass im Gegensatz zur einheitlichen Desktopumgebung einer jeweiligen Windows-Version bei Linux zwischen verschiedenen Desktop-Lösungen gewählt werden kann. Die Bezeichnung Linux beschreibt eine Betriebssystemfamilie, die auf dem Linux-Kernel und GNU-Software (Open-Source-Software unter GPL¹⁶) basiert. Linux-Systeme sind daher immer open-source, erlauben aber durchaus den Betrieb proprietärer Anwendungen. Soll Linux als Desktopsystem verwendet werden, wird es zusammen mit einer Desktopumgebung ausgeliefert. Übliche Umgebungen sind Gnome, KDE, Cinnamon, XFCE, Mate und viele weitere. Die Auswahl der Desktopumgebung richtet sich nach dem Nutzungsprofil (etwa kreativ oder technisch), nach der Leistungsfähigkeit der Hardware und nach persönlichen Vorlieben. Grundsätzlich ist alle Linux-Software auf jeder dieser Desktop-Umgebungen lauffähig. Dagegen lässt sich wenig Anwendungssoftware sowohl auf Windows als auch auf Linux-Systemen nutzen. Der Trend zu webbasierten Benutzeroberflächen hat die Zahl dieser Programme in den vergangenen Jahren jedoch erhöht. Manche Software-Systeme liegen in einer Windows- und Linux-Version vor, so dass sich die hiermit bearbeiteten Dateiformate zwischen den verwendeten Betriebssystemen austauschen lassen.

¹⁶Die GPL (GNU General Public License) ist eine freie Software-Lizenz. Siehe → Lizenzen im Glossar

	Windows Desktop	Linux Desktop
Vorteile	<ul style="list-style-type: none"> • weitest verbreitetes Desktop-System im privaten und geschäftlichen Bereich • höchste Kompatibilität von Anwendungssoftware • größtes unterstütztes Hardwarespektrum durch Treibersupport • beste Integration in Office Umgebung, Server-Anwendungen und Cloud der Firma Microsoft • integrierte Features der Client-Verwaltung über Domain-Controller und Gruppenrichtlinien 	<ul style="list-style-type: none"> • Multiplikatorwirkung durch erhöhte Sichtbarkeit von Open-Source-Technologien • tendenziell stabilerer Betrieb • geringe Angriffsfläche für Schadsoftware • zeitgemäße und ggf. individuell anpassbare Oberfläche • ggf. erhöhte Produktivität und Freude bei der täglichen Arbeit durch persönlich angepassten Desktop • komfortable Weiternutzung älterer Hardware möglich • einheitliche abgesicherte Software-Installation über Paketverwaltung (mehr Selbstverwaltung durch die Nutzer*innen möglich ohne Gefährdung der Sicherheit) • größere Auswahl an und leichtere Verfügbarkeit von freier Anwendungs-Software • keine Lizenzkosten und dadurch langfristige Kostenplanung möglich

	Windows Desktop	Linux Desktop
Nachteile	<ul style="list-style-type: none"> • multiple Abhängigkeit von der Lizenzpolitik eines Unternehmens • Multiplikatorfunktion als Bildungseinrichtung für Microsoft-Produkte • geringere Anpassbarkeit an Nutzungsumgebung und persönliche Bedürfnisse • zunehmender Zwang zur Nutzung von Cloud-Diensten • schwer kontrollierbare externe Erfassung von Nutzer*innenprofilen und Einblendung von Werbeeinhalten • größere Angriffsfläche für Schadsoftware • schwierigere Workflows bei den Themen wissenschaftliches Arbeiten und Publizieren sowie Software-Entwicklung 	<ul style="list-style-type: none"> • wesentlich geringerer Marktanteil • Probleme mit Hardware-Kompatibilität bei sehr neuer oder exotischer Hardware • fehlende Unterstützung durch manche Anwendungssoftware • eventuelle Inkompatibilität zu proprietären Protokollen der verwendeten Serveranwendungen • mehr Weiterbildung nötig für Mitarbeiter*innen mit reinen Windows-Kenntnissen • schwierigere Workflows bei den Themen Grafikdesign und Videoschnitt

5.4 Aufwandsschätzung der Hauptstudie

Studienteil	Bemerkung	min. Projekt- tage	max. Projekt- tage
Arbeitsplatzanalyse und Anforderungen erfassen	In mehreren Abteilungen der RLS werden Mitarbeiter*innen befragt und tägliche Routinen bei der PC-Arbeit analysiert. 5-7 Tage Recherche + 4-5 Tage Auswertung	9	12
Aufschlüsselung der Client-Applikationen	Liste aller Client-Applikationen. Recherche FOSS-Alternativen, Beschreibung und Prüfung auf Praxistauglichkeit.	8	12
Analyse der Server- und Backend-Systeme	Bestandsaufnahme aller Server-Systeme, Recherche und Prüfung von FOSS-Alternativen. Eigene Proof-Of-Concept Tests.	12	16
Anpassung der organisatorischen Prozesse & Rechteverwaltung	Analyse der aktuellen Prozesse und Konzept für die Abbildung innerhalb von alternativen Software-Umgebungen.	9	12
Konzeption Linux Desktop	Auswahl Linuxdistribution, Möglichkeiten automatischer Installation/Update/Provisionierung, Analyse der verschiedenen Möglichkeiten der Integration in Verzeichnis- und Dateiserver. Rückkoppelung bzgl. notwendiger Anpassungen an Server-Systeme. Konzeption Virtualisierung für proprietäre Programme.	14	18
Schulungskonzept	„Curriculum“, Konzeption Materialien, Mitarbeiter*innen-Interviews.	4	6
Zeitplan und Kostenschätzung	Entwicklung Projektplan, Kostenschätzung Umsetzung/ext. Beratung/ggf. Lizenzen/Hardware, etc.. Schätzung der langfristigen Kosten auch im Vergleich zu proprietärer Software unter Berücksichtigung von Lizenzen. Einschätzung der zukünftigen Entwicklung.	8	12

Studienteil	Bemerkung	min. Projekt- tage	max. Projekt- tage
Policy Checkliste für den Einsatz neuer Software	Policy erarbeiten, ggf. Konzept für die Installation ausgewählter Software durch Nutzer*innen.	1	2
Einbindung Externer und “Bring your own device (BYOD)”	Mögliche BYOD-Strategien: z.B. VPN-Server, Terminal-Nutzung, direkter Zugriff. Schutz vor Schadsoftware.	2	3
Interne (sichere) Kommunikation bei der RLS	Analyse der aktuellen Möglichkeiten. Mögliche Erweiterungsszenarien auf Basis freier Software (E-Mail-Verschlüsselung/Chat-Kommunikation/Video-Telefonie)	7	9
Textproduktion, Koordination, Rücksprache, Korrektur, Layout		14	16
Summe Projekt- tage		88	118

5.5 Glossar

Bring your own device bezeichnet die Möglichkeit, private Geräte wie Smartphones, Notebooks etc. für den Einsatz in Unternehmen oder Universitäten zu nutzen und mit diesen sowohl private als auch arbeitsrelevante Tätigkeiten zu erledigen.

Cloud Cloud dient als gemeinsamer Begriff für tatsächlich recht unterschiedliche Konzepte. Gemeinsam ist ihnen, dass Daten nicht mehr auf dem eigenen Rechner gespeichert oder berechnet werden sondern als Service auf anderen Computern, häufig bei großen Drittanbietern außerhalb der eigenen Organisation. Gemeinsam ist allen, dass bestimmte Datenoperationen nicht mehr mit eigener Software auf dem eigenen Rechner laufen sondern als Service-Leistung bei Dritten. Dies reicht von „Software as a Service (SaaS)” (z.B. Microsoft 365) über „Plattform as a Service” wenn z.B. bestimmte Betriebssysteme als Service-Leistung in Anspruch genommen werden bis zu „Infrastructure as a Service”. Über die rein technischen Aspekte hinaus ist relevant, dass es einen großen Trend gibt viele Aspekte der IT nicht mehr entweder auf dem eigenen PC oder in der IT-Abteilung der eigenen Organisation abzuwickeln sondern als Dienstleistung einzukaufen. Das berührt viele Fragen u.a. von Datenschutz, Datensicherheit und Datenautonomie. Für viele Konzerne ist die Cloud ein interessantes Geschäftsmodell, weil sie Kunden dauerhaft mit oft kostenpflichtigen Verträgen bindet und sie oft gleichzeitig die Daten der Kunden für eigene Zwecke nutzen können. Beim klassischen Vertrieb von Software dagegen konnten Kunden – solange ihnen der Funktionsumfang eines Programms ausreichte – eine einmal gekaufte Version auch in der Zukunft unbegrenzt nutzen.

Diese Möglichkeit entfällt mit Cloud-Konzepten meistens.

Desktop Oft als Kurzform von Desktop PC oder auch Arbeitsplatzrechner genutzt – in Abgrenzung zu Servern manchmal auch zu Laptops. Hier ist aber mit Desktop die sogenannte Desktop-Umgebung gemeint, also die Gesamtansicht auf die grafische Benutzeroberfläche. Unter Windows gibt es in der Regel nur einen einzigen Desktop, der von Nutzer*innen nach ihren Bedürfnissen beschränkt konfiguriert werden kann. Für Linux existieren zahlreiche Desktop-Umgebungen mit zum Teil sehr unterschiedlichen Bedienkonzepten.

Freie Software Software, deren Quellcode für alle einsehbar ist und unter bestimmten Lizenzbestimmungen auch zur freien Verfügung steht. Der Begriff Freie Software wird u.a. von der Free-Software-Foundation propagiert – in Abgrenzung zu → Proprietärer Software aber auch um den Aspekt der Freiheit als politischen Aspekt gegenüber → Open-Source-Software zu betonen.

Groupware Software, die der Zusammenarbeit im Team dient. Meistens wird der Begriff für ein (oder mehrere integrierte) Programme gebraucht, bei dem Funktionen wie Kalender, Aufgaben, Adressbücher in die Oberfläche eines E-Mail-Programms integriert sind. Populärstes (proprietäres) Beispiel ist Microsoft Outlook. Freie Alternativen existieren unter anderem mit Kolab, Zimbra, OpenXchange.

Gruppenrichtlinien Gruppenrichtlinien sind ein Konzept aus der Desktop-Client-Verwaltung des Betriebssystems Windows. Clients, die einer „Domäne“ hinzugefügt wurden, können so zentral an einer Stelle verwaltet werden. Dies reicht von der Anmeldung über verfügbare Netzwerklaufwerke bis hin zu diversen lokalen Konfigurationseinstellungen.

Menüerweiterung Im Gegensatz zu einer → Migration soll bei einer Menüerweiterung nicht ein vollständiger Wechsel zu einem anderen Betriebssystem vollzogen werden. Vielmehr sollen Komponenten Freier Software als Alternativen zur Verfügung stehen – und im Idealfall sukzessive von einer größeren Zahl an Mitarbeiter*innen genutzt werden.¹⁷

Migration Im Kontext von IT-Infrastruktur ist mit Migration der Wechsel von einer Plattform auf eine andere gemeint. Das kann einzelne Programme betreffen oder auch das ganze Betriebssystem. Unter „Linux-Migration“ wird meistens der Wechsel von einer Server- und Client-Infrastruktur, die unter Microsoft-Windows läuft hin zu Linux-Servern und/oder Linux- (→)Desktops verstanden.

Lizenzen Für Software greift grundsätzlich das Urheberrecht. Sie darf also nicht unberechtigt kopiert, weitergegeben oder verändert werden. Freie Lizenzen garantieren das Recht, den Quellcode einzusehen, zu verändern und unter bestimmten Bedingungen weiterzugeben. Sie unterscheiden sich vor allem im Hinblick darauf ob (wie z.B. bei der GNU General Public License – GPL1) bei der Nutzung des Quellcodes

¹⁷Der Begriff stammt von Markus Euskirchen. Vgl.:Euskirchen. 2017. a.a.O., S. 17

auch darauf basierende Produkte wiederum unter der gleichen Lizenz veröffentlicht werden müssen (Copyleft) oder nicht (z.B. LGPL¹⁸).

Open-Source-Software Software, deren Quellcode offenliegt. Meist wird der Begriff synonym zu Freier Software verwendet. Jedoch ist auch quelloffene Software möglich, deren Lizenzmodell eine freie Nutzung und Weiterverbreitung ausschließt.

Proprietäre Software Von lat. proprius, jmdm. ausschließlich gehörig. Gemeint ist Software, deren Nutzung aufgrund von Eigentumsrechten beschränkt ist und deren Quellcode im allgemeinen auch nicht öffentlich verfügbar ist. Proprietäre Software ist somit der Gegenbegriff zu Freier Software.

Terminal-Server Bei einem Terminal-Server laufen die Sitzungen vieler Nutzer*innen zentral auf einem Server anstatt auf den eigenen Arbeitsplatz-Computern. Die Nutzer*innen greifen mit sogenannten Thin-Clients (meist einfache Rechner ohne eigenen Festplatte) auf diese entfernten Sitzungen zu (Remote-Session). Terminal-Server gibt es sowohl für Microsoft-Windows als auch für Linux-Systeme. Sie können die Administration und das Rollout erleichtern, weil alles zentral verwaltet und gewartet wird. Außerdem können sie energieeffizienter arbeiten, weil die Thin-Clients nur wenig Ressourcen benötigen. Auch die Anschaffung neuer Arbeitsplatz-computer bei benötigter höherer Rechenleistung entfällt oft.

Virtualisierung Von Virtualisierung wird meist gesprochen, wenn durch Software ein eigenständiges Computersystem dargestellt wird. Dadurch lassen sich innerhalb eines gestarteten Systems weitere – auch unterschiedliche – Betriebssysteme nutzen. Im Desktop-Bereich können somit z.B. Programme gestartet werden, die im eigentlichen System nicht laufen. Z.B. können Programme, die ein MS Windows-Betriebssystem erfordern, innerhalb eines Linux-Systems gestartet werden. Im Serverbereich wird Virtualisierung üblicherweise genutzt um mit weniger Hardware-Bedarf Dienste logisch voneinander zu trennen.

5.6 Autoren

Diese Studie wurde durch Carsten Ungewittwer, Niels Jacob Reich und Pavel Dahncke erstellt. Alle drei sind Mitarbeiter der folgend vorgestellten Unternehmen. Beide Unternehmen verstehen sich als mitarbeiter*innengeführte Kollektivbetriebe.

5.6.1 UNI:CODE IT Solutions

UNI:CODE ist ein junges Unternehmen, das sich aus einem Zusammenschluss von Selbstständigen aus dem Bereich der Informationstechnologie sowie der Ingenieurs- und Naturwissenschaften entwickelt hat. Wir arbeiten dezentral in einem virtuellen Büro zusam-

¹⁸ „GNU General Public License“. 2019. Wikipedia. https://de.wikipedia.org/w/index.php?title=GNU_General_Public_License&oldid=189780785 (12. Juli 2019).

men. Neben unserem Kernteam verfügen wir über ein ausgedehntes Kompetenznetzwerk, auf das wir für unsere Projekte zurückgreifen können.

Als sozial ausgerichtetes Unternehmen pflegen wir langfristige Kontakte zu unserer Kundschaft, bauen unser Kernteam sukzessive aus und tragen zu Open-Source-Projekten bei.

Website: <https://www.unicode-it.de>

5.6.2 datenkollektiv.net

Das datenkollektiv.net besteht seit 2014 als kollektiver Zusammenschluss von bis dahin freiberuflichen IT-Administratoren mit langjähriger Erfahrung in Systemintegration, der Planung und der Administration von lokalen Netzwerken und dem Betrieb von Serversystemen. Wir kennen sehr gut die Bedürfnisse von Non-Profit-Organisationen und beraten bei der Nutzung und beim Umstieg zu Freier Software. Die vom datenkollektiv.net betreuten Organisationen arbeiten zum größten Teil mit Linux-Desktops.

Website: <https://www.datenkollektiv.net>