

FrOSCon '24 · 17.08.2024

Geld ist nicht alles!

**Neue Wege
zur nachhaltigen Beschaffung Freier Software**

Johannes Näder · jn@fsfe.org · [johas@mastodon.social](https://mstdn.social/@johas)
Policy Project Manager · Free Software Foundation Europe



Auf neuen Wegen
lauern Fallen...



- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?





**Die Free Software Foundation Europe
ist ein gemeinnütziger Verein,
der Menschen im selbstbestimmten Umgang
mit Technik unterstützt.**

Was ist Freie Software?

Verwenden

Die Software kann für jeden Zweck verwendet werden, ohne Einschränkungen.

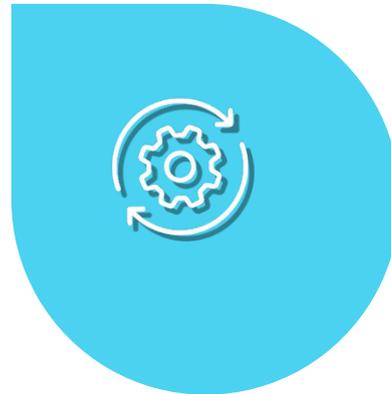
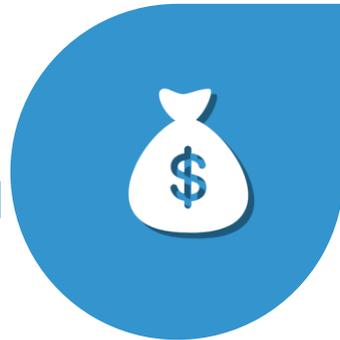


Verstehen

Die Software kann uneingeschränkt untersucht werden.

Verbreiten

Die Software kann uneingeschränkt kopiert und weitergegeben werden.



Verbessern

Die Software kann von den Nutzern oder anderen verbessert und verändert werden.

- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?



Woher kommen wir?

Öffentliche Verwaltungen beschaffen Software.

Freie Software ist meist kein Kriterium.

Den Zuschlag erhalten in der Regel Anbieter proprietärer Software

Geld fließt für Lizenzen, gleiches Produkt wird immer wieder bezahlt, egal ob es besser wird oder nicht

Herstellerabhängigkeit, Gatekeeper und Monokulturen, Preisbildung durch Hersteller, Staat hat keine Kontrolle über die digitale Infrastruktur seiner Verwaltungen

Wohin soll die Reise gehen?

Öffentliche Verwaltungen beschaffen Software.

Freie Software ist meist kein Kaufobjekt.

Den Zuschlag erhalten Anbieter proprietärer Software.

Geld fließt über Grenzen, gleiches Produkt wird immer wieder bezahlt, egal ob es besser wird oder nicht.

Herstellerabhängigkeit, Gatekeeper und Monokulturen. Preisbildung durch Hersteller, Staat hat keine Kontrolle über die digitale Infrastruktur seiner Verwaltungen.

Öffentliche Verwaltungen beschaffen Freie Software.

Öffentliches Geld fließt in Freie Software.

Den Zuschlag erhalten Anbieter Freier Software.

Geld fließt für Maintenance, Entwicklung, Betrieb und Support. Produkte werden besser und stehen der Allgemeinheit zur Verfügung.

Statt Herstellerabhängigkeit: Konkurrenz und Innovation, Preisbildung durch den Markt, Staat und Öffentlichkeit haben Kontrolle über digitale Infrastruktur.



Wir wollen rechtliche Grundlagen, die es erfordern, dass mit öffentlichen Geldern für öffentliche Verwaltungen entwickelte Software unter einer Freie-Software- und Open-Source Lizenz veröffentlicht wird.

**Wenn es sich um öffentliche Gelder handelt,
sollte auch der Code öffentlich sein!**

<https://publiccode.eu>

UNTERSTÜTZENDE ORGANISATIONEN

Die folgenden Organisationen unterstützen unseren [Offenen Brief](#). Wenn deine Organisation sich ebenfalls dem Aufruf für Public Code anschließen will, [kontaktiere uns bitte](#).

Alle Internationale NGOs Verwaltungen Nationale NGOs Regionale NGOs



Das Problem: Vergabekultur 1.0



Beschaffung proprietärer Silo-Produkte.

Ausschreibung Freier Software: Es lauern Fallstricke ...

- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?





PUBLIC MONEY

PUBLIC CODE

**Openwashing: den falschen Eindruck erwecken,
proprietäre Produkte seien Freie Software.**

PUSH RED BUTTON
BEFORE
TURNING HANDLE

Openwashing ist kein neues Phänomen und wird nicht verschwinden.

Es schadet dem Prinzip "Public Money? Public Code" und Freier Software.

Wir müssen Openwashing-Methoden kennen, darüber sprechen und sie eindämmen.

← Zurück

↻ Free Software Foundation Europe teilte



Johannes Näder

@johas

🌐 5. Juni

Thanks to everyone who already gave input on the topic of [#OpenWashing](#) !

As many companies claim to be working on “Open Source” or “Free Software” while distributing proprietary software products, we at the [@fsfe](#) would like to learn more about current market practices.

We're interested in your views and experiences on openwashing! You can submit them here:

share.fsfe.org/apps/forms/s/Z6...

Please spread the word, any input is welcome!

Übersetzen

 share.fsfe.org
FSFE Collaboration

#FreeSoftware

#OpenSource

#PublicCode

← 6

↻ 74

☆ 71



 mastodon

🏠 Startseite

🔔 Benachrichtigungen

🔍 Entdecken

🌐 Live-Feeds

@ Private Erwähnungen

🔖 Lesezeichen

★ Favoriten

📄 Listen

⚙️ Einstellungen



← Zurück

↻ Free Software Foundation Europe teilte



Johannes Näder

@johas

🌐 5. Juni

Thanks to everyone who already gave input on the topic of #OpenWashing!

As many of you know, we are currently distributing about current...

We're interested in your feedback, so please let them here: share.fsfe.org

Please spread the word!

Übersetzen

@johas @fsfe @miklo only for a moment I was hoping for a open source washing machine...

Übersetzen

↩ 1



share.fsfe.org

FSFE Collaboration

#FreeSoftware

#OpenSource

#PublicCode

↩ 6

↻ 74

★ 71



📄 Listen

⚙ Einstellungen



← Zurück

↺ Free Software Foundation Europe teilte

 **Johannes Näder**
@johas

Thanks to everyone who already gave input on the topic

As many comments
distributing
about current

We're interested in
them here:
share.fsfe.org

Please spread the word
[Übersetzen](#)

@johas @fsfe
washing machine
[Übersetzen](#)

↶ 1

share.fsfe.org

FSFE Collaboration

#FreeSoftware

#OpenSource

#PublicCode

↶ 6

↺ 74

★ 71



README

OpenWasher

Open-source controller for washing machine based on STM32F103.

Control:

- Bluetooth 2.0 based on HC-05 module
- Wifi
- Front panel

Software:

- WinForms client
- Android client
- QT client

Current supported platforms:

- Arcadia without drying, collector motor - replace CPU PCB



Languages



Definition

Openwashing:

Proprietäre Software vertreiben und dabei den irreführenden Eindruck erwecken, dass man Freie Software entwickelt, anbietet oder anderweitig unterstützt.

- 1. Proprietäre Software:** Softwareprodukt ist nicht (vollständig) Freie Software
- 2. Irreführende Kommunikation:** Der Hersteller erweckt den Eindruck, dass diese Software Freie Software ist.

Openwashing ist eine Form von Etikettenschwindel – meist mit der Absicht, einen Marktvorteil für proprietäre Softwareprodukte zu erzielen.

Methoden & Strategien

- Buzzwording, irreführende Benennung oder Lizenzierung proprietärer Software
- Nur ein Teil der Software ist Freie Software ("Open Core" / "Enterprise Edition" / versteckte proprietäre Komponenten / blobs)
- Freie Software wird unangekündigt proprietär
- Freie Client-Software benötigt proprietäre Server-Software
- Neue Features sind zunächst proprietär, Produkt wird aber als Freie Software beworben
- Nutzende werden in der Ausübung ihrer Freiheiten behindert
- AI: Trainingsdaten, Modell, Gewichtung und/oder Software proprietär

Warum ist Openwashing ein Problem?

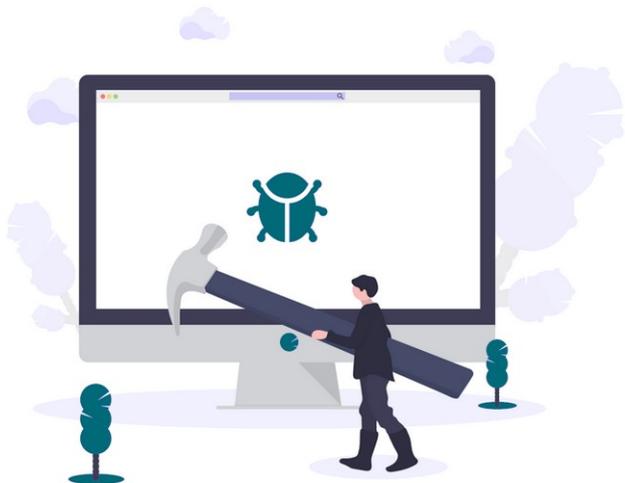
- Täuschung der Öffentlichkeit über den nachhaltigen Einsatz von Steuermitteln
- Unlautere Konkurrenz für Freie Software.
- Wer Freie Software entwickelt, wird nicht entlohnt.
- Imageschaden für Freie Software
- Verfehlen von Souveränitäts- und Nachhaltigkeitszielen: keine Kontrolle digitaler Infrastruktur, Herstellerabhängigkeit ohne Wechselmöglichkeit, mangelnde Interoperabilität

Warum ist Openwashing ein Problem?

- Täuschung der Öffentlichkeit über den nachhaltigen Einsatz von Steuermitteln
- Unlautere Konkurrenz für Freie Software.
- Wer Freie Software entwickelt, wird nicht entlohnt.
- Imageschaden für Freie Software
- Verfehlen von Souveränitäts- und Nachhaltigkeitszielen: keine Kontrolle digitaler Infrastruktur, Herstellerabhängigkeit ohne Wechselmöglichkeit, mangelnde Interoperabilität

Das Ziel „Public Money? Public Code!“ wird untergraben.

Maslows Hammer



„If the only you have is a hammer, it is tempting to treat everything as if it were a nail.“

Abraham Harold Maslow, *The Psychology of Science: A Reconnaissance*, 1966, S. 15.

**Openwashing ist nicht der einzige Fallstrick bei der Vergabe Freier Software.
Nicht jedes problematische Verhalten einer Bieterpartei hat mit Openwashing zu tun.**





PUBLIC MONEY
PUBLIC CODE

**Freie Software anbieten, die von Dritten entwickelt wird,
ohne zur Entwicklung und Maintenance (upstream) beizutragen.**

Methoden & Strategien

- Nutzung eines kompletten Produkts oder einzelner Komponenten
- Vertrieb ggf. unter anderem Namen, “[Originalprodukt] inside”, „powered by [Originalprodukt]“ o.ä.
- Dumping-Preise im Vergabeverfahren
- Kein eigenes Issue Management
- Kein (guter) Support, Kunden werden an Hersteller verwiesen

Warum ist das ein Problem?

- Reseller: Marktvorteil durch Verkauf der Arbeit Dritter, ohne zurückzugeben
- Hersteller: kein Zuschlag für eigenes Produkt, kein Profit von Vermarktung
- Hersteller: zusätzliche Arbeit/Kosten (Maintenance, Support, Issue Management)
- Verbesserungen fließen nicht upstream zurück, es profitiert nur der Reseller
- Fehlende Erfahrung, Netzwerke und Fachkräfte, fehlendes Interesse an Freier Software beim Reseller – Imageschaden für Hersteller und Freie Software

Warum ist das ein Problem?

- Reseller: Marktvorteil durch Verkauf der Arbeit Dritter, ohne zurückzugeben
- Hersteller: kein Zuschlag für eigenes Produkt, kein Profit von Vermarktung
- Hersteller: zusätzliche Arbeit/Kosten (Maintenance, Support, Issue Management)
- Verbesserungen fließen nicht upstream zurück, es profitiert nur der Reseller
- Fehlende Erfahrung, Netzwerke und Fachkräfte, fehlendes Interesse an Freier Software beim Reseller – Imageschaden für Hersteller und Freie Software

Der Allgemeinheit werden positive Effekte von „Public Money? Public Code!“ vorenthalten.

Das Freie-Software-Ökosystem nimmt Schaden.

Weitere Fallstricke

- Fehlende Community / Single Vendor
- Schlechtes Geschäftsmodell
- Negative Track Record
- Abhängigkeit von einzelnen anfälligen Komponenten



Weitere Fallstricke

- Fehlende Community / Single Vendor
- Schlechtes Geschäftsmodell
- Negative Track Record
- Abhängigkeit von einzelnen anfälligen Komponenten

Höhere Ausfall- und Lock-In-Risiken

Steuergeld wird nicht nachhaltig verwendet
Kontrollierbarkeit digitaler Infrastrukturen eingeschränkt
Schaden für das Freie-Software-Ökosystem



Challenge

Ziel:

Unabhängigkeit und Wechselfähigkeit

Freie-Software-Ökosystem erhalten

Herausforderung:

Lernen, Fallstricke bei der Vergabe Freier Software zu erkennen und zu vermeiden.

Vergaben so gestalten, dass problematische Entwicklungen berücksichtigt und prognostiziert werden können.

Geld ist nicht alles – gute Vergabe braucht eine erweiterte Bewertungsmatrix:

Preis + Leistung + weitere Kriterien

- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?



Was tun andere?

Schweiz: Praxisleitfaden OSS in der Bundesverwaltung (2019)

Zukunft einer Software realistisch vorhersagen anhand von

- Entwicklercommunity: Aktivität und Heterogenität (Abhängigkeiten)
- Codebasis
- Organisationsgrad
- Sicherheit und Vulnerabilität

Was tun andere?

Schweden: Rahmenvereinbarung „Öppna programvaror 2010“

- Nicht nur Lieferung, sondern auch Services (Installation, Integration, Anpassungen, Migration, Support, Maintenance, Training)
- Ausschreibungskriterien (Nachweise):
 - Bietende sollen nur Freie Software anbieten
 - Engagement Freie-Software-Gemeinschaft, aktive Beiträge
 - Fachwissen über rechtliche Auswirkungen
 - Erfahrung in der Schulung von Nutzenden und Technikpersonal

Was tun andere?

Italien: Guidelines Agenzia per l'Italia Digitale (AGID)

- Maintenance-Nachweis
- Eigenbetrieb möglich
- Prüfung: Abhängigkeit von anderer Software, Vereinbarungen mit Dritten
- Indikatoren zur Prognose (Repository-Aktivität, Release History, Community-Aktivität, Anzahl der Entwickler ...)

- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?



Tools

Health-/Nachhaltigkeits-Prognose durch quantifizierbare Kriterien, z.B.:

- **OpenSSF**

FLOSS Best Practice Criteria

OpenSSF Scorecard

help open source maintainers improve their security best practices and to help open source consumers judge whether their dependencies are safe

- **CHAOSS**

Metrics + Metric Models

create metrics, metrics models, and software to better understand open source community health on a global scale

CHAOSS Metric Models

- Community Activity
- Collaboration Development Index
- Community Service and Support
- Funding
- Starter Project Health
- Project Engagement
- Development Responsiveness
- Project Viability:
- Compliance + Security
- Community
- Governance
- Strategy

Beispiel: Community Activity (CHAOSS)

Wie aktiv wird eine Freie Software entwickelt?

Metriken:

• Contributors

- *contributor_count*: Anzahl aktiver Commit-/Issue-Autoren, Review-/Issue-Kommentar-Teilnehmer (90 Tage)
- *maintainer_count*: Ø Maintainer pro Repository

• Code Change Commits

- *commit_frequency*: Ø Commits pro Woche (90 Tage)

• Activity Dates and Times

- *updated_since*: Ø Monate seit letztem Update pro Repository
- *created_since*: Ø Monate seit Erstellung pro Repository
- *comment_frequency*: Ø Comments pro Issue (90 Tage)
- *updated_issues_count*: Issue-Updates (90 Tage)
- *recent_releases_count*: Releases (Vorjahr)
- *meeting_count*: Meetings (90 Tage)
- *meeting_attendee_count*: Ø Teilnehmer der Meetings

• Contribution Attribution

- *org_count*: Anzahl unterschiedlicher Organisationen, zu denen Contributors gehören

• Change Request Reviews

- *code_review_count*: Ø Review-Kommentare je Pull Request (90 Tage)

• Issues Closed

- *closed_issues_count*: Geschlossene Issues (90 Tage)

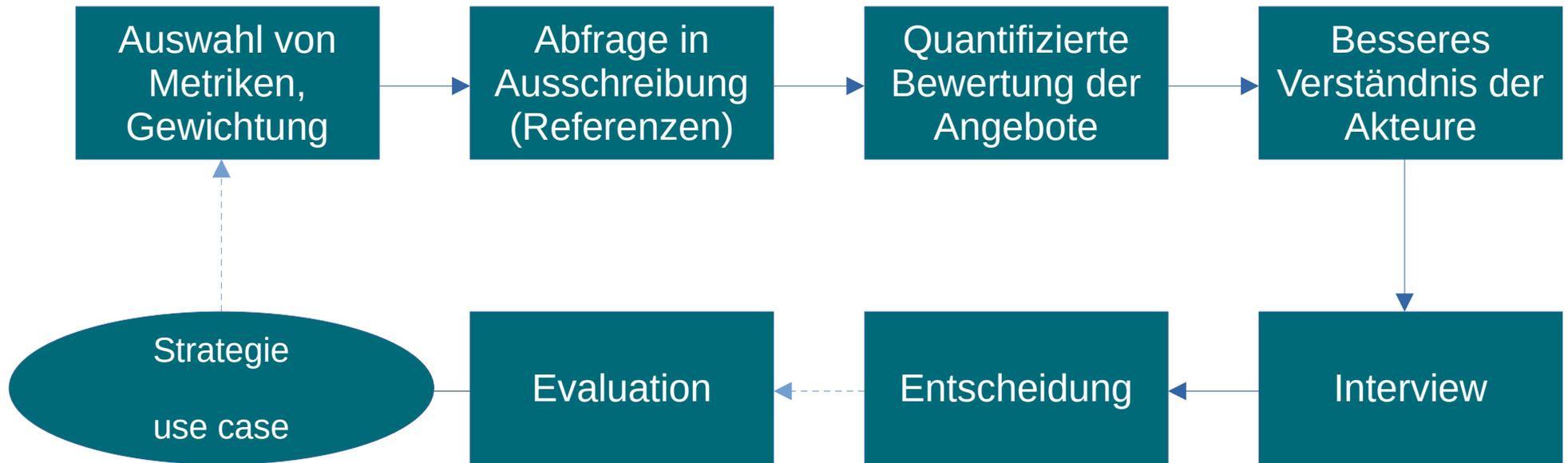
<https://chaoss.community/kb/metrics-model-community-activity/>

- 1 Von wo nach wo geht die Reise?
- 2 Welche Fallstricke gibt es?
- 3 Wie versuchen andere sie zu vermeiden?
- 4 Welche Tools können wir nutzen?
- 5 Wie wenden wir sie an?



Operationalisierung

Wie lassen sich diese Metriken operationalisieren, um die Fähigkeiten, das Involvement und das Commitment von Organisationen hinsichtlich einer Freien Software beurteilen und prognostizieren zu können?



Kriterienauswahl

Aktivitäten und Fähigkeiten von Contributors

- Code schreiben / reviewen, Bug Triaging, QA/Testing, Security, UI/UX, Barrierefreiheit, Troubleshooting/Support, andere referenzierte Freie-Software-Aktivität)
- Zertifikat oder Empfehlung durch Top Contributor der angebotenen Software / des Herstellers
- Dokumentation, Bildungsaktivitäten, Community-Building
- PA-Aktivitäten

Kriterienauswahl

Aktivitäten des Anbieters

- Open Standards
- Track Record Freie Software
- Single Vendor
- Track Record Openwashing / Wechsel zu proprietärer Lizenz

Kriterienauswahl

Openwashing-Prognose/Monitoring anhand von Indikatoren, z.B.

- nicht paketiert für wichtige Distributionen
- Lizenz nicht von FSF/OSI/Debian anerkannt, fehlende Lizenzangaben
- closed development
- Single Vendor
- CLA-Verpflichtung
- Veröffentlichung unter Freie-Software-Lizenz angekündigt

Herausforderungen

- Wer entwickelt Kriterien? Monitoring?
- Wie wird gewichtet? Muss- vs. Soll-Kriterien?
- Unerwünschte Wirkung: Abschreckung durch Komplexität
- Erwünschte Wirkung:
Ausschluss problematischer Akteure und Produkte
Anreiz an Hersteller, die Kriterien zu erfüllen

Unabhängigkeit, Transparenz

Herausforderungen

- Kompetenzen aufbauen
(Freie-Software-Ökosystem / problematisches Verhalten)
- Kompetenzen institutionalisieren
- Wissen in Verwaltungen transferieren
- Fehler- und Lernkultur

Los geht's!



Los geht's!

<https://fsfe.org/donate>



Vielen Dank!

Quellen:

DSC 2018-05-23 Pitfall: CC BY-NC-ND 3.0 theEyZmaster

<https://www.deviantart.com/theeyzmaster/art/DSC-2018-05-23-Pitfall-746337946>

Icon Art: <https://undraw.co>

Ada & Zangemann Illustration: CC-BY-SA Sandra Brandstätter, Matthias Kirschner