Open source business models: myths, realities, practical examples

Carlo Daffara
European Working Group on Libre Software
Conecta Research
“Open-source software is deliberately developed outside of market mechanisms... fails to contribute to the creation of value in development, as opposed to the commercial software market ... does not generate profit, income, jobs or taxes ... The open-source licenses on the software aim to suppress any ownership claims to the software and prevent prices from being established for it. In the end, the developed software cannot be used to generate profit.” University of Muenster, Muenster Institute for Computational Economics “Open Source-Software: An Economic Assessment”
… yet OSS represents 25% of the total sw+services market
Where does this overall value comes from?

Table 24: The software economy: sales, services and in-house.

<table>
<thead>
<tr>
<th>Region</th>
<th>Proprietary software licenses</th>
<th>Software services (development and customisation)</th>
<th>Internal development</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU-15</td>
<td>19%</td>
<td>52%</td>
<td>29%</td>
</tr>
<tr>
<td>U.S.</td>
<td>16%</td>
<td>41%</td>
<td>43%</td>
</tr>
<tr>
<td>Japan</td>
<td>N/A</td>
<td>N/A</td>
<td>32%</td>
</tr>
</tbody>
</table>

Source: FISTERA thematic network.

- firms contributing code to FLOSS projects have in total at least 570 thousand employees and annual revenue of 263 billion € (FLOSSPOLS study, 2005)
- Adapted to 2008: 890 thousands employees, 315 B€ total annual revenue
<table>
<thead>
<tr>
<th>OSS Vendor Business model</th>
<th>Dual Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor example</td>
<td>MySQL</td>
</tr>
<tr>
<td>Number of covered products</td>
<td>single or few</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Dissemination for the product with reduced costs, creation of external ecosystem of add-ons (outside the source), visibility, self-segmentation of the market</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>The adopter may opt for the open source edition if it is deemed sufficient; for the proprietary part, reduction in cost may give better price/quality ratio</td>
</tr>
<tr>
<td>Potential disadvantages of the model</td>
<td>Low external participation (limited code contributions)</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Integration of the product with non-OSS components in externally distributed products</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>license choice</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>Limited (very little external contributions, mainly debug activity and external products)</td>
</tr>
<tr>
<td><strong>OSS Vendor Business model</strong></td>
<td><strong>Open Core</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>Zimbra</td>
</tr>
<tr>
<td>Number of covered products</td>
<td>single or few</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Reduction of R&amp;D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of add-ons, self-segmentation of the market for the proprietary add-ons</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>The adopter may opt for the open source edition if it is deemed sufficient; for the proprietary part, reduction in cost may give better price/quality ratio</td>
</tr>
<tr>
<td>Potential disadvantages of the model</td>
<td>Difficult to estimate the right balance between open and closed parts, external groups may create substitutes for the proprietary parts</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Need for the proprietary additions or need of support</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>license choice, segmentation on features</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>Potentially large, depending on the balance open/proprietary</td>
</tr>
</tbody>
</table>
Open Core is actually a combination of different models:

- Open source modules
- Proprietary modules
- Dually-licensed wrapper
- Open source components

So, we can infer from our research that the wrapper will receive little or no contributions from the outside (no moral judgment – a practical one..)
<table>
<thead>
<tr>
<th>OSS Vendor Business model</th>
<th>Product Specialist</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vendor example</strong></td>
<td>Acquia, OpenCascade</td>
</tr>
<tr>
<td><strong>Number of covered products</strong></td>
<td>single or few</td>
</tr>
<tr>
<td><strong>Economic advantage for the vendor</strong></td>
<td>Reduction of R&amp;D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of add-ons</td>
</tr>
<tr>
<td><strong>Economic advantage for the adopter</strong></td>
<td>Reduction in cost may give better price/quality ratio for the adopted software, stability, integrated support reduces external costs</td>
</tr>
<tr>
<td><strong>Potential disadvantages of the model</strong></td>
<td>Low barrier of entry for third-parties</td>
</tr>
<tr>
<td><strong>Sale condition</strong></td>
<td>Need of support or specialized modifications, lack of internal expertise</td>
</tr>
<tr>
<td><strong>Freeriding protection</strong></td>
<td>License choice, segmentation on features</td>
</tr>
<tr>
<td><strong>External ecosystem</strong></td>
<td>Yes</td>
</tr>
<tr>
<td>OSS Vendor Business model</td>
<td>Platform Providers</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>RedHat, Novell (partially)</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>many</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Reduction of R&amp;D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of software and additions</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>Reduction in cost may give better price/quality ratio for the adopted software, stability, integrated support reduces external costs, legal protection is included, easy to find trained personnel, availability of long-term options</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>Platform engineering requires large R&amp;D efforts even with shared resources</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Value perceived by user must be higher than the cost of going to an unsupported recompilation (e.g. CentOS); usually mission-critical environments, need of support or lack of internal expertise</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>License choice, copyrighted and trademarked elements included in the product</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>Yes</td>
</tr>
<tr>
<td>OSS Vendor Business model</td>
<td>Software Selection</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>Navica</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>many</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Cost of software certification and selection can be partially shared across customers, as most adopters have a large share of common needs</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>Reduced selection costs; reduced risk of wrong choice</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>Limited market, difficulty in following rapid evolution of the products covered (evaluation costs)</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Complex requirements, many areas or strict vertical requirements to match, possibly large company size</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>Selection documents are usually proprietary; selection requires human intervention (non-replicable)</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>No</td>
</tr>
<tr>
<td><strong>OSS Vendor Business model</strong></td>
<td><strong>Aggregate support providers</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>OpenLogic</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>many</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Cost of support can be partially shared across customers, economies of scale</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>A single point of control and cost for a large number of project, reduced negotiation efforts for large number of individual vendors, simplified management and governance</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>Limited market, may be perceived as in partial competition with existing specialists</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Large number of managed projects, use in mission-critical infrastructure</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>Inherent in the non-transferability of support contracts</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>No</td>
</tr>
<tr>
<td><strong>OSS Vendor Business model</strong></td>
<td><strong>Legal certification and insurance</strong></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>Palamida, Black Duck</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>many</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Cost of legal certification and secondary-level insurance can be shared across the most used OSS projects</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>Equivalent to insurance; provides a materialized and stable costs against uncertain, difficult to quantify negative events</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>Limited market, difficult to estimate risk probabilities, need to cover separate legal frameworks across the world with different rules</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Potential legal risk</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>Inherent in the non-transferability of certification and insurance contracts</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>No</td>
</tr>
<tr>
<td>OSS Vendor Business model</td>
<td>Indirect Revenues</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>Mozilla, most hardware suppliers</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>Single or few</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Source availability reduces engineering costs and increase visibility on multiple platforms</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>Adopters obtains a quality product at no cost; potential large ecosystem for extensions</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>Requires a large external market for incentives, may be dependent on a single (or small number) of actors increasing risk</td>
</tr>
<tr>
<td>Sale condition</td>
<td>There should be an external source of revenue linked to adoption (eg. Ecommerce sales of related products, search engine back-payments, etc.) Usually linked to high adoption numbers</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>license choice, copyrighted and trademarked elements included in the product</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>OSS Vendor Business model</strong></td>
<td><strong>Training and documentation</strong></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>GbDirect</td>
</tr>
<tr>
<td>Number of cov. prod.</td>
<td>many</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>A significant portion of training development costs can be shared across customers, economies of scale, reuse of community-developed material</td>
</tr>
<tr>
<td>Economic advantage for the adopter</td>
<td>Lower cost for training compared to self-managed training (from source code, publicly available documentation)</td>
</tr>
<tr>
<td>Potential disadv. of the model</td>
<td>May be perceived as in partial competition with existing specialists, human intensive, most of it cannot be replicated at low cost</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Lack of internal experts (or too high cost for creation of internal skills), complex configuration and setup of OSS product</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>Training material are usually non-public, trainers are inherently non-replicable</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>No</td>
</tr>
<tr>
<td>OSS Vendor Business model</td>
<td>R&amp;D cost sharing, Open Inside</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Vendor example</td>
<td>Eclipse</td>
</tr>
<tr>
<td>Number of covered products</td>
<td>single or few</td>
</tr>
<tr>
<td>Economic advantage for the vendor</td>
<td>Reduction of R&amp;D, reduced maintenance costs</td>
</tr>
<tr>
<td>(same as vendor- in this case, vendor and adopter coincide)</td>
<td></td>
</tr>
<tr>
<td>Potential disadvantages of the model</td>
<td>Establishing the management and contribution structures may be complex and costly, requires constant effort</td>
</tr>
<tr>
<td>Sale condition</td>
<td>Significant R&amp;D costs, higher than the cost of management of the shared community</td>
</tr>
<tr>
<td>Freeriding protection</td>
<td>license choice</td>
</tr>
<tr>
<td>External ecosystem</td>
<td>yes</td>
</tr>
</tbody>
</table>
Open source business models: myths, realities, practical examples
Embedded Developer Challenges
Application complexity

- Networking
  - Coffee machines, cars, factories, etc.

- Quality
  - More software, more bugs
  - Recalling embedded systems is expensive

- Multimedia/Features
  - Mobile phone (calling, camera, MP3, Gaming)

Project Lines of Code vs. Number of Embedded Developers

CAGR: 26%
CAGR: 5%

Year
Typical Lines of Code per Project
# of Embedded Developers

CODE LESS. CREATE MORE.
<table>
<thead>
<tr>
<th>Project size (lines of code)</th>
<th>% of OSS</th>
<th>total cost (Keuro)</th>
<th>Savings</th>
<th>duration (years)</th>
<th>avg. staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000</td>
<td>0</td>
<td>1703</td>
<td>0%</td>
<td>1.7</td>
<td>20.5</td>
</tr>
<tr>
<td>1000000</td>
<td>50</td>
<td>975</td>
<td>43%</td>
<td>1.3</td>
<td>15.4</td>
</tr>
<tr>
<td>1000000</td>
<td>75</td>
<td>487</td>
<td>71%</td>
<td>0.9</td>
<td>8.6</td>
</tr>
<tr>
<td>10000000</td>
<td>0</td>
<td>22000</td>
<td>0%</td>
<td>3.3</td>
<td>141.7</td>
</tr>
<tr>
<td>10000000</td>
<td>50</td>
<td>12061</td>
<td>45%</td>
<td>2.6</td>
<td>103.2</td>
</tr>
<tr>
<td>10000000</td>
<td>75</td>
<td>3012</td>
<td>86%</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>100000000</td>
<td>0</td>
<td>295955</td>
<td>0%</td>
<td>7.5</td>
<td>818</td>
</tr>
<tr>
<td>100000000</td>
<td>50</td>
<td>160596</td>
<td>46%</td>
<td>5.9</td>
<td>631.2</td>
</tr>
<tr>
<td>1000000000</td>
<td>75</td>
<td>80845</td>
<td>73%</td>
<td>3.8</td>
<td>421</td>
</tr>
</tbody>
</table>
There is no “good” or “bad” model - there are models that work and those that do not.
Models change with time, and market conditions.
To survive, you must understand what you are doing, and what can be improved; a good way to do it is using a business model schema.

In the Osterwalder model:
• Customer Segments (For whom are we creating value?)
• Value Propositions (What value do we deliver to the customer? Which one of our customer’s problems are we helping to solve?)
• Key Resources (What is essential for our work to be possible?)
• Key Activities & Key Partners
• Cost Structure
• Revenue Streams (No, giving things for free does not count)
• Channels (How we reach our customers – or potential ones)
• Customer Relationships
• The basic idea behind business models: I have something or can do something, and it is more economical to pay me to do or get this “something” instead of doing it yourself. The most important elements: **customer segments** and **value proposition**

• With Open Source, usually “property” is non-exclusive (with the exception of Open Core, where part of the code is not open at all)

• The remaining element is efficiency, that is the ability to perform an action with a lower cost (both tangible and intangible)

• **Efficiency and property are the underlying justification behind OSS business models**
• Exclusive properties are barriers that can be enforced weakly or strongly (eg. Trademarks in RHEL). Other barriers are convenience, control
• Efficiency increases monotonically except when a new technology is introduced, when effort stops efficiency decreases (also monotonically)
Work necessary for a task is the inverse of efficiency, and this provides an higher bound for revenues:

```
I'll do it myself
We have done it a few times
We do it for a living
```

Open source business models: myths, realities, practical examples
• If no one pays me, my only possibility is to invest my time to create something. In any case, I have increased my efficiency in that activity.
• I can trade this efficiency and convert it into money if the work that I am doing is repeatable to a large extent.
• .. and thus you turn into a **product specialist**
• You will be in business as long as the efficiency differential is sufficient to provide a margin for your activity (including overheads) and your research costs (since you need to continue to maintain the efficiency differential in time).
• Anyone else can become a competitor, but he will have to invest to reach your efficiency level (or he will not be able to maintain competition for long)
• If you focus on a single step of the software adoption cycle:

...you obtain as a result most of the possible business models: training, legal support, aggregate support...
• Difference between an open source company and a proprietary one: the R&D sharing
• The main efforts that can be done to improve this sharing advantage: increasing and facilitating contributions, or the “platforming” (the use of your product as a basis for something else)
• If you use property as a basis, to increase revenues the best approach is not forcing monetization (as the available budget remains zero) but customer segmentation (who can be targeted for the enterprise product? who can use the product as basis for another service?)
In fact, the approach used to “force” people to pay (for example by removing features from the community edition) will not increase monetization in a significant way, because those that are most affected are those that have no budget available; those people will turn to forks.

The company should target those non-users that could become adopters, and at the same time increase dissemination, as a fixed percentage of all users will become paying in one form or the other.

“Viral” dissemination does not work across different groups, so marketing is still essential, even with an open source approach.
Thanks!
Carlo Daffara

cdaffara@conecta.it
http://carlodaffara.conecta.it