

FrOSCon 2024

Making Open Source education an integral part of the curriculum in Computer Science programs

Prof. Dr. Daniel Kulesz

2024-08-17

Prof. Dr. Daniel Kulesz

- Professor for Software Engineering at Bingen Technical University of Applied Sciences



Photographer: Andreas Ebinger



Prof. Dr. Daniel Kulesz

- Professor for Software Engineering at Bingen Technical University of Applied Sciences
- Previous educational experience:
 - Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
 - University of Stuttgart
 - Stuttgart University of Applied Sciences (HfT)
 - Baden-Wuerttemberg Cooperative State University (DHBW) Stuttgart



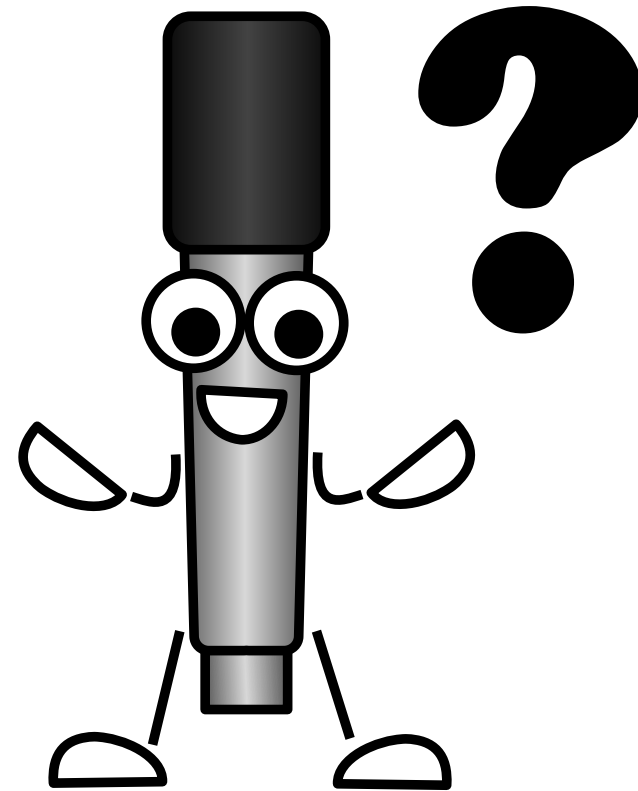
Photographer: Andreas Ebinger

Warning: Personal Reflection

- Talk based on my personal experience
- Be prepared for claims with lack of evidence
- Proper empirical research required, cooperations welcome

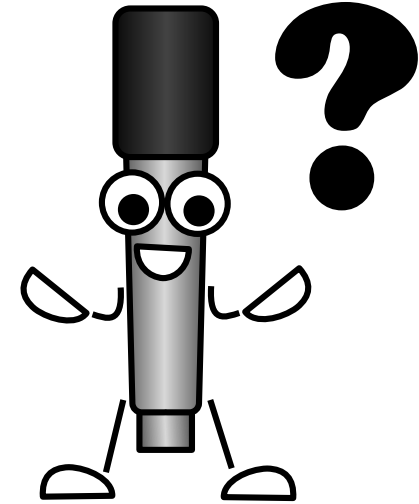


A few questions for you ...



A few questions for you ...

- Who knows a person that is studying computer science or a closely related subject?
- Who knows a person that is studying computer science or a closely related subject but lacks fundamental understanding about F(L)OSS¹?
- Who knows a person that has successfully completed their degree in computer science or a closely related subject but still lacks fundamental understanding about F(L)OSS?



1: F(L)OSS = Free (libre) open source software

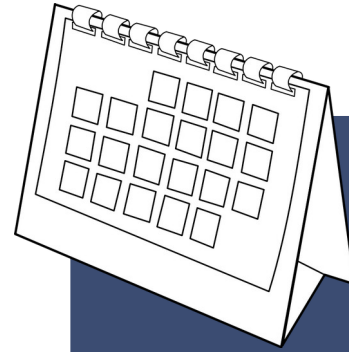


Learning Objectives

- You can explain why fanboyism/fangirlism can impair important business decisions about software.
- You are less shocked when observing the phenomenon that computer science people lack F(L)OSS understanding.
- You are familiar with proposals that can be integrated in computer science curricula to raise this understanding.
- You absorbed some of my experience from F(L)OSS projects in student settings and are aware of challenges that can occur.

Agenda

- ☒ Motivation
- ☐ Fanboyism/Fangirlism for Proprietary Software
- ☐ Causes
- ☐ Countermeasures
- ☐ Lighthouse Projects
- ☐ Challenges
- ☐ Summary



Any fanboys/fangirls in here?

- Soccer clubs?
- Beverages?
- Laptops?
- Cars?
- Proprietary Software?

Example Case

- July 2021: Teacher association demands use of proprietary services
- Some „demanders“ are:
 - Teachers
 - Pupils
 - Parents
 - ...
- Have they all been paid by the vendor?

Source: <https://www.rnd.de/digital/lehrerverband-schulen-sollten-weiterhin-microsoft-und-co-nutzen-duerfen-HHJQX5FZOYLREIQGEFJHZ3F7G4.html>, last accessed 2024-08-16, 03:31pm

Startseite > Digital > Lehrerverband: Schulen sollten weiterhin Microsoft und Co. nutzen dürfen

Lehrerverband: Schulen sollten weiterhin Microsoft und Co. nutzen dürfen



Viele Schulen nutzen während der Pandemie Microsoft Teams. Datenschutzbeauftragte hatte sich im vergangenen Jahr allerdings gegen den Einsatz ausgesprochen, weil die Server in den USA stehen. Der Deutsche Lehrerverband kritisiert den Schritt, jetzt ein bewährtes Arbeitsmittel zu verbieten.

Unlikely! They are just fanboys/fangirls!

- Interestingly, people spend time and money for the sake of someone else's profits!
- Definitions taken from the „Wiktionary“ [1]:
 - Fanboy: „A male fan who is *obsessive* about a particular subject (especially, something or someone in popular entertainment media).“
 - Fanboyism: „Behaviour characteristic of fanboys; blind, aggressive devotion.“

[1] <https://en.wiktionary.org/w/index.php?title=fanboy&oldid=75888256>

Effects of Fanboyism/Fangirlism

- Fanboyism/fangirlism can lead to undesirable effects (often limited to the affected people and their social environment).
- Sure - it's the people's personal choice what music they listen to, which soccer club they cheer for etc...
- However, sometimes fanboys/fangirls are allowed to make decisions that also affect *others*:
 - Which cars to buy for the car fleet?
 - Which hardware (e.g. laptops) to buy for the company?
 - Which software (services) to use for purpose X?

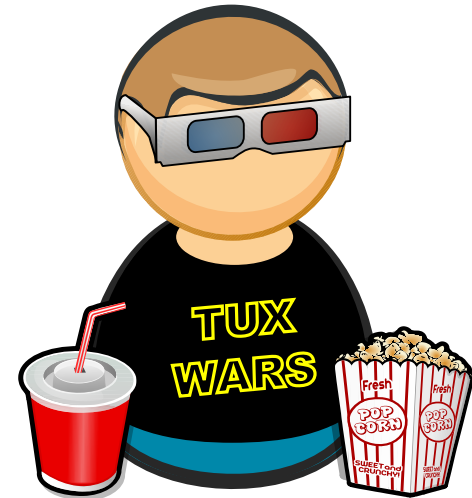


What's wrong with fanboys/fangirls?

- Nothing - as long as they stay in their own world and their own (walled) garden!
- But does anyone really expect fanboys/fangirls to make unbiased decisions, only based on facts?
- Decisions fanboys/fangirls make for others:
 - Personally okay'ish, if you would make the same decisions
 - Pretty bad if you don't agree with the decisions made
- One area where this becomes apparent is software and services.

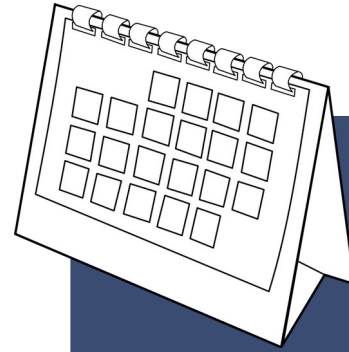
Biased software/service decisions

- Narrowed focus on functional aspects and user experience
- Fanboys/fangirls often claim to take costs into account as well.
- However, they often „overlook“ crucial cost categories:
 - Security (and its Auditability)
 - Maintainability
 - Digital sovereignty
 - Data Privacy
 - Vendor Lock-in
 - ...



Agenda

- ☒ Motivation
- ☒ Fanboyism/Fangirlism for Proprietary Software
- ☐ Causes
- ☐ Countermeasures
- ☐ Lighthouse Projects
- ☐ Challenges
- ☐ Summary



Lack of Knowledge in Practice



Many people are puzzled about F(L)OSS

- Yes, we all know that not all politicians have sufficient knowledge about free and open source software.
- However, from my observations, not only computer science students but also IT professionals ...
 - can't name the four essential freedoms
 - can't explain the copyleft principle correctly
 - think that F(L)OSS is mostly produced by unemployed hobbyists
 - argue that F(L)OSS is only an option for the poor



What could be the reasons?

#1: Topic is not covered explicitly

- From my observations, the topic of F(L)OSS is not mentioned explicitly in many teaching plans or module handbooks.
 - Do lecturers take an understanding about F(L)OSS for granted?
 - Do lecturers consider it irrelevant?
- I don't know!

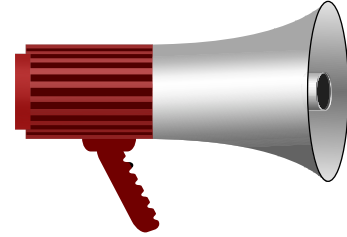


#2: Teachers don't understand it either

- I have met many lecturers (even professors), who do not seem to understand F(L)OSS at all.
- Some seem highly biased and even tell their students things like „open source can never work“ or „you can use open source for your hobby but only consider serious software for business“.



#3 Indoctrination



- Works great due to lecturers' laziness
- Free training, e.g.: chip vendor providing free three-day seminar with qualified lecturers about parallel programming (using the vendor's proprietary tools, running only on hardware certified by the vendor, ...)
- Free teaching material (slides, exercises, exam questions, ...)
- Free or heavily discounted educational licenses
- Free services for educational institutions (e.g. SaaS offerings)
- ...

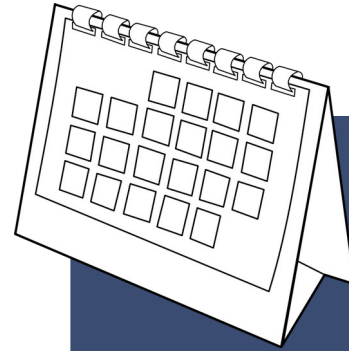
#4 Market dominance



- Once a product reaches a significant market share or even dominates the market, teachers are tempted to teach the use of this specific product in their courses
- Underlying conflict: Staying independent vs. qualifying for the job
- Extreme cases: Lecturers offering product-specific courses and asking product-specific questions in the exams
- ...

Agenda

- ☒ Motivation
- ☒ Fanboyism/Fangirlism for Proprietary Software
- ☒ Causes
- ☐ Countermeasures
- ☐ Lighthouse Projects
- ☐ Challenges
- ☐ Summary

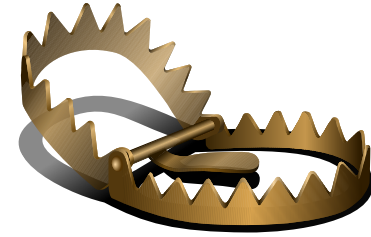


What to achieve?

- The students of today are the decision makers of tomorrow!
- As a society, we want these decision makers to make well-informed decisions in a professional fashion.
- The demand for professionalism excludes fanboyism/fangirlism:
 - Students must understand implications of both F(L)OSS and proprietary software.
 - If you think you are a fanboy/fangirl yourself, make this conflict transparent.

Any particular recommendations?

#1 Avoid tool/service pitfalls



- Focus on teaching methods and approaches/principles, not tools
- Use tools only for deepening the understanding in exercises, not for the sake of teaching the tools themselves
- Stay independent from any tool or service vendors
- Avoid tools that only run on one operating system
- Don't force your students to accept malicious license agreements or to share data with third parties against their will
- Use self-hostable tools, avoid third-party services

#2 Teach F(L)OSS Basics

- Public Domain und Licensing (to the degree legally possible)
- Business models for proprietary software and common limitations
- History of free and open source software
- Open Source Licenses (to the degree legally possible)
- Practical steps for publishing F(L)OSS
- Business Models around F(L)OSS

F(L)OSS Business Models?

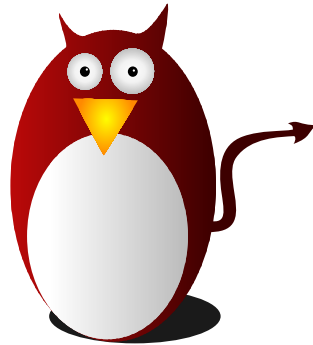


#2 Teach F(L)OSS Basics - cont'd

- Donations
- Selling Media
- Selling Hardware
- Charge for binary builds
- Keep tests closed source
- Open Core
- Partnerships
- Certified Partners
- Dual Licensing (especially in combination with Copyleft)
- Services (Support, Training, Consulting, Certification)
- Hosting (paid or supported by advertisements)
- Customizing
- Contract work (especially for customers from government)

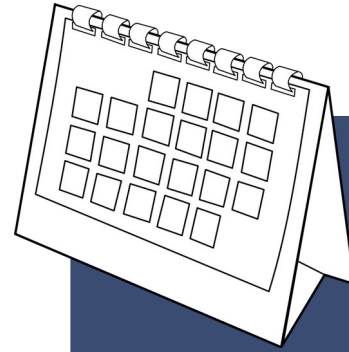
#3 Encouragement and Enablement

- Encourage students and help them to publish their work but don't force them in compulsory modules
- Use open sourced work of previous students in exercises for students that follow (→ „look, you can earn glory and pride!“)
- When offering projects and theses, state your expectations regarding the publication of the results
- Say no to projects which help closed ecosystems grow.
- Evil „pro trick“: Offer theses extending existing work that is subject to a license with strong copyleft ;-)



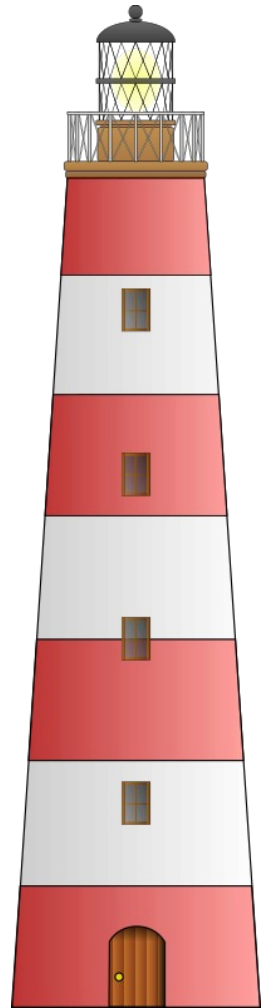
Agenda

- ☒ Motivation
- ☒ Fanboyism/Fangirlism for Proprietary Software
- ☒ Causes
- ☒ Countermeasures
- ☐ Lighthouse Projects
- ☐ Challenges
- ☐ Summary



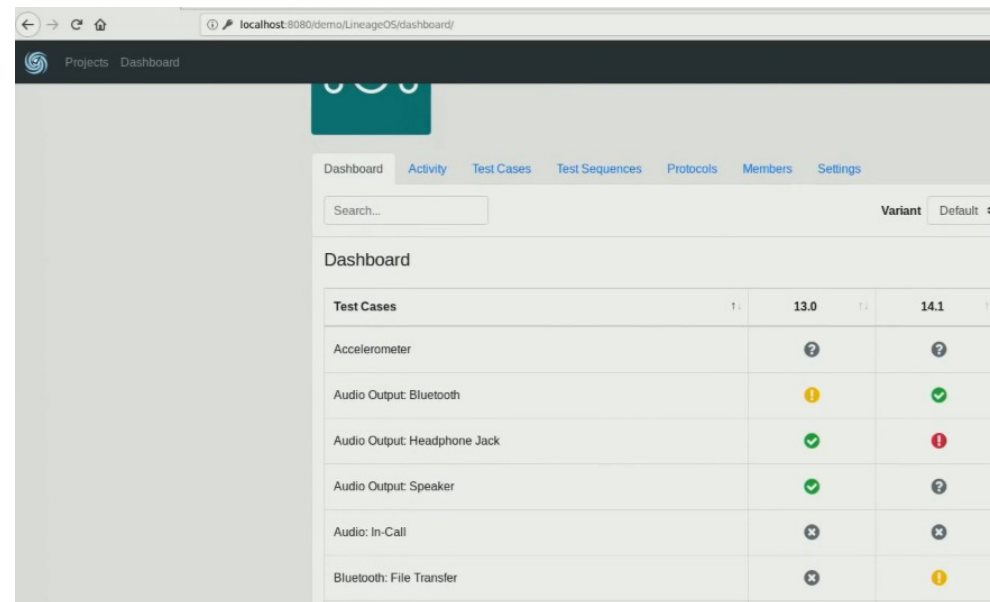
Light House Projects

- I supervised dozens of student software projects.
- Most of them have been published as F(L)OSS.
- In the following, I can only show a few selected examples – many more would have deserved to be mentioned as well!



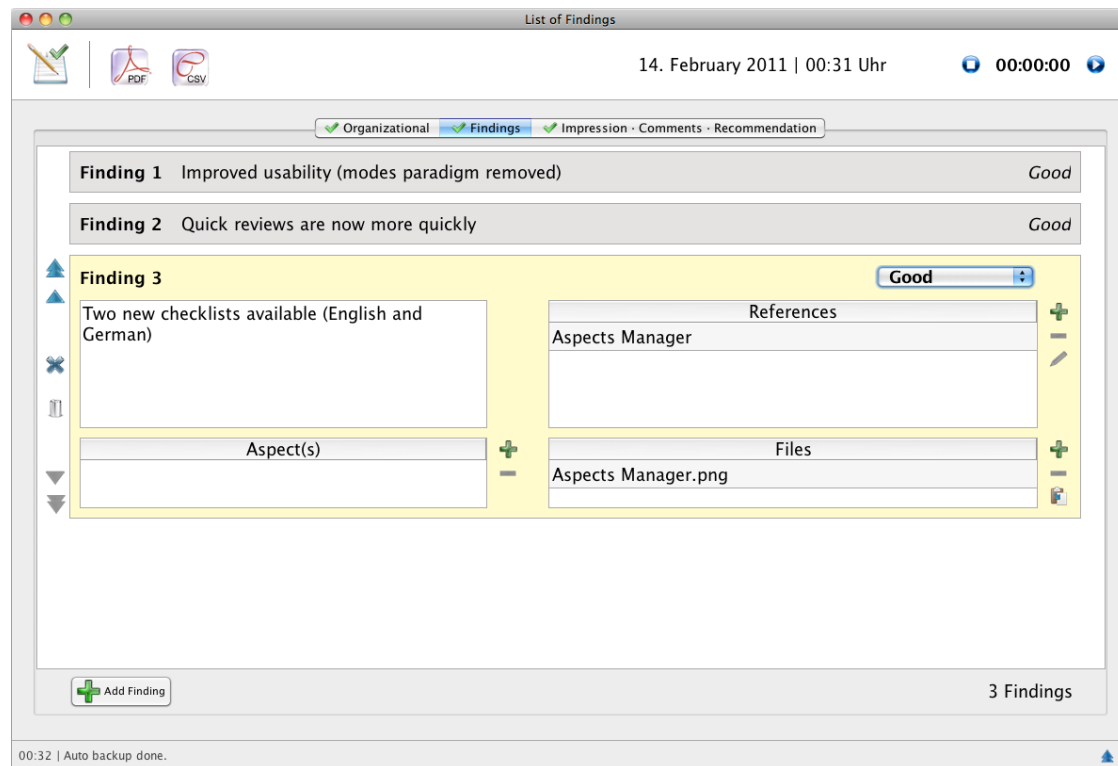
SystemTestPortal

- Plan, organize and document manual software tests
- Initially written in Golang, porting effort to Rust in progress
- Presented at various conferences, e.g. FrOSCon 2019.
- <https://www.systemtestportal.org/>



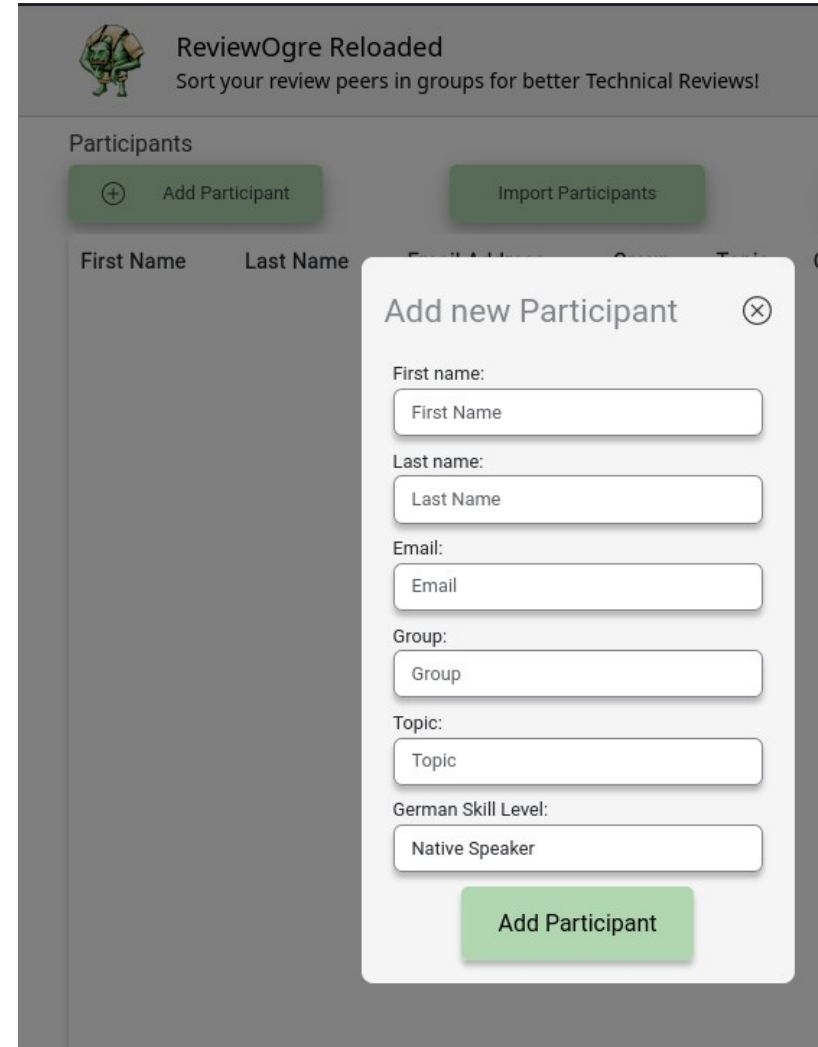
RevAger

- Plan and execute Technical Reviews
- Written in Java (Swing UI) back in 2009
- <https://revager-org.github.io/>



ReviewOgre Reloaded

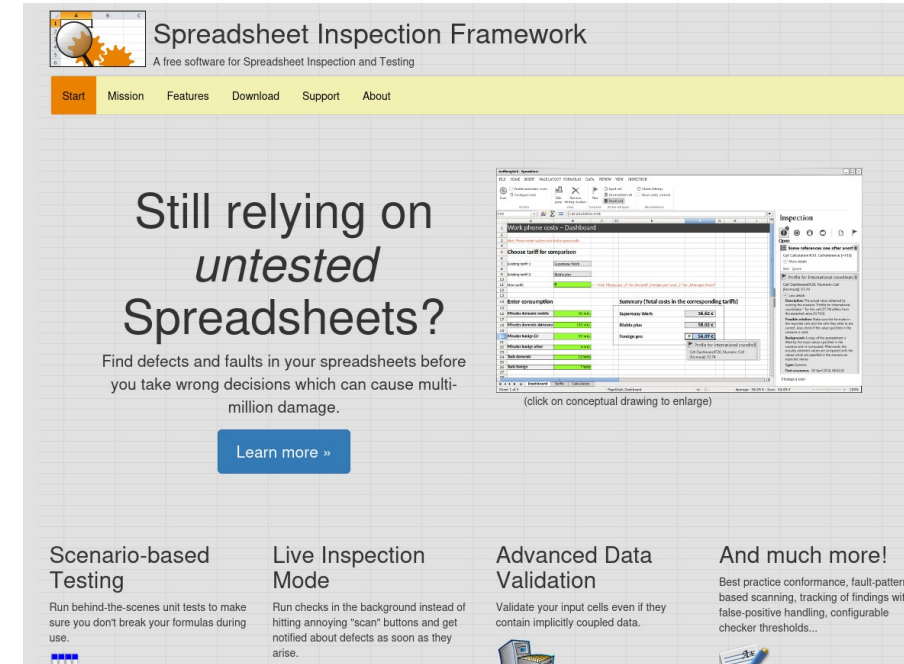
- Organize Technical Review sessions by computing review teams (primarily for use by lecturers)
- Written as ProgressiveWebApp
- <https://github.com/RichardWuensch/ReviewOgre>



The screenshot displays the 'ReviewOgre Reloaded' web application. At the top, there is a header with a logo of a green ogre and the text 'ReviewOgre Reloaded' and 'Sort your review peers in groups for better Technical Reviews!'. Below the header, there is a 'Participants' section with two buttons: 'Add Participant' (with a plus icon) and 'Import Participants'. A modal titled 'Add new Participant' is open, featuring a close button (X) in the top right corner. The modal contains several input fields: 'First name:' with a placeholder 'First Name', 'Last name:' with a placeholder 'Last Name', 'Email:' with a placeholder 'Email', 'Group:' with a placeholder 'Group', 'Topic:' with a placeholder 'Topic', and 'German Skill Level:' with a placeholder 'Native Speaker'. At the bottom of the modal is a green 'Add Participant' button.

Spreadsheet Inspection Framework

- Apply various techniques to conduct systematic and continuous testing of spreadsheets
- Main research prototype for my dissertation, result of 12 student theses over several years
- Backend written in Java, Frontend written in C#
- <https://www.spreadsheet-inspection.org>



freeLABorga

- Management for university labs (inventory, lend/return items, ...)
- Frontend: Web application with HTML, CSS and vanilla JS
- Backend: Rust
- <https://github.com/freeLABorga/freeLABorga>

Labor-Verwaltung

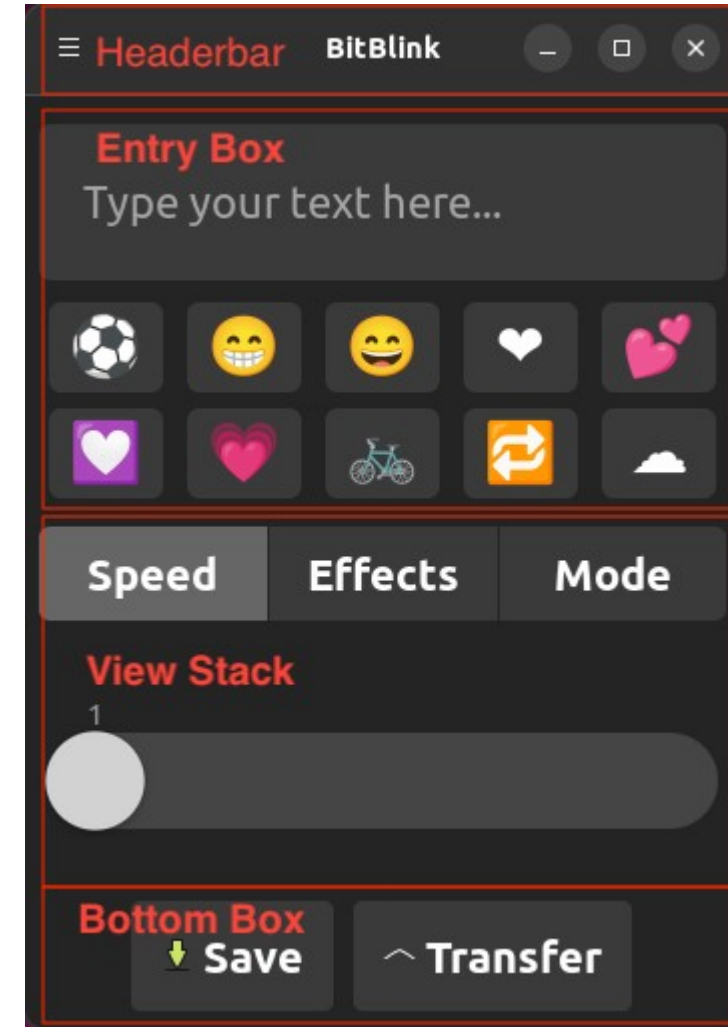
Inventar Heatmap Personen Admin-Bereich

Laborgegenstände Übersicht

Name	Kategorien	Anzahl	Verfügbar
Bangle.js	Android Schwarz Smart-Watch	2	1
DELLE SE2422H	PC-Monitor Schwarz	2	2
ESP32 NodeMCU	ESP Microcontroller WiFi	1	1
Espruino Pico	Espruino Microcontroller	3	2
HW 470 G7	HW Laptop Windows 10	3	2
Mojo G4 Play	Android Schwarz Smartphone	4	3
Monitor	PC-Monitor	1	1
NAO V6	Roboter Weiß	1	0
NAO V7	Roboter Weiß	1	1
NEP P5540 Projektor	Beamer Weiß	1	1
Pinephone 1	FLOSS Pinephone Smartphone Weiß	1	0
Pinephone pro	FLOSS Pinephone Schwarz Smartphone	1	1
Raspberry Pi 3, 1 GB RAM	Einplatinencomputer Raspberry Pi	1	1
Raspberry Pi 4, 8 GB RAM	Einplatinencomputer Raspberry Pi	2	1
Sharp PN-61TW3A	Touch-Bildschirm	1	1
Samsung Galaxy Tab A8	Android Grau Samsung Tablet	1	1
Sunny HDR-CX240E	Videokamera	1	0
Venolo IdeaPad 5	Grau Laptop Ubuntu Venolo	1	1

bit_blink

- Program LED name badges via bluetooth
- Written in Rust with GTK4 and libadwaita
- https://github.com/JoGehring/bit_blink



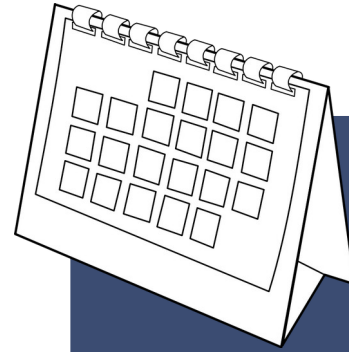
aidGer

- Tool for faculty-wide management of student asisstants including contracts, accounting etc.
- Written in Java
- Used for several years by the faculty on a daily basis
- <https://github.com/aidGer/aidGer>

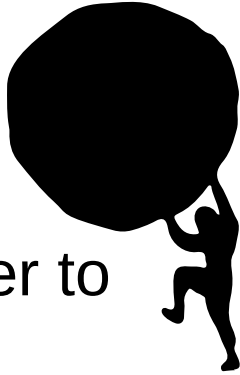


Agenda

- ☒ Motivation
- ☒ Fanboyism/Fangirlism for Proprietary Software
- ☒ Causes
- ☒ Countermeasures
- ☒ Lighthouse Projects
- ☐ Challenges
- ☐ Summary

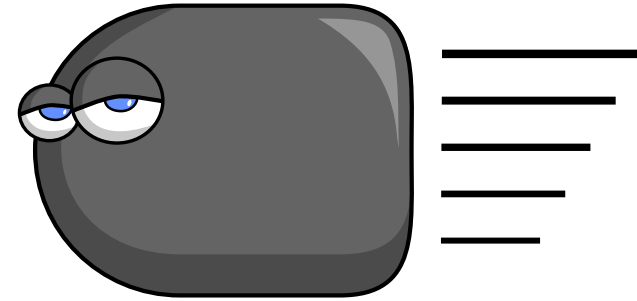


Challenges



- Unlike other F(L)OSS projects, work done by students in order to obtain a grade is likely to be abandoned once the project has been graded.
- Possible solutions:
 - Organize follow-up projects (con: lecturer becomes main driver)
 - Make students create something that is so useful that a community emerges and motivates the students to continue their work or forks the work
 - Give students more freedom regarding the topic of their project – building something the students can also use themselves might help (but has cons regarding didactics).

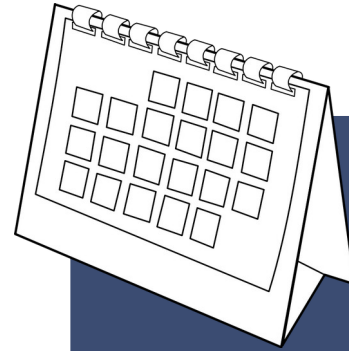
No silver bullet



- I tried all of the aforementioned suggestions.
- Unfortunately, while I succeeded in making students create great F(L)OSS products that have been used for several years, I was not successful in establishing a long-term maintainership that works without my activity.
- I am sure some of you did better (no matter whether on the student or lecturer side)!
- I warmly welcome your suggestions!

Agenda

- ☒ Motivation
- ☒ Fanboyism/Fangirlism for Proprietary Software
- ☒ Causes
- ☒ Countermeasures
- ☒ Lighthouse Projects
- ☒ Challenges
- ☐ Summary



Verdict

- The students of today are the decision-makers of tomorrow!
- One-sided Fanboyism/fangirlism must be avoided.
- Open source education should be part of every computer science curriculum, so students understand how it works and can make informed decisions later on.
- Actively encouraging and helping students with publishing their work gives them at least partial F(L)OSS experience.
- Student projects are likely to suffer from lack of maintenance. Silver bullets welcome!

Thank you!



Foto: Andreas Ebinger

Prof. Dr. Daniel Kulesz

Follow me in the Fediverse: @kuleszdl@fosstodon.org