Logiciel libre: économie et impact

De la théorie à la pratique

Roberto Di Cosmo

Software Heritage - Inria - University Paris Diderot
roberto@dicosmo.org
www.dicosmo.org

13 Mars 2018

Qui je suis

professeur d'Informatique, chercheur (Erdos #: 3)

30 ans de recherche, *20 ans* de contribution au Logiciel Libre

1998 *Hold up planétaire* – vulgarisation enjeux sociétaux de l'informatique

1999 *DemoLinux* – première distro live GNU/Linux

2007 *GTLL Systematic* 150 members 40 projects 200Me

2010 *IRILL* `www.irill.org`

2015 *Software Heritage* `www.softwareheritage.org`

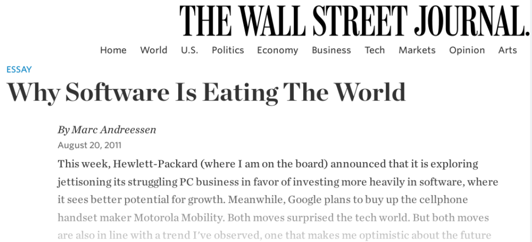
Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Software is everywhere

Software is eating the world...

Business



THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Arts

ESSAY

Why Software Is Eating The World

By Marc Andreessen

August 20, 2011

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future

Software companies

outperform or buy out

hardware companies

Marc Andreesen, 2011

Technology

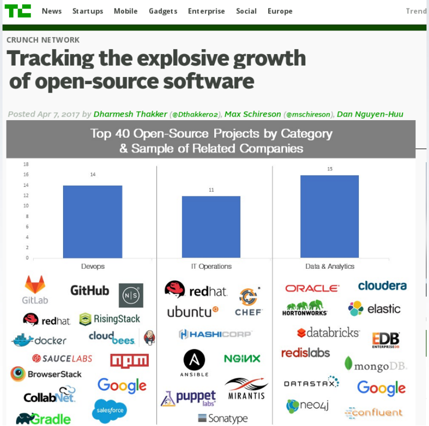
Software Defined Everything

Hardware gets commoditised

Software becomes the new value!

Worldwide Software Defined Everything (SDE) Market to grow at a CAGR of 27.9% over the period 2016–2022 to aggregate $143.35 billion by 2022

. . . Open Source is eating the Software World



CRUNCH NETWORK

Tracking the explosive growth of open-source software

Posted Apr 7, 2017 by Dharmesh Thakker (@Dthakkero2), Max Schireson (@mschireson), Dan Nguyen-Huu

Top 40 Open-Source Projects by Category & Sample of Related Companies

Open Source Software

can be openly (re)used, modified, (re)distributed, *with full access to its source code!*

Free Software

Free Software

Software that offers to *its users* the freedom to:

0. use the software
1. study and adapt the software
2. distribute software copies
3. distribute modified copies

Why bother?
Free Software has changed the way software is:

- developed
- tested
- deployed
- maintained
- marketed
- sold
- designed
- taught
- ...

Free Software: "la rançon de la gloire"

Going mainstream...

Today, everybody loves Free Software, even ancient opponents

"Microsoft loves Linux"
Satya Nadella, October 2014

... is not an easy journey

Myths, misunderstandings, hype, ... are all around us.

Let's dispel some of this.

Myths surrounding free software

Anarchic development ("Bazaar", "Wisdom of software crowds")

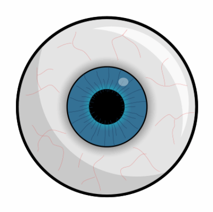
Software is a technical object.

A mass of random coders does not create beautiful software.

Software Quality and Free Software

> *With enough eyballs, all bugs are shallow*
> — *Eric Raymond*

That's a *logical implication*!
You need enough eyeballs first.



Myths surrounding free software

The community will take care of it

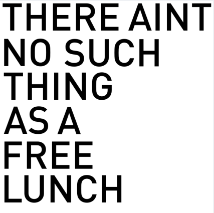
Making software available is *necessary*.

But it is not sufficient to create a community that curates it.

Free software and cost

Creating and maintaining beautiful software has a cost that must be paid for.

The fact that you do not pay for a software licence is a detail.



THERE AINT
NO SUCH
THING
AS A
FREE
LUNCH

Understanding Software Economy

Back to the basics

economics: the study of how society chooses to allocate *scarce* resources to produce, exchange, and consume goods and services.

Ruffin, Gregory, "Principles of Economics", 1990

Without *scarcity*, there is no *economy*.

45 years of proprietary software economy

Started in 1969, with the *IBM Unbundling of software and services*.
Based on the artificial "scarcity" of *verbatim copies* of an *existing* piece of software!

This usually followed a standard *push* approach:

- identify a market
- develop a "one size fits all" software solution
- sell licences to a lot of users, . . . if you can
- make sure the users will need new versions, *often*. . .

Economy of free software?

free software …

removes the “scarcity” of copies

Vast literature on “economic models of free software”

- GNU Manifesto, early vision of Richard Stallman, circa 1985
- Chris Hecker: Setting up shop, 2000
- John Koenig: Seven open source business strategies for competitive advantage, 2004
- Gasperoni, Comar: Open Source in Dependable Systems
- Livre blanc Aful, 2007
- Livre blanc April, 2007
- Livre bleu du GTLL, 2015…

Common starting point: selling “licences” is gone; one needs *other scarce resources*

Economy of free software!

And yet it moves!

In France, for 2017, the market for free software was *over 4.5 billion euros!*

Free software exposes the *truly scarce* resources

- know-how
- commit rights
- community connection
- infrastructure
- process, industrialization
- customization, qualification

Looking for a (free) software business model?

Start by looking for a resource that is *scarce*

and valuable to *a group of users*

Building a *successful* FOSS project

Martin Michlmayr (former Debian project leader) studied successful FOSS projects (see `http://opensource.mit.edu`).

They all show a similar pattern of evolution.

Cathedral phase		Transition phase		Bazaar phase
Original “idea” Project Author Core developers Unix philosophy	⇒	”Interest” Prototype *Modular design*	⇒	Distributed development environment Community Parallel perfective and corrective maintenance Peer reviews

The transition does not come for free!

Basic principles for free software success

In other words

- identify a need
- develop a software prototype
- build a community
- set up an *ecosystem*, with:
  - *users*
  - developers
  - architects
  - service providers...

all working together, and playing by the rules

The first two phases are the less difficult to get right.

The challenge is in the second two.

A few success stories for R&D into FOSS

Focus on publicly funded R&D projects

Quite different from the usual FOSS success stories:

- research dimension (long to medium term)
- high technology focus
- transfer from Academia
- public funding through grants
- users are *too often* an afterthought

The Hilite success story

Project info: `http://www.open-do.org/projects/hi-lite/`

Goals: Formal methods tooling for *high-integrity* software.

Funding: 1.4Me French funding, over 4.1Me project cost

Duration: 3 years (may 2010/may 2013)

Cluster: Free Software thematic group (GTLL) in Paris

Project partners

Leader: AdaCore (SME)

Academia: CEA-LIST, Inria

Industry:

- Altran
- Astrium Space Transportation
- Thales Communications

The Hilite success story

Scientific and technical Results

SPARK 2014 : new version of high integrity Ada

Why 3 : new version of the proof platform

E-ACSL : new annotation language for C

Adoption, Community, Business

embedded.com : *Next-generation of SPARK static verification toolset released*, Bernard Cole, May 2014

lists/forge : 69 members, thousands of mails exchanged

collaborations : joint AdaCore/Inria lab, CNAM and Kansas State University, Mitsubishi Electric...

SPARK Pro 15 : professional edition, with new clients and upgrades of old clients

The Hilite success story

Key success factors : consortium

leadership : active *editor* of a Free Software solution, *SME*

academia : strong partners with *development* background

users : big companies onboard *are real potential users*

The focus was on *the product*, from the start.

Key success factors : community

insiders : core community *inside* the project from the start

academia : partnerships established through conferences and collaboration

outsiders : precise focus on the industry sector that *uses* the technology

The *community* does not need to be *large*...

...it must be *pertinent* and *active*

The Squash success story

Project info: http://www.squashtest.org/

Goals: Unified approach to Functional Testing

Funding: 1.3Me French funding, over 3Me project cost

Duration: 2 years (march 2011/june 2013)

Cluster: Free Software thematic group (GTLL) in Paris

Project partners

Leader: Henix (SME)

Academia:

- University Paris 8
- Loria

Industry:

- GDF Suez
- Kalis

The Squash success story

Scientific and technical Results

Two new OSS products

Squash TM : test management

Squash TA : test automation

Adoption, Community, Business

downloads : more than 1.000 downloads per month, including many big companies

user base : large international market (RTBF, for example...)

contributions : no contribution good enough to deserve inclusion, but...

service : enabler for a healthy service activity that ensured a real, full Free Software editor strategy (no freemium/open-core, etc.)

The Squash success story

Key success factors : consortium

leadership : active *editor* of an Free Software solution, *SME*

users : big companies onboard *are real potential users*

The focus was on *the product*, from the start.

Key success factors : community

insiders : core community *inside* the project from the start

outsiders : a healthy community of *users* of the technology, despite no real community of contributors to the code
maintained through traditional marketing
Again: access to the code is not enough!

There is not necessarily an *external community* of developers

Conclusion

Lessons learned

users a key success factor

- must be *in* the project *from the start*
- a *business unit* from a large company may give more impact than its R&D department

community necessary to ensure sustainability

- must be *in* the project *from the start*
- may be a *developer* or a *user* community, or both

leadership is essential

- coordination by a free software *editor* is a definite plus

Questions ?