

Foss North 2016

This is my notes in a simple list form and pictures from the Foss-north conference held in Gothenburg 2016-05-26.

The meetup is arranged by the non-profit organization [Foss North](#) which is a joint venture between [Foss GBG](#) and [Foss Sthlm](#) to create a larger meetup than the regular bi-monthly/quarterly meetings that Foss GBG arranges here in Gothenburg.

In general the meetup was pretty good, well arranged, good speakers with a broad variation on topics, all the way from hardware design and low-level linux kernel stuff to user interface design, licensing and software patents. More breaks would have been welcome though. A tight schedule with many speakers can be pretty exhausting when the areas are interesting.

At the bottom of this post you can find some pictures from the event, mostly on presentations. I'll add a link to the presentations when they are available.

Attendees from op5: [Unknown User \(chelmertz\)](#), [Unknown User \(jekstrand\)](#) and me, [Fredrik Mikker](#)

We also met Anders from Ericsson there, a regular on the Dev-beer at the Gothenburg office there.

Amy Sanford - We're all entrepreneurs

AES Consulting

Keywords: Dare, innovate, inspire, entrepreneurship

- The startup of your extraordinary disruption: Digitization, Innovation and the Tech Entrepreneur
- Disruption - the fourth industrial revolution
- How to awake the inner entrepreneur. We're all entrepreneurs.
- Be open to new things, fail forward: Dare to make mistakes and fail

Youtube: The start-up of you: rediscovering the entrepreneurial spirit in all of us. Book/Youtube: No ordinary disruption - #NoOrdinaryDisruption
Youtube: Inside Sweden's silicon Valley

Reflections: I'm not the target audience, but the ideas about the fourth industrial revolution and that the traditional hubs of innovation is behind Mumbai and some other new tech-hubs around the world.

Lewis Horne - Opensource electric car

Uniti - teamuniti.com

Keywords: Inspire, dare, software, openness, think different, automotive

- Why start over, and think different?
- No heritage, no compability issues. No ordinary steering wheel mechanics, steer-by-wire?
- The average trip by car in Sweden is 1.2 KM. Why drive a 2 ton tank when you can use a more suitable transportation for short distances.
- Not aiming for maximum profit but rather to make a new, more suited car and improving the environment at the same time.
- Aurdino is used heavily for prototyping, and Uniti ARC is the in-house electronics that will control the vehicle.
- Open hardware with open specifications, and open source code: Uniti linux distro, A VM for testing, Documentation and source on Github
- The powertrain is from Bosch.

Reflections: Great idea about a new non-traditional electric car, but he's really a sales guy.. I'd love a test drive though. Only available via Oculus Rift so far.

Adam Dunkels - Make your software eat the world

Thingsquare, Contiki open source OS for embedded devices. - <http://www.contiki-os.org/> / <http://www.thingsquare.com/>

Keywords: IoT, embedded

- Hardware is getting cheap, which makes the traditional industry transform to software development. The same goes for clock cycles that

has skyrocketed the last decade.

1. Open source software
2. ??????
3. Profit!

- Find the problem you want to solve, and don't just go for profit. Create good software and the value will come.
- Throwing VC/cash on a company that doesn't have a well defined product doesn't end well.

Reflections: Talks a bit like a sales guy, looks like Joe MacMillan in Halt and catch fire. But a decent speaker about a area I don't care that much about for the time being. Codes a lot for being a CTO.

Mirko Boehm - Protecting open source through patent pooling and defensive publications

Open invention network

Keywords: Free software, software patents, collaboration

- Patents are not applicable to open source software, you don't sit in your basement and invent something by your self, and wait 2 years until the patent has gone through the patent office, you will be behind in the industry before you've started. It's a collaborative age where people do things together, so how should we address this in the world of open source?
- Open source and patents does not add up. Often a large portion of the software stack is derivative work.

- You don't need to have a patent to join the network, but what you can do is to benefit from Red Hat, Google etc that they won't sue you over software patents. Any company that works with open source can benefit from this and stand on the shoulders of giants if a lawsuit is filed.
- It only covers the open source parts of the software stack, not proprietary parts.
- It's free, as in beer.

- Defensive publications - Linuxdefenders.org Protection from patent trolls. Declare that this is a known innovation and it can't be patented.

Outstanding questions that I need to look up:

- Linux system definition?
- European patent office is granting software patents?

Reflections: Quite interesting, if you don't like software patents. I don't, so, yeah. But it contained a lot of legal terminology that isn't that easy to get at a first glance.

Alessandro Rubini - Time in software and hardware

Consultant/CERN/fsfe

Running linux since before 1.0

- All automatic systems need the concept of time, especially a particle accelerator.
- But not everything is simple in the field
- The human representations of time is perverse, according to Alessandro :)
- The act of measuring time is taking time!

Reflections: Low level linux kernel and hardware stuff, not my cup of tea. He stuttered, and had a heavy italian accent which made it hard to understand everything - but a funny guy that actually made the talk good anyway.

Jonas Öberg - State of free software - Legal and policy issues in Europe

(And elsewhere)

Free software foundation Europe

Keywords: Free software, licenses, database of signatures

- Sony TV's contains FOSS, such as almost all other vendors of tv's, set-top boxes, smartphones etc.
- Software provenience - Where does the software originate? The term comes from the art community.
- If you want to know where software in your device comes from, you need a database.
- Black duck created one for their purposes.
- 2015 - 78% of software used is free and open, according to Black Duck software. (Hello Peter).

We had a demo with Peter Andersson that introduced op5 to Black Duck's database of signatures to find out where the works/software originates from.

Other ones that are free?

Reflections: No slides and a lot of legal stuff again.

Anders Arnholm - Software Craftmanship

HiQ

Keywords: Collaborate, learning, be proud of your work

- Why are you here? Think about it.
- Care about you craft and be proud of your work and make stuff that you can be proud of. Criticise your self, it's healthy. Ask or read if you don't know. Nobody knows everything, but everybody knows something. Collaborate and learn, and make learning a habit.
- Don't live with broken windows, if you see something that's broken - fix it! Blame is counterproductive, improve stuff instead.
- Sign your work that you have done, and be proud of it. It can be anything that helps a project.

My reflections: Lots of good ideas about thinking of the work you do as a craft, like a great mason, carpenter etc.

Alexandra Leisse - Death by a Thousand paper cuts

Ardoq

Keywords: UX, Design, Complexity

- Ease of use, is complexity a problem?
- Overly complex products leave users overwhelmed if they don't understand what to expect from a function. People want to be in control.
- Is there a way to make complex things simple?
 - Probably not, creating a simple app that can preform complex tasks is hard, but we could make them clear and understandable instead.
- You need to make tradeoffs, all the time.
- One example is a coffee maker with one button, it makes the coffee black, in a cup. But sometimes I want milk, espresso, smaller cup etc. Ends up with a lot of buttons. Limit the amount of choices. What tradeoffs should we make to have the best coffeemaker?
- Mental model: What you think about a function can become the truth, since your knowledge is limited. You think you know how something should work, but that's often not the case.
- Data model: Code is clear and beautiful as per design from a backend coders perspective. But can demand complex functions higher up in the stack to deliver the functionality to the user.
- Try to think from the users perspective to solve the problem, through the whole application to reduce complexity in other layers of the stack.
- Focus on the problem, worry about corner cases later. Deliver a product that is understandable and intuitive.
- Think big, ship small.

Outstanding questions:

- Hicks Law?

Reflections: From my novice eyes, she seems to have a hang of UX/Design, and made the talk really interesting.

Aarne Ranta - Grammatical Framework

Digital grammars

Keywords: Translation, localization, Fortran

The talk was about a opensource translation framework that is developed by a community of PhD's in linguistics. An offline application for translation. Isn't based on statistics of a large dataset as google translate, bing etc.

- Small in terms of application size. The application is ~30mb.
- Built for producer tasks, such as translating single words in a large dataset, not really consumer tasks such as complex scentences, but moving forward in that area too.

Reflections: Complex stuff, not really that interesting from my point of view, but a good talker. Finnish has a lot of words for beer..

Daniel Stenberg - Everyone, everywhere runs this code all the time

[Mozilla/curl/haxx.se](https://mozilla/curl/haxx.se)

Why CURL? Because the internet doesn't follow specs and needs a tool that can talk via multiple protocols.

Curl is a 18 years old project, it started with httpget, a few hundred lines of code from a Brazilian developer that only could do HTTP. Became cURL in 1998 due to implementatation and support for more protocols.

The name is from clientURL or c(language)URL, made it up along the way.

- Curl is everywhere, in all the things - Tvs, Phones, Cars etc. ~1 billion active devices approx. A software that eats the world, it's everywhere.
- One part of this is that it's not GPL, and can be included everywhere (MIT Licensed).
- Paid contributions for features occurs. Need x-feature, how much, how long?
- Why open source? It would have died in the 90s if it wouldn't been opensource.

How to write software that ends up being used by millions of users?

- Write lots of code
- Provide useful options and aipis
- Rinse, repeat.
- Repeat until it succeeds

Does it ever get done?

- No, it truly never gets done
- Protocols keep evolving
- Open source code survives (if done correctly, craftsmanship, docs etc)
- No slow-down in sight
- You can help!
- 185 command line switches

Reflections: A good speaker that has been in the FOSS-world for ages, a lot of fun anecdotes.

Foss-North 2016

There are no images attached to this page.