Some results from the CALIBRE project

Olivier BERGER (GET/INT – France)
ObjectWeb and OrientWare Open Source software seminar
CISIS 2006
Dalian (China), 23 June 2006
Contents

• Intro
• Calibre's context
• Some results
  – software engineering
  – sociology, economics
  – dissemination
• New characterisation of OSS?
About GET & INT

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

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

• Several teams specialised in research and practice on Libre Software
General context
Definition of Free/Libre/Open Source software

• 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 ...
Translations of *Libre* software

- Chinese: 自由軟件 / 自由軟體 (?)
- Japanese: 自由なソフトウェア (?)
  (フリーソフトウェア) (?)
- Korean: 자유 소프트웨어 (?)
- French: logiciel *libre*
- Italian: software *libero*
- Spanish: software *libre*
CALIBRE project

CALIBRE
Coordination Action for LIBRE software
Context of CALIBRE project

- European Community (EC)
- DG Information Society of European Commission
- 6\textsuperscript{th} 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 to 2006
- Ending July (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 funds 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)
Software Engineering
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
Concepts and techniques

• « 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
Building a discipline

• 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
Macro analysis: Distributions

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

• "Classical" methodology
• Usual profile: linear
• Linux: superlinear
(BETA3) CVS Analysis for the KDE project

Contents
- Index
- About
- Statistics
- Modules
- Committers
- Inequality
- FAQ
- Credits

Module Search: [ ] [Go]

Committer Search: [ ] [Go]

Language: English [Go]

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

Historical data

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

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

<table>
<thead>
<tr>
<th></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>11.58</td>
<td>1</td>
<td>0.09</td>
<td>0.0311</td>
</tr>
<tr>
<td>Committers</td>
<td>915</td>
<td>37157.42</td>
<td>2223.51</td>
<td>52215.73</td>
<td>69.10</td>
</tr>
<tr>
<td>Commits</td>
<td>2935436</td>
<td>1342900.11</td>
<td>40386.54</td>
<td>36.13</td>
<td>41752.64</td>
</tr>
<tr>
<td>Files</td>
<td>175657</td>
<td>930816.03</td>
<td>191.97</td>
<td>0.06</td>
<td>69.10</td>
</tr>
<tr>
<td>Aggregated Lines</td>
<td>106048029</td>
<td>2273196.14</td>
<td>196265.02</td>
<td>51.18</td>
<td>70704.55</td>
</tr>
<tr>
<td>Removed Lines</td>
<td>73534466</td>
<td>411564.09</td>
<td>35533.95</td>
<td>11.08</td>
<td>12601.12</td>
</tr>
<tr>
<td>Changed lines</td>
<td>179582495</td>
<td>118816.30</td>
<td>80365.54</td>
<td>36.13</td>
<td>41752.64</td>
</tr>
<tr>
<td>Final Lines</td>
<td>32513563</td>
<td>9428</td>
<td>724323</td>
<td>151</td>
<td>346241.00</td>
</tr>
</tbody>
</table>

File-type statistics for all modules

<table>
<thead>
<tr>
<th>File type</th>
<th>Modules</th>
<th>Commits</th>
<th>Commits</th>
<th>Files</th>
<th>Lines</th>
<th>Lines</th>
<th>Lines</th>
<th>Lines</th>
<th>Removed</th>
<th>Removed</th>
<th>Removed</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>25107823</td>
<td>118816.30</td>
<td>87738</td>
<td>9428</td>
<td>101880</td>
<td>1997-04-10</td>
<td>2004-03-22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i18n</td>
<td>25</td>
<td>813279</td>
<td>61994713</td>
<td>52113231</td>
<td>64826</td>
<td>90</td>
<td>724323</td>
<td>1997-08-15</td>
<td>2004-03-22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>documentation</td>
<td>64</td>
<td>461647</td>
<td>16104796</td>
<td>10840701</td>
<td>5264095</td>
<td>95325</td>
<td>346241</td>
<td>1997-04-13</td>
<td>2004-03-22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evolution of commits in time for top committers by percentage for each time interval

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>425620</td>
</tr>
<tr>
<td>2.</td>
<td>Germany</td>
<td>95800</td>
</tr>
<tr>
<td>3.</td>
<td>United Kingdom</td>
<td>60768</td>
</tr>
<tr>
<td>4.</td>
<td>Canada</td>
<td>49109</td>
</tr>
<tr>
<td>5.</td>
<td>France</td>
<td>44587</td>
</tr>
<tr>
<td>6.</td>
<td>China</td>
<td>36517</td>
</tr>
</tbody>
</table>

... ... ...

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

- **By regions:**

<table>
<thead>
<tr>
<th>Region</th>
<th>Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>12,560</td>
</tr>
<tr>
<td>Asia</td>
<td>127,275</td>
</tr>
<tr>
<td>EU</td>
<td>401,845</td>
</tr>
<tr>
<td>Europe</td>
<td>466,792</td>
</tr>
<tr>
<td>North America</td>
<td>485,679</td>
</tr>
<tr>
<td>Oceania</td>
<td>46,422</td>
</tr>
<tr>
<td>South America</td>
<td>36,330</td>
</tr>
</tbody>
</table>

*(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 01/02/2004
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
Dissemination
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, ...

• Link with academia and Commission
OSS 2.0
The Transformation of Free/Libre/Open Source Software
## From FLOSS to OSS 2.0

<table>
<thead>
<tr>
<th>Lifecycle</th>
<th>FLOSS</th>
<th>OSS 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>&quot;an itch worth scratching&quot;</td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td>part of general agreed-upon knowledge in s/w development</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>modularity to achieve separation of concerns and reduce learning curve</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-commit test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel debugging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organising principle</td>
<td>having a tail-light to follow in bazaar</td>
<td></td>
</tr>
<tr>
<td>Optimistic concurrency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning</td>
<td>vendor-led purposive strategies</td>
<td></td>
</tr>
<tr>
<td>Analysis &amp; design</td>
<td>more complex in vertical domains where business requirements not universally understood</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>development process becoming less bazaar-like</td>
<td></td>
</tr>
<tr>
<td>More holistic and planned development approach</td>
<td>with paid developers considering marketing a consumable product</td>
<td></td>
</tr>
</tbody>
</table>
## From FLOSS to OSS 2.0 cont'd

<table>
<thead>
<tr>
<th>Product</th>
<th>FLOSS</th>
<th>OSS 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domains</td>
<td>● Horizontal 'invisible' infrastructure</td>
<td>● More visible vertical domains</td>
</tr>
<tr>
<td></td>
<td>● Value-added/service-enabling</td>
<td>● Hybrid models</td>
</tr>
<tr>
<td></td>
<td>● Loss-leader/market-creating</td>
<td>● Value-added service enabling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Bootstrapping</td>
</tr>
<tr>
<td>Business Models</td>
<td></td>
<td>● Market-creating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Loss leader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dual product/licensing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cost reduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Accessorising</td>
</tr>
<tr>
<td>Support</td>
<td>● Fairly haphazard bulletin boards, specialised firms - &quot;the thrilling adventure of installing open source&quot;</td>
<td>● Leveraging community development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Leveraging open source brand</td>
</tr>
<tr>
<td>Licensing</td>
<td>● GPL, LGPL, Artistic, BSD, emergence of commercial MPL</td>
<td>● Open Source Service Networks (OSSN)</td>
</tr>
<tr>
<td></td>
<td>● 'Viral' term used</td>
<td>○ Customer will pay for 'whole product' professional approach</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Plethora of licenses (from Microsoft alone!)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 'Reciprocal' term used</td>
</tr>
</tbody>
</table>
Conclusion
OSS 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
OSS 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
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
xie xie