FLOSS 2.0 ?
Some results from the CALIBRE project

Olivier BERGER (GET/INT - France)

phpGroupware Conference 2006

Paris (France), 6 November 2006

Partly Funded by the European Commission via IST Project 004337 CALIBRE
This work is licensed under a Creative Commons Attribution-ShareAlike License. http://creativecommons.org/licenses/by-sa/2.0/
Contents

• Intro
• Calibre's context
• Some results
• New characterisation of FLOSS ?
About GET & INT

• GET is a group of several public higher education schools in France:
  – teaching + research
  – field of Telecommunication and IT

• Inside GET, INT (National Institute of Telecommunications), near Paris: business school + engineering school

• Several teams specialised in research and practice on Libre Software
About me

- Research Engineer
- Software developer
- Libre software activist since 1996
- Member of the board of APRIL:
  - oldest Libre software promotion non-profit association in France (est. 1996)
  - 27 companies
  - >350 individuals
PicoLibre/PicoForge

- Web platform for collaborative software development (« forge »)
- Based on existing mature libre software :
  - phpGroupWare (web virtual desktop, general ACL infrastructure, file-manager, ...)
  - OpenLDAP (glue)
  - TWiki (project Wikis) (soon)
  - Sympa (mailing-lists)  http://www.picolibre.org/
  - WebDAV (web folders)
  - CVS, SubVersion (soon)
Libre Software
Definition (FSF)

Definition\(^1\) : a program is Free Software only when there are 4 Freedoms for the Public :

- Freedom to *run the program*, for any purpose
- Freedom to *study* how the program works, and *adapt* it to your needs
- Freedom to *redistribute* copies
- Freedom to *improve* the program, and *release your improvements*

Liberté, Égalité, Fraternité

• Freedom: Make copies, improve, distribute
• Equality: Same rights for everyone
• Friendship: Co-operation of all to build something together
Free/Libre/Open Source software (FLOSS)

- CALIBRE : « libre software »
- « Libre », as in liberty (or free as in freedom)
- [Free Software / Open Source] licence
- Several names, same phenomenon
- Free + Libre + OSS = FLOSS …
CALIBRE project
Context of CALIBRE project

• European Community (EC)
• DG Information Society of European Commission
• 6th Framework Programme (FP 6) : R&D funding programme of EC
• Academic consortium : research by academic institutions funded in FP6
• FP6 ending in 2006 (FP7)
« Coordination Action for LIBRE software »

- IST FP6 Project: 2 year: 2004-2006
- Ended September 2006
- Multi-disciplinary research team:
  - Economy,
  - Software Engineering,
  - Sociology, ...
- Critical mass of Europe's academic research in Libre software
CALIBRE Partners

• Universities and research centers in 12 European countries + China
  • In France: GET + UPMC

• More details on Calibre on http://www.calibre.ie/
Goals of CALIBRE
FLOSS as a 'silver bullet'

- Proponents claim FLOSS can solve "software crisis" (cost, quality and duration of development)
- Research needed to confirm
- Not one only model
- Future model for work and society
  - Wikipedia, open science, human genome
- Pitfalls?
  - FLOSS and Navajo Indians!
Why EC funded this research on FLOSS (>1.5 M euros)

- Libre/Open Source software model seen as big potential for European Industry
- To the next generation methods and services?
- From FLOSS to OSS 2.0?
- Foster Academic research / clustering
- Transfer lessons to the industry (Calibration industry forum)
Research on Libre Software ?
History of research

- « Cathedral and bazaar » (Eric S. Raymond) 1997
  - « Cathedral » : heavyweight process in hierarchical structure
  - « Bazaar » : loosely coordinated development teams
- Libre software community's own research
- Academic researchers have become interested for several years
Software engineering challenges

• Huge amount of freely available public data relating to libre software development projects

• Successful development model(s)

• Hope that data obtained from public sources can help understand the undergoing processes
Not only Computer Science

• Would appear to be primarily a ‘nerdy’ software topic

• Much interest from such a diverse range of research disciplines: sociology, economics, management, psychology, public policy and law, for example
Analysis of software

• Browsing source code to identify authors and metrics
• Some research paths:
  – Research in revisions repositories
  – Social networks analysis
  – Software evolution
• Tools to automate development repositories mining
Public data sources

Diagram showing relationships:
- Mailing Lists
- Source Code
- Versioning Repository
- Bug Tracking
- Other

Relationships:
1. Actor to Mailing Lists
2. Actor to Source Code
3. Actor to Versioning Repository
4. Actor to Bug Tracking
5. Other connections
Macro analysis: Distributions

Source [Robles]
Evolution of one software (SLOC growth)

- "Classical" methodology
- Usual profile: linear
- Linux: superlinear

Linux kernel source line count evolution Source: [Robles]
(BETA3) CVS Analysis for the KDE project

(BETA3) CVS Analysis for the KDE project - General Statistics

<table>
<thead>
<tr>
<th>Historical data</th>
<th>Date</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>First commit</td>
<td>1997-04-09</td>
<td>00:25:19</td>
</tr>
<tr>
<td>Last commit considered (note)</td>
<td>2004-03-22</td>
<td>20:59:43</td>
</tr>
<tr>
<td>Number of days</td>
<td>2539.9</td>
<td></td>
</tr>
</tbody>
</table>

(∗) CVSAnaly analysis date. This date is considered as the reference point for further analysis.

<table>
<thead>
<tr>
<th>Category</th>
<th>Number</th>
<th>Mean per module</th>
<th>Mean per committer</th>
<th>Mean per commit</th>
<th>Mean per day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules</td>
<td>79</td>
<td>1</td>
<td>0.09</td>
<td>3E-05</td>
<td>0.311</td>
</tr>
<tr>
<td>Committers</td>
<td>915</td>
<td>11.58</td>
<td>1</td>
<td>0.0003</td>
<td>0.36</td>
</tr>
<tr>
<td>Commits</td>
<td>2935436</td>
<td>37157.42</td>
<td>3208.13</td>
<td>0.0000</td>
<td>1155.73</td>
</tr>
<tr>
<td>Files</td>
<td>175657</td>
<td>2223.51</td>
<td>191.97</td>
<td>0.06</td>
<td>69.10</td>
</tr>
<tr>
<td>Aggregated Lines</td>
<td>106048029</td>
<td>1342390.11</td>
<td>115899.49</td>
<td>36.13</td>
<td>41752.64</td>
</tr>
<tr>
<td>Removed Lines</td>
<td>73534466</td>
<td>930816.03</td>
<td>80365.54</td>
<td>25.05</td>
<td>28951.72</td>
</tr>
<tr>
<td>Changed Lines</td>
<td>179582495</td>
<td>2273196.14</td>
<td>196265.02</td>
<td>61.18</td>
<td>70704.55</td>
</tr>
<tr>
<td>Final Lines</td>
<td>32513563</td>
<td>411564.09</td>
<td>35533.95</td>
<td>11.08</td>
<td>12601.12</td>
</tr>
</tbody>
</table>

File-type statistics for all modules

<table>
<thead>
<tr>
<th>File type</th>
<th>Modules</th>
<th>Commits</th>
<th>Files</th>
<th>Lines Changed</th>
<th>Lines Added</th>
<th>Lines Removed</th>
<th>Removed External files</th>
<th>CVS flag</th>
<th>First commit</th>
<th>Last commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>development</td>
<td>74</td>
<td>1061173</td>
<td>75050</td>
<td>35989453</td>
<td>25107823</td>
<td>11881630</td>
<td>87738</td>
<td>9428</td>
<td>101860</td>
<td>1997-04-10</td>
</tr>
<tr>
<td>i18n</td>
<td>25</td>
<td>813279</td>
<td>2305</td>
<td>114107944</td>
<td>61994713</td>
<td>52113231</td>
<td>64826</td>
<td>96</td>
<td>724323</td>
<td>1997-08-15</td>
</tr>
<tr>
<td>documentation</td>
<td>64</td>
<td>451647</td>
<td>24242</td>
<td>16104796</td>
<td>10040701</td>
<td>5264095</td>
<td>95235</td>
<td>151</td>
<td>346241</td>
<td>1997-04-13</td>
</tr>
</tbody>
</table>
Evolution of commits in time for top committers by percentage for each time interval

Number of Commits

Intervals (time) - 0: Project start - 10: today

Source: [CVSAnalY]
Developpers
## Counting in SourceForge

### By countries:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Country</th>
<th>Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>United States</td>
<td>425,620</td>
</tr>
<tr>
<td>2.</td>
<td>Germany</td>
<td>95,800</td>
</tr>
<tr>
<td>3.</td>
<td>United Kingdom</td>
<td>60,768</td>
</tr>
<tr>
<td>4.</td>
<td>Canada</td>
<td>49,109</td>
</tr>
<tr>
<td>5.</td>
<td>France</td>
<td>44,587</td>
</tr>
<tr>
<td>6.</td>
<td>China</td>
<td>36,517</td>
</tr>
</tbody>
</table>

... ... ...

(source: Gregorio Robles and Jesús M. González Barahona - 2006)
## Counting in SourceForge (2)

- **By regions:**
  - **Region**  **Developers**
  - Africa  12 560
  - Asia  127 275
  - EU  401 845
  - Europe  466 792
  - North America  485 679
  - Oceania  46 422
  - South America  36 330

(source: Gregorio Robles and Jesús M. González Barahona - 2006)
Socio-technical Analysis

- Structure of organisation = hints on software structure
- Analysis techniques for social networks
Developers network

Linux 1.0 (1994)
Developers linked by common authorship to same files

source [Robles]
Classical analysis of Apache modules feb. 2004 (source [Robles])
Reshaped with Girvan-Newman algorithm (source [Robles])
Apache 01/01/1999
Apache 01/01/2000
Apache 01/09/2000
Apache 01/02/2004
Valuing FLOSS

• Example: Debian 2.2 GNU/Linux (2001)
• Source lines of code: 55,201,526 (of which the Linux kernel forms under 6%)
• If written in a software company:
  – Estimated effort: 14,005 person years
  – Estimated schedule: 6.04 years (team of 2,318!)
  – Development cost: US$ 1,891,990,000

(Source: "Counting potatoes" by Gonzalez-Barahona et al)
New SE era?

- Public data sources are an important knowledge source for software projects.
- Non-intrusive observation is possible for technical or social analysis.
- Exhaustive analysis of huge amount of libre software projects is possible.
- Possibility to define methodologies which can be applied in real-life projects.
Limitations

• Some informations are not public (surveys)
• Some data sources are incomplete
• Necessary validation by the projects
• Respecting privacy
CVSAnaly & phpGW
http://localhost/~olivier/cvsanaly-web-phpgroupware/
OSS & Industry
Calibration industry forum

- One of the ways to disseminate knowledge, and strategic decision criteria
- Targeted at big European Industry
- Not targeted at pure software firms
- Example of current members: Philips Medical Systems, Eurocontrol, Telefónica, Thales, Vodafone, Thalès ...
- Link with academia and Commission
Commoditisation of software
(FL)OSS 2.0 ?

The Transformation of Free/Libre/Open Source Software
### Domain and planning

<table>
<thead>
<tr>
<th>Past</th>
<th>Present</th>
<th>Future Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driven by individual developer needs (an itch worth scratching).</td>
<td>Driven by purposive strategies by major players trying to gain competitive advantage.</td>
<td>Balancing organisational &amp; individual efforts &amp; rewards.</td>
</tr>
<tr>
<td>Generally, horizontal infrastructure (operating systems, utilities, compilers, DBMS, web &amp; print servers).</td>
<td>More visible IS applications in vertical domains.</td>
<td>How to stimulate development in vertical domains not immediately attractive to global development community.</td>
</tr>
</tbody>
</table>
Analysis & Design

<table>
<thead>
<tr>
<th>Past</th>
<th>Present</th>
<th>Future Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Part of conventional agreed-upon knowledge in software development.</td>
<td>- More complex in spread to vertical domains where business requirements are not universally agreed upon</td>
<td>- Managing requirements elicitation and specification in open software networks.</td>
</tr>
<tr>
<td>- Firmly based on principles of modularity to accomplish separation of concerns.</td>
<td>- More formalized software development processes.</td>
<td>- Organisational and network aspects of ensuring OSS quality – e.g. parallel distributed development leads to excessive modularity which potentially creates maintainability problems.</td>
</tr>
<tr>
<td>- Often done by one person/core group as ‘a tail-light to follow’ in the bazaar.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Calibre D2.3 Roadmap for Research in the Domain of Libre/OSS – May 06 - UL
## Implementation, Acquisition & Exploitation

<table>
<thead>
<tr>
<th>Past</th>
<th>Present</th>
<th>Future Challenges</th>
</tr>
</thead>
</table>
| - Development lifecycle characterised by distributed / parallel:  
  1. Coding  
  2. Reviewing  
  3. Pre-commit testing  
  4. Development releasing  
  5. Parallel Debugging  
  6. Production Releasing  
- Ad hoc acquisition and back office exploitation | - Development lifecycle is part of a larger more formalized development process (but less bazaar-like).  
- Formalised IT acquisition strategies and exploitation in both end user and back office contexts. | - Managing complex OSS projects (particularly inter-organisational / network aspects).  
- Inner source – how to transfer benefits of OSS development methods to conventional development, especially in context of global software development.  
- Ensuring flexible software (agile development methods in OSS networks).  
- Evaluation & appraisal methods for OSS – making the business case for developing & deploying OSS (inc deriving appropriate total cost of ownership (TCO) measures). |

Source: Calibre D2.3 Roadmap for Research in the Domain of Libre/OSS – May 06 - UL
## Productisation & Business strategies

<table>
<thead>
<tr>
<th>Past</th>
<th>Present</th>
<th>Future Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal infrastructure (operating systems, utilities, compilers, DBMS, web &amp; print servers)</td>
<td>More visible IS applications in vertical domains</td>
<td>How to stimulate development in vertical domains not attractive to global development community</td>
</tr>
<tr>
<td>Primary Business Strategies</td>
<td>Value-added service enabling</td>
<td>Further exploration of hybrid business models</td>
</tr>
<tr>
<td>o Value-added service enabling</td>
<td>Value</td>
<td>Deriving appropriate total cost of ownership (TCO) measures for open source</td>
</tr>
<tr>
<td>o Loss-leader/market-creating</td>
<td>Market-creating</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Loss-leader</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Dual product/ licensing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Cost reduction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Accessorising</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leverage community development</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leverage OSS brand</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘Whole Product’ approach</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customers willing to pay for a professional ‘whole product’ approach</td>
<td>Effecting the ‘whole product’ approach</td>
</tr>
<tr>
<td>Haphazard Product Support</td>
<td>Plethora of licenses (85 to date validated by OSI or FSF)</td>
<td>Safeguarding against IPR infringement</td>
</tr>
<tr>
<td>- much customer reliance on email lists/bulletin boards, or on support provided by specialized software firms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Licensing: GPL, LGPL, Artistic License, BSD &amp; commercially-oriented MPL</td>
<td></td>
<td>Achieving balance between ‘value for money’</td>
</tr>
<tr>
<td>Key Tension: Achieving balance between collectivist v.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion
FOSS 2.0 Challenges - Research

- Transferring lessons to conventional development
  - Open sourcing an unknown workforce
  - Expanded role of users and altered user developer relationship

- Elaboration of business models
FOSS 2.0 Challenges – Practice

• Balancing 'value creation' with 'acceptable community values'
• Stimulating development in vertical domains
• Implementing Open Source Service Networks and 'whole product' approach
• Safeguarding against IPR infringement
  – Indemnification of end users
European R&D

- FP6 IST
  - QualiPSo
  - QualOSS
  - SQO OSS
- ITEA
  - COSI
etc.
Credits

- Many thanks to my Calibre colleagues
  - Dr Gregorio Robles-Martínez (Universidad Rey Juan Carlos, Spain),
  - Brian Fitzgerald (University Limerick), leader of the CALIBRE project,
  - Rishab Aiyer Ghosh (MERIT, Netherlands).
merci
thanks