Economic Aspects of Libre Software

Master on Libre Software (URJC)

http://master.libresoft.es

Jesus M. Gonzalez-Barahona, Felipe Ortega

jgb@gsyc.es jfelipe@libresoft.es
@jgbarah @felipe GSyC/LibreSoft, Universidad Rey Juan Carlos

November 2012

FLOSS business models

The million dollar question

Once upon a time, the key question for companies/projects/individuals entering the FLOSS universe was:

- *How can we make money with FLOSS?*

The million dollar question

Once upon a time, the key question for companies/projects/individuals entering the FLOSS universe was:

- *How can we make money with FLOSS?*

But now, the question is

- *Which are the most successful business strategies that we can adopt around FLOSS?*

Origin: open innovation

- Some (few) people are still against the added-value provided by knowledge aperture (source code is just an example).
- Case study: **GoldCorp** `http://www.goldcorp.com/`
    - Traditional mining company, needs to find new veins.
    - Executive President Rob McEwen publishes open contest (2000): 575K$ for participants who offer best methods to choose new extraction points.
    - 80 % of answers produced positive results.
    - 100$ invested in 1993 is worth over 3.000$ in 2005.
- However, **crowdsourcing is orthogonal to libre software**

Current trends

- "*There is no silver bullet*"
- Every company/project/individual should select appropriate strategies for their particular needs.
- And being alert to change/evolve their model on the fly, in response to rapid changes in the market and surrounding conditions.
- Luckly, there exist **many** options to choose from.

False myths

- Many companies justify the cost of proprietary software with the economic cost of production (development) process.
- Two types of product value:
  - Use value: Economic value as a tool, productivity growth (value as an intermediate good).
  - Sale value: Value as a market product (value as a final product).
- Aprox. 75 % of salaried programmers work is devoted to maintenance tasks, not to development of new features.

False myths

- Contrary to other products, the true value of software does not lie in its production or replacement cost (food, cars, engines, appliances...).
- The upper limit of the value of software for clients is imposed by the expected value of the future service that sellers offer to clients.
    - Software stability.
    - Help-desk, support services.
    - Documentation.
    - Software customization.
    - Training and certification.

False myths

- Contrary to other products, the true value of software does not lie in its production or replacement cost (food, cars, engines, appliances...).
- The upper limit of the value of software for clients is imposed by the expected value of the future service that sellers offer to clients.
  - Software stability.
  - Help-desk, support services.
  - Documentation.
  - Software customization.
  - Training and certification.

services == source of revenues

False myths

- The organizational model of libre software allows to provide services in a more profitable, scalable and sustainable way.
  - Users become clients.
  - Scalability in failures identification and solution.
  - Sharing risks and production costs.
  - Monopolistic practices made difficult.

False myths

- Myth: "The Commons tragedy".
- Any business model would be doomed to fail, or else, to result in a new scenario with closed solutions.
- Antidote: In fact software use does not reduce but it increments its value.
    - Larger user community (potential clients).
    - Benefit from patches and proposed solutions.
    - Benefit from new features.
    - Contributing to leverage product quality (guarantee absence of conflicts with business interests).

"Inverse commons"

Central idea

The business model must always pivot on the use value, not on the sale value.

FLOSS: A guide for SMEs

FLOSS by itself is not, and it has never been, a business model.

Companies must desing a strategy, create a business plan and ensure securing benefits aimed to a sustainable growth.

FLOSS can improve viability and efficiency with many business models.

FLOSS: A guide for SMEs

- Report ellaborated for FLOSSMETRICS, EU FP6 project.
- Analysis of 218 companies receiving at least 25 % of their total revenues directly or indirectly from FLOSS.
- Identifying common business strategies around FLOSS.
- Recommendations about required conditions to apply each model.

3 axes influencing software landscape

- Software model
  - Proprietary vs. libre software.
- Development model.
  - Barriers to collaboration.
  - Single developer/reduced group vs. large community, global outreach.
- Business model.
  - Type of revenues model.
  - Numerous options: Training, support, on-demand changes, productizing, SaaS, etc.

Strategic uses of libre software

- The Gartner consulting group pointed out: in 2012, use of libre software would reach 90 % of companies worldwide.
- Companies forced to adapt their software to work under multiple conditions.
- Emerging trend in big companies towards source code release under FLOSS licenses. More active interaction with nearby projects.
- Proprietary software success is now guaranteed only if there are no alternative and reliable FLOSS options.

Carlo Daffara taxonomy (FLOSSMETRICS)

- *Dual licensing*: FLOSS version and proprietary version.
- *Open core*: Allows mixing FLOSS and proprietary elements.
- *Product specialists*: Superior knowledge, additional services.
- *Platform providers*: Integration, product testing.
- *Aggregate support providers*: Primer nivel de soporte para diferentes tipos de software libre.

Carlo Daffara taxonomy (FLOSSMETRICS)

- *Selection/consulting companies*: Closer to the analyst role, minimum impact on FLOSS communities.
- *Legal certification and consulting*: Assessment on license compatibility.
- *Training and documentation*: Either as part of a broader support contract or companies exclusively devoted to this market area.
- *R&D cost sharing*: Initial investment + creating community to reduce R&D costs.
- *Indirect revenues*: Baseline for sales of associated products or services (commodities).

Dual (or multiple) licensing

- Examples: MySQL (GPL+prop.), Mozilla software (MPL, GPL, LGPL).
- The owner of source code IPR selects a double licensing strategy.
  - **Copyleft license**: To obtain some benefits (community contribution, integration in other libre software, etc.).
  - **Proprietary license of permissive free license**: Other companies might integrate this software with proprietary software.
- Leverage adoption and visibility, offering solutions to proprietary software companies.
- But it may also limit contributions from FLOSS developers (we force them to release the ownership of the code to maintain the right of dual licensing).

Open core

- Also based on dual licensing, but this is not the business foundation.
- Examples: SugarCRM, SendMail.
- Different features linked to licenses:
    - **Libre core**: Basic version of the software, including minimal working functinalities, is libre software.
    - **Proprietary added-value functionalities**: Most sophisticated features, targeting the enterprise sector and expert audience, are released under proprietary license.
- Revenues concentrate on commercialization of proprietary components, plus support and extension/customization.
- FLOSS developers might distrust the project, for the same reasons as dual licensing.

Product specialists

- Company offering or creating maintainance for a specific software product, usually in a concrete market.
- Example: Alfresco (Content Management System).
- Main revenues come from training, consulting, specialized support for installation, deployment and configuration, extensions, customization, etc.
- Typical model "best product knowledge here" or "best source code here".
- It is mandatory to retain the best team of specialists.
- High risk of other competitors entering at the same level of knowledge and specialization in the same market.

Platform providers

- Selection, support, integration and services around a set of projects integrated in a single, tested and verified product.
- Key points:
  - Verification process to meet quality standards.
  - Additional services (support, training, customization).
- Copyright ownership prevents direct copy (not cloning).
- Example: RedHat.
- The source code is libre software, but the product name and logo are trademarks.
- Considerable effort to eliminate them from files, documentation, etc.

Aggregate support providers

- Centralized support point for different libre software products.
- Two strategies:
  - Employing developers specialized in those products.
  - Redirecting petitions to other support specialists.
- Clear benefit for large projects whose costs could raise due to excessive diversification of support channels (comprehensive help-desk).
- It only works for clients with many different products running on mission-critical platforms. They assure exclusive support contracts.
- Example: OpenLogic.

Selection/consulting companies

- Usually, they do not develop software.
- Assessment about selection and evaluation of software products and services.
- It might save costs for clients (software selection process can add up to 45 % of total costs of software adoption).
- Experience to undertake selection process is not easily replicable.
- Sometimes, they develop dedicated tools aimed for selection, that they do not release as free software to stop competence.
- Example: Open WebApps.

Legal certification/consulting

- They do not usually develop software, either.
- Legal assessment for companies about complex problems derived from multiplicity of libre software licenses and their compatibilities.
- They might save costs for integrators less familiarized with selection process and licensing checking.
- Example: Palamida.

Training and documentation

- Focused on important aspects of libre software, but not directly related to development.
  - Training users and tech staff.
  - Support for certification processes (technical staff and end-users).
  - Creation of high quality documentation.
- It is very helpful if supported products have a large user base (size of target market).
- Certification programs might provide extra income, conditions for collaboration agreements (partnerships).
- Example: Gbdirect.

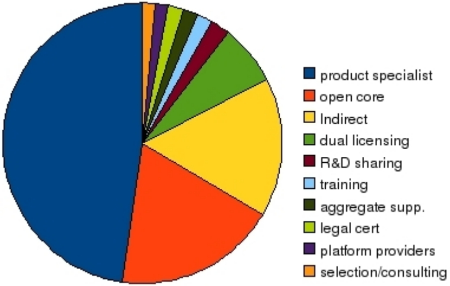
R&D cost sharing

- Company interested in software development decides to release the code to benefit from contributions from large community of interested developers.
- Development cost of closed source alternative would be prohibitive.
- Viability conditions:
  - Great branding and public image to attract attention from stakeholders.
  - Solvency to develop and nurture community participation.
  - Investment in project support (IDE, events, promotion, etc.).
- Example: Maemo (Nokia).

Indirect revenues

- Companies that decide to develop libre software or funding libre software projects to leverage their business on related products.
- Some examples:
  - Hardware makers: Intel, Dell. Broaden their target audience base providing better support for libre software platforms.
  - Tech books publishers: O'Reilly (contributes to Perl community, and it is the main publisher of Perl books).
- Stakeholders that indirectly benefit from success of libre software projects.

Distribution of models identified in the study



Evolution CAGR in FLOSS business

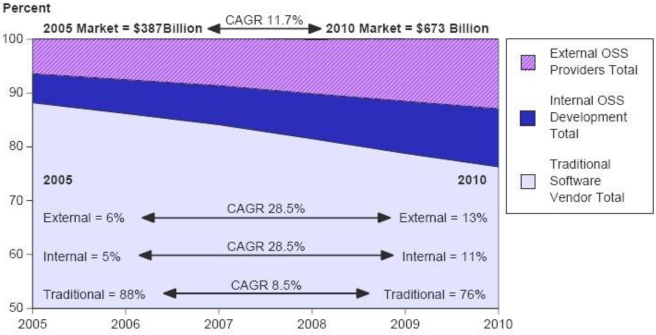


Figura: Taken from "Open Source going mainstream", Gartner group report.

CAGR: Compound Annual Growth Rate

Companies running business around FLOSS (selection)

Figura: Taken from “FLOSS: A guide for SMEs”, Carlo Daffara (FLOSSMETRICS EU FP6)

New ways of collaboration

- Libre software can be adopted as a mean to facilitate new ways of collaboration among different companies. *Coopetition*.
- Several possibilities (collaboration without formal contracts or agreements).
- Creating business ecosystems.
- Empowering business opportunities.
- Inspiration for new applications of information sharing.

References

- "FLOSS: A guide for SMEs".
  http://guide.conecta.it/index.php/Main_Page
- "The Magic Cauldron", Eric Raymond.
- "State of Open Source", Gartner Group, 2008.
  http://arstechnica.com/open-source/news/2008/02/gartner-80-percent-of-commercial-software-programs-will-include-open-source-by-2012.ars
- The rejuvenetion of community controlled open source.
  http://blogs.the451group.com/opensource/2009/10/15/the-rejuvenation-of-community-controlled-open-source/