

FLOSS 2.0 ?

Some results from the CALIBRE project

Olivier BERGER (GET/INT - France)
phpGroupware Conference 2006
Paris (France), 6 November 2006



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 96
- 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)

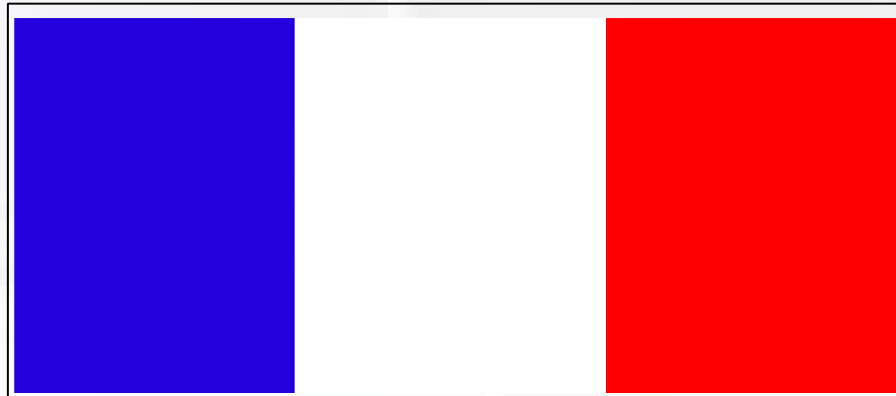
Definition¹ : a program is Free Software only when there are 4 Freedoms for the Public :

- 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*

1. <http://www.gnu.org/philosophy/free-sw.html>

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 ?

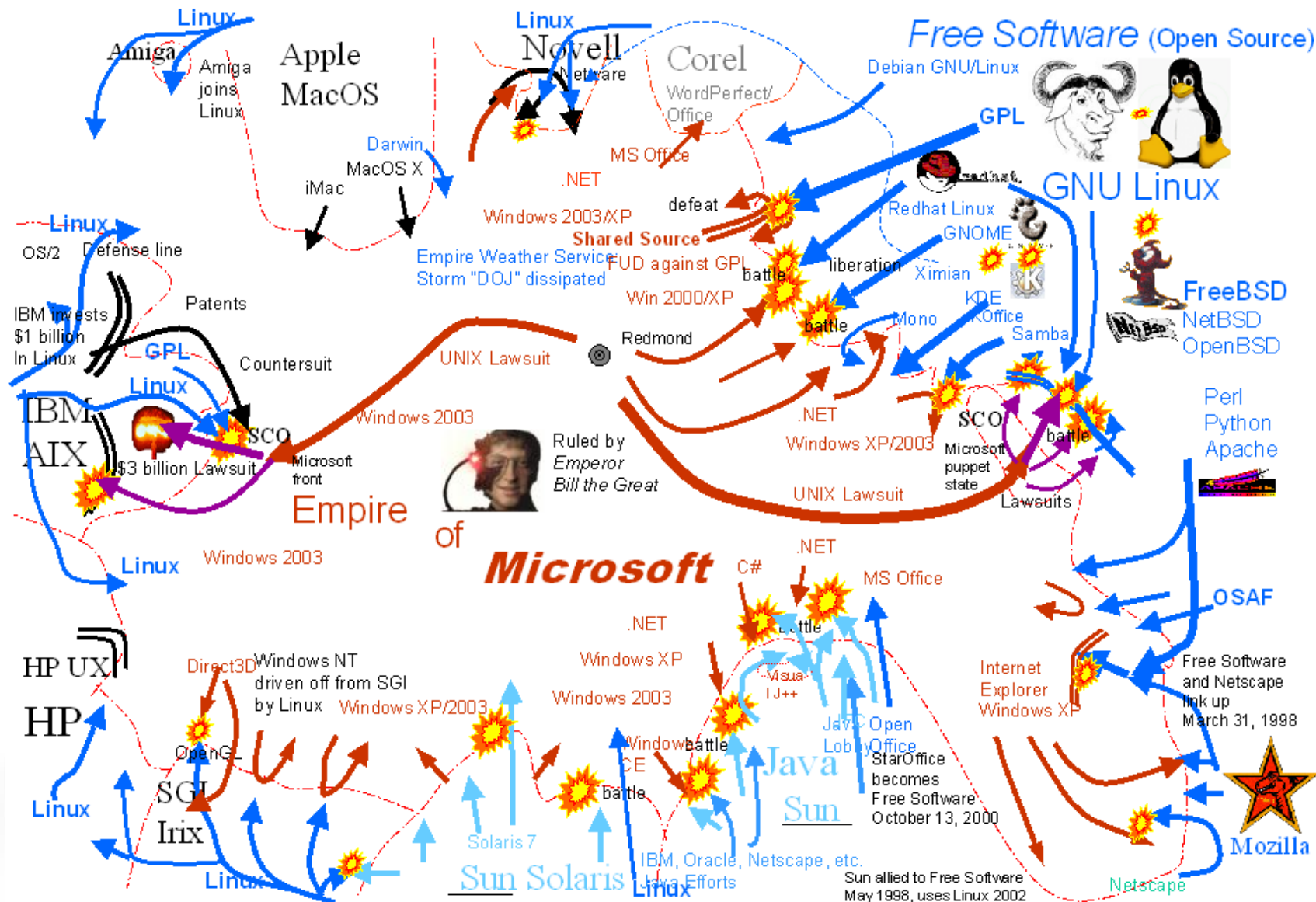
History of research

- « 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

<http://www.atai.org/softwarewar.png>

Empire Strikes Back

Free Software (Open Source)



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



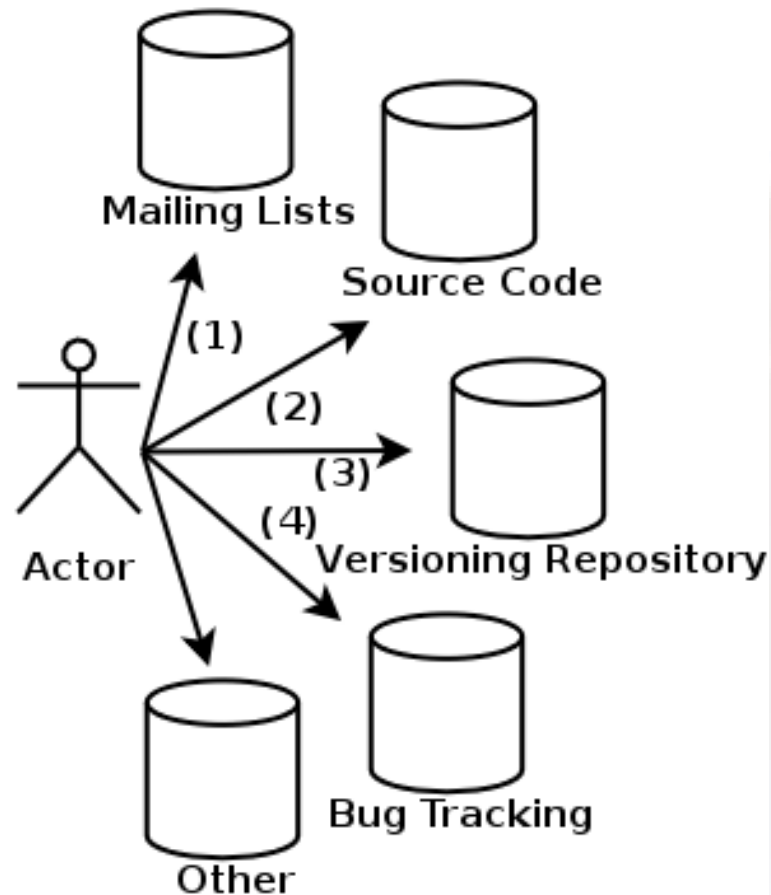
Not only Computer Science

- 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

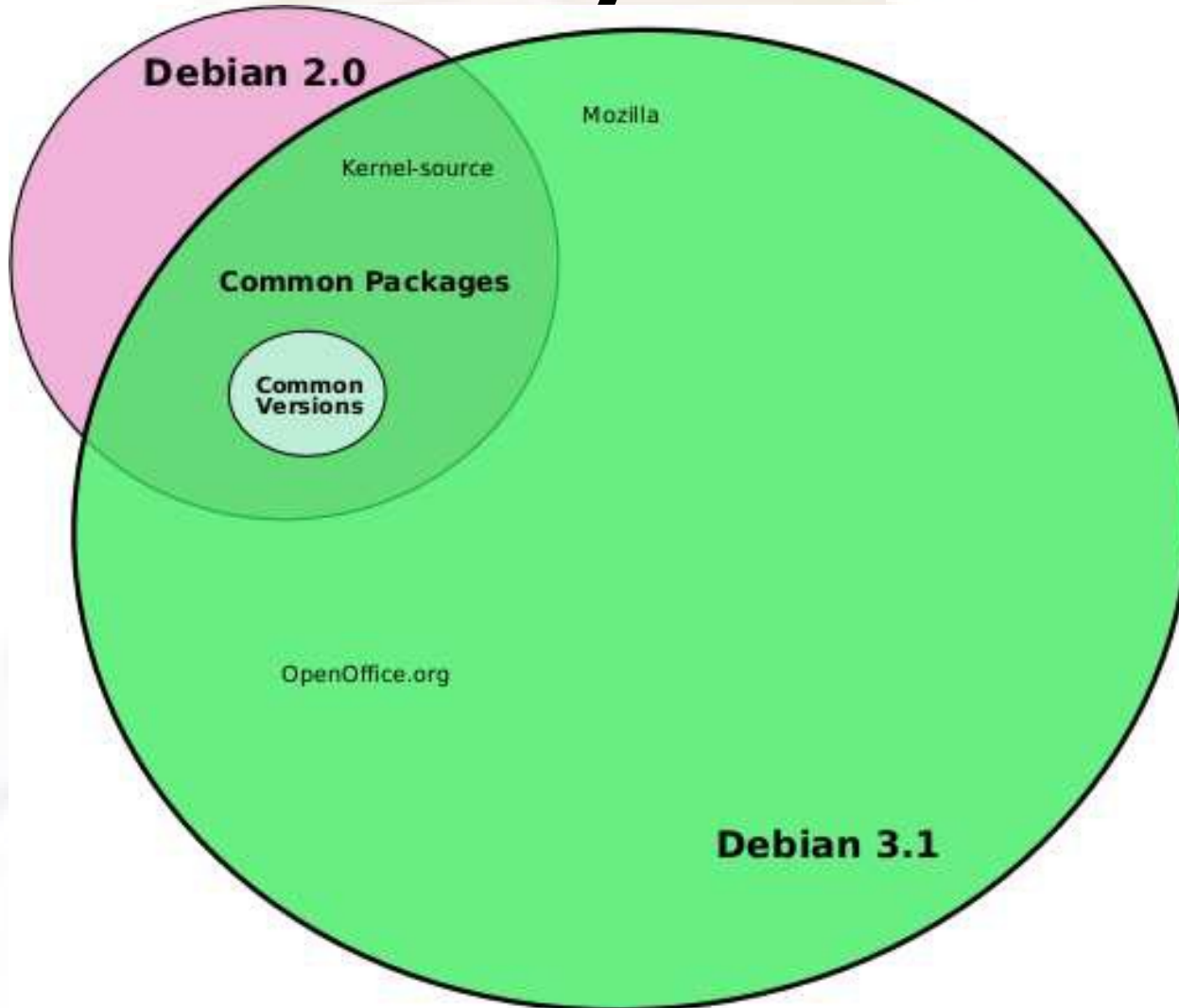
Analysis of software

- 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



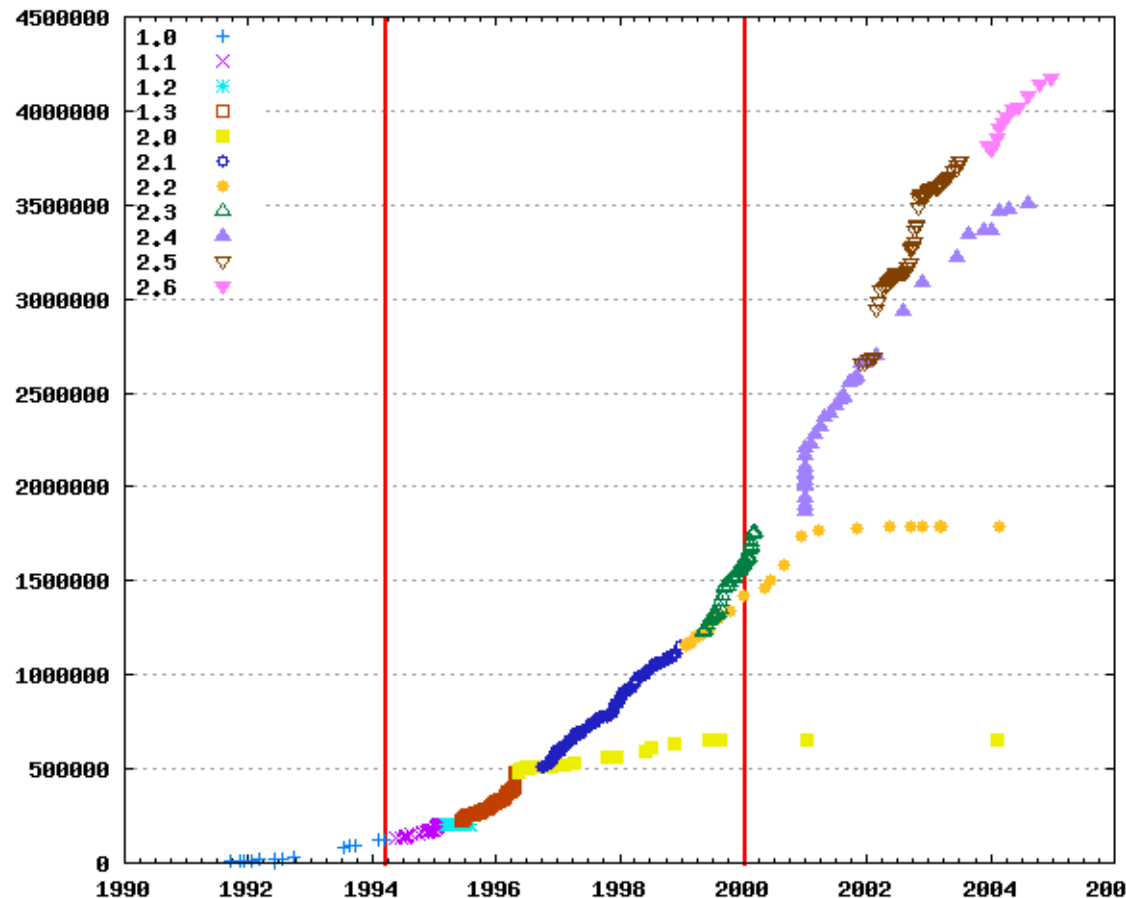
Macro analysis : Distributions



Source
[Robles]

Evolution of one software (SLOC growth)

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



Linux kernel source line count evolution Source: [Robles]

CVSAnalY

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

File Edit View Go Bookmarks Tools Help

http://libresoft.urjc.es/cvsanal/kde3-cvs/index.php?menu=Statistics

Libre Software Engineering

(BETA3) CVS Analysis for the KDE project

Contents

- [Index](#)
- [About](#)
- [Statistics](#)
- [Modules](#)
- [Committers](#)
- [Inequality](#)
- [FAQ](#)
- [Credits](#)

Module Search

Committer Search

Language

English

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

Historical data

First commit	1997-04-09 00:25:19
Last commit considered (*)	2004-03-22 20:59:43
Number of days	2539.9

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

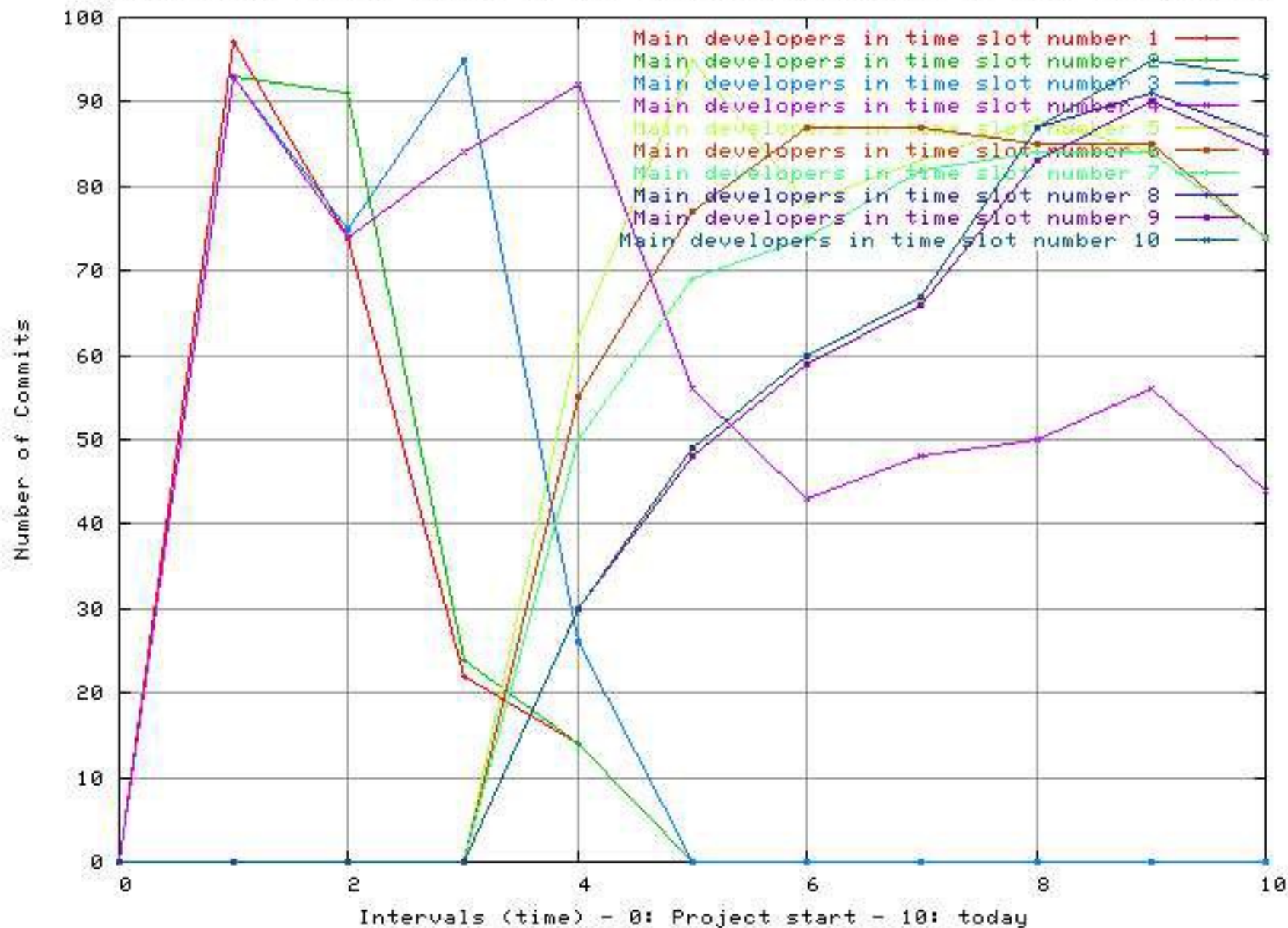
	Number	Mean per module	Mean per commiter	Mean per commit	Mean per day
Modules	79	1	0.09	3E-05	0.0311
Committers	915	11.58	1	0.0003	0.36
Commits	2935436	37157.42	3208.13	1	1155.73
Files	175657	2223.51	191.97	0.06	69.16
Aggregated Lines	106048029	1342380.11	115899.49	36.13	41752.84
Removed Lines	73534466	930816.03	80365.54	25.05	28951.72
Changed lines	179582495	2273196.14	196265.02	61.18	70704.55
Final Lines	32513563	411564.09	35533.95	11.08	12801.12

File-type statistics for all modules

File type	Modules	Commits	Files	Lines Changed	Lines Added	Lines Removed	Removed External files	CVS flag	First commit	Last commit
development	74	1061173	76505	36989453	25107823	11881630	87738	9428	101860	1997-04-10 2004-03-22
i18n	26	815279	2305	114107944	61994713	52113231	64926	96	724323	1997-08-15 2004-03-22
documentation	64	461647	24242	16104796	10840701	5264095	85235	151	346241	1997-04-13 2004-03-22

Done

Evolution of commits in time for top committers by percentage for each time interval



Source : [CVSAnalY]

Developeppers





Counting in SourceForge

- **By countries:**

Rank	Country	Developers
1.	United States	425620
2.	Germany	95800
3.	United Kingdom	60768
4.	Canada	49109
5.	France	44587
6.	China	36517
...

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

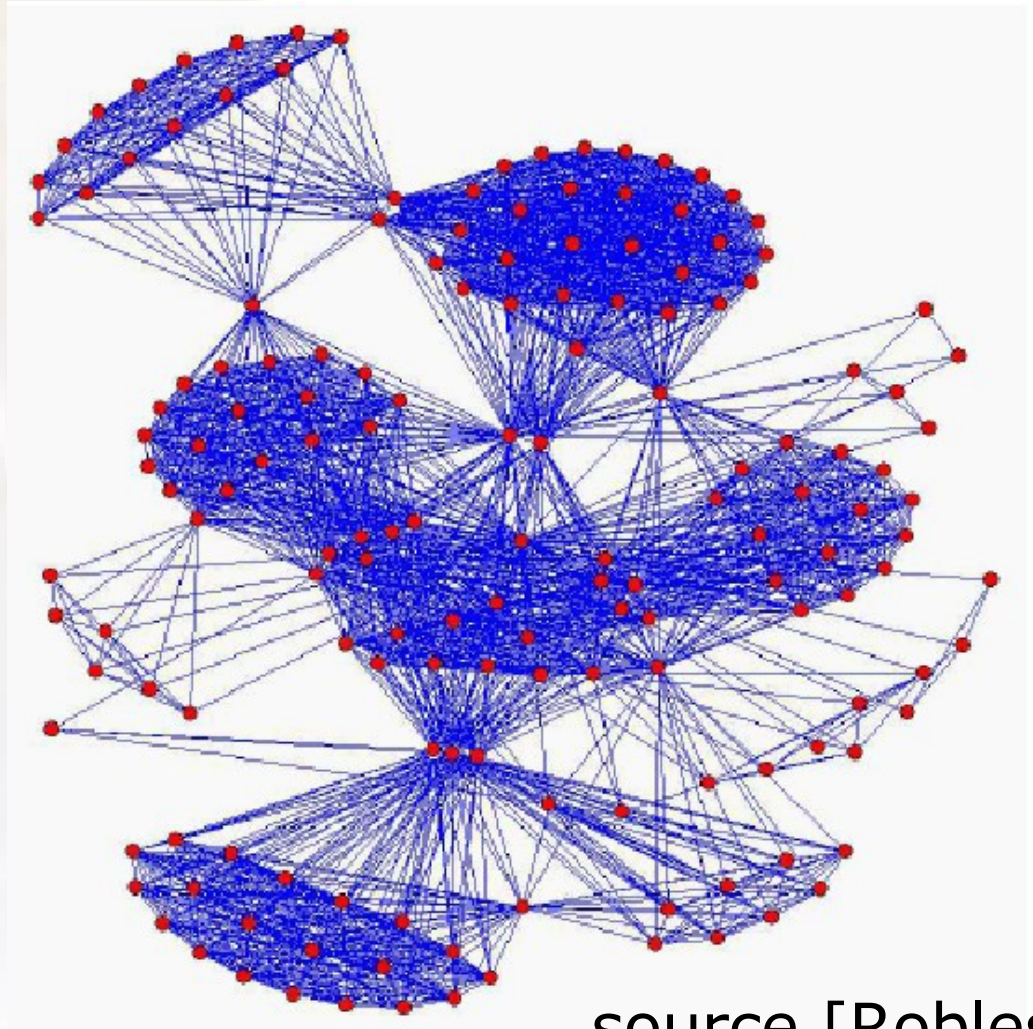
Socio-technical Analysis

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

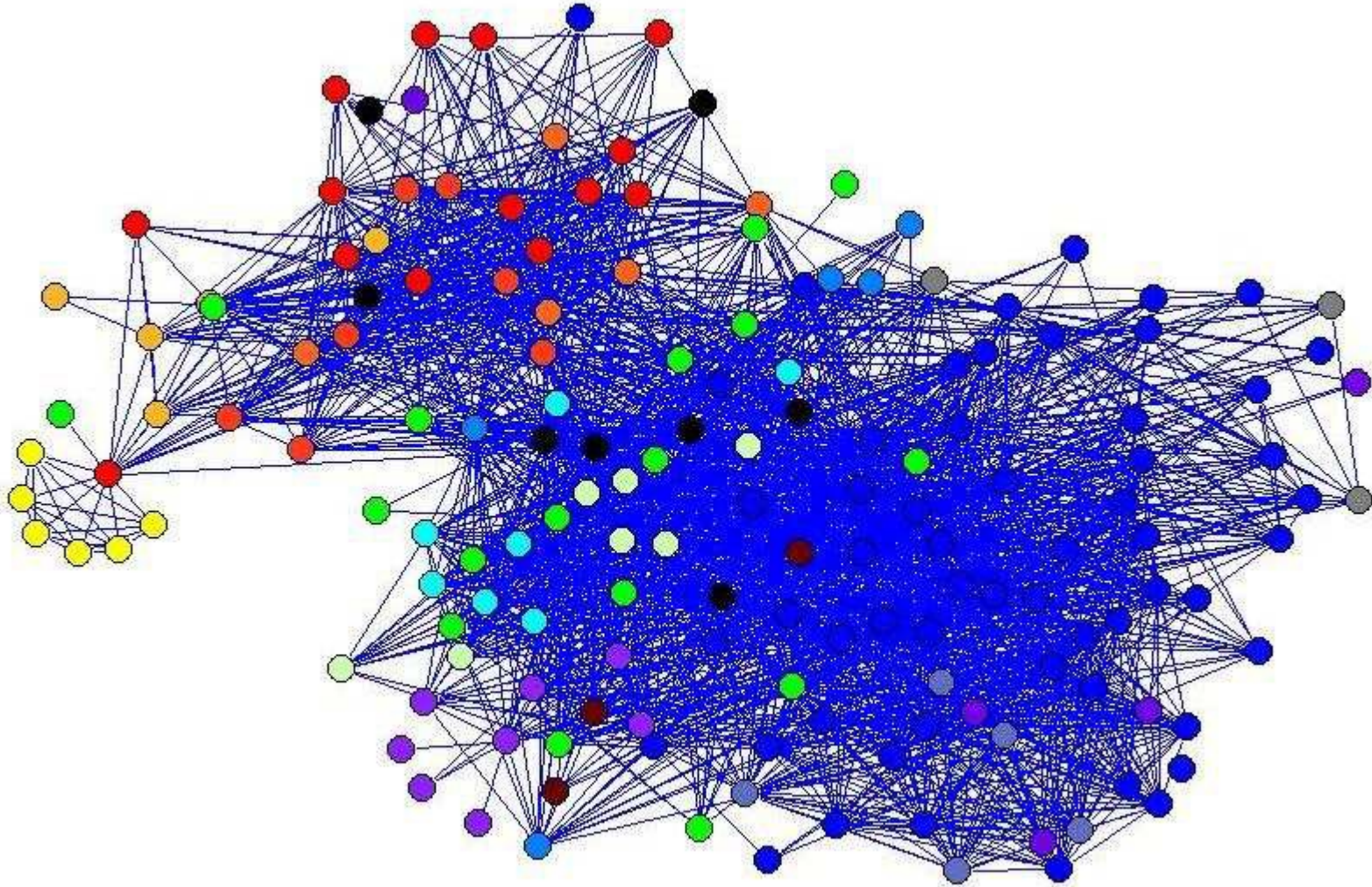
Developers network

**Linux 1.0
(1994)**

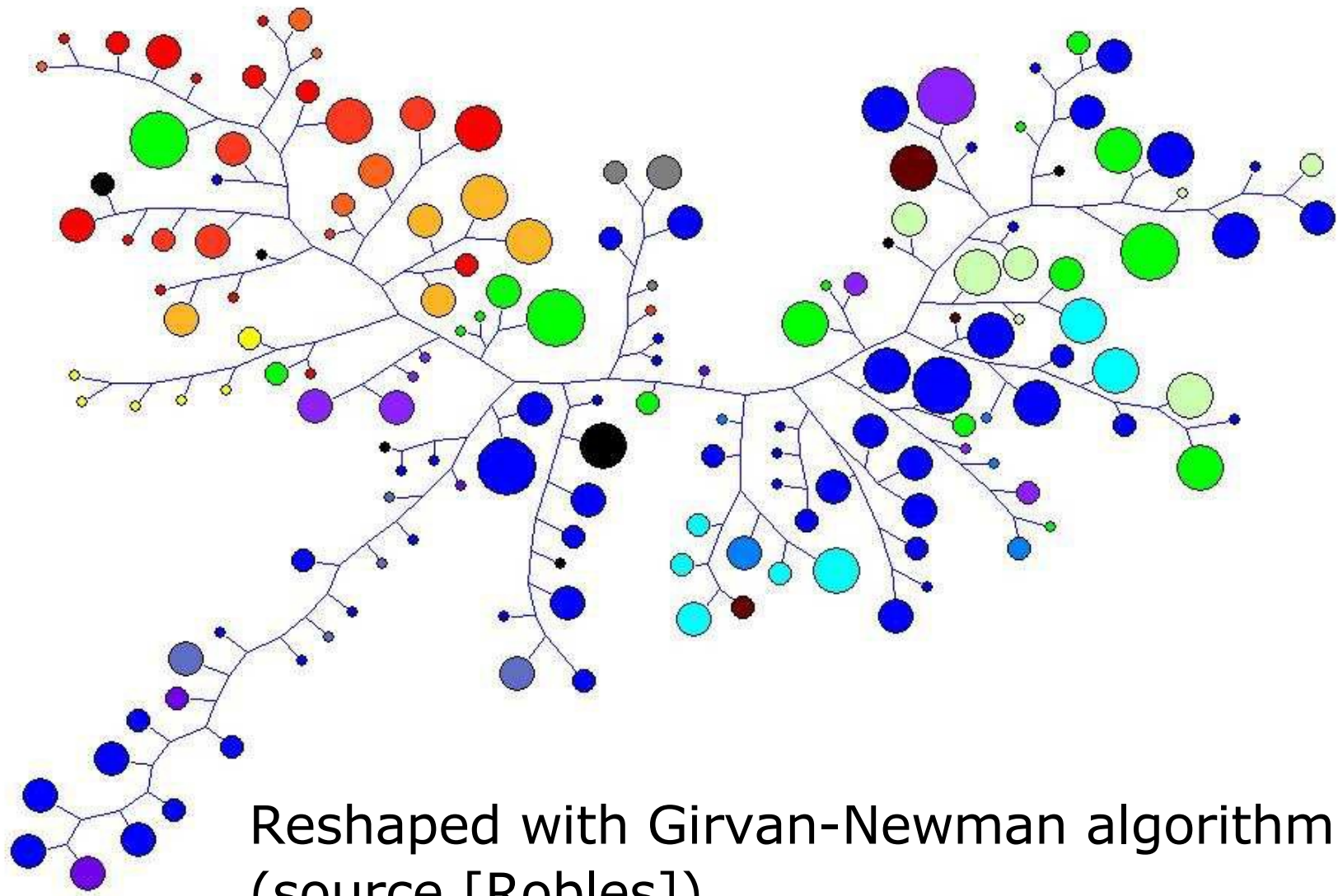
Developers
linked by
common
authorship to
same files



source [Robles]

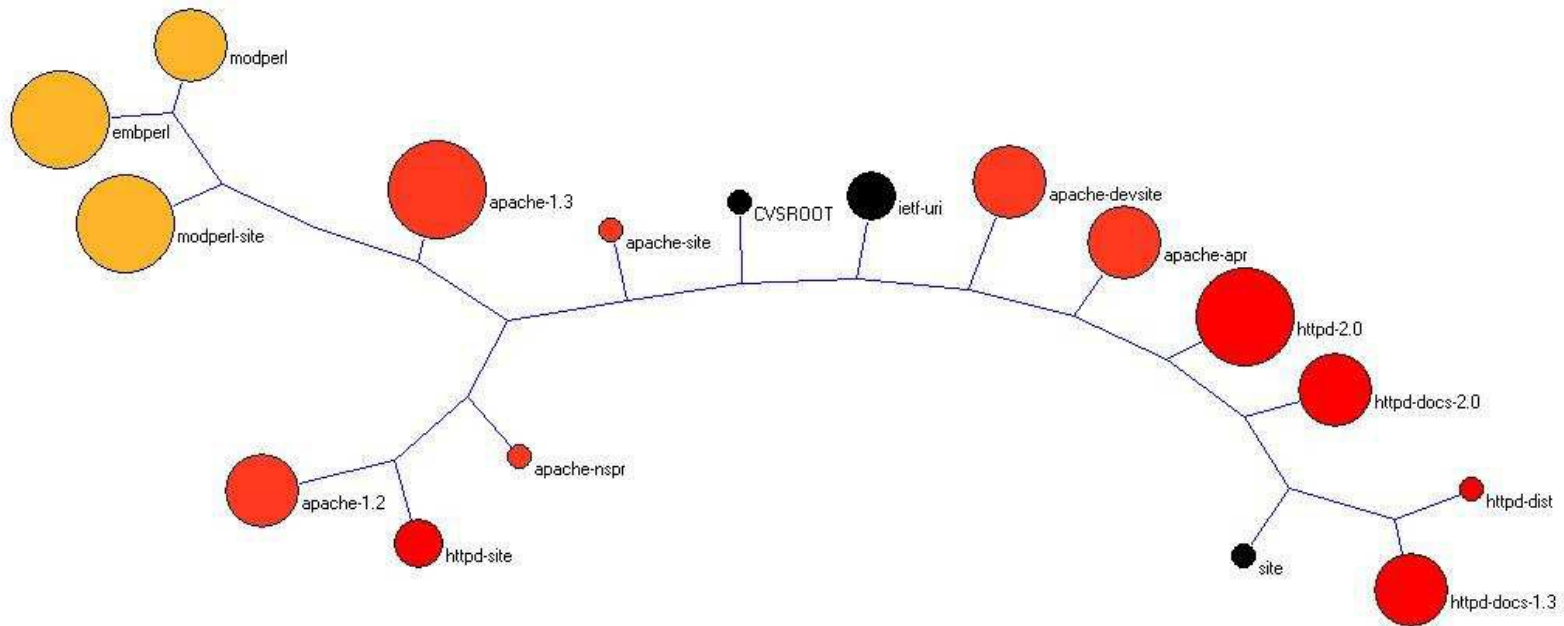


Classical analysis of Apache modules feb. 2004 (source [Robles]
get

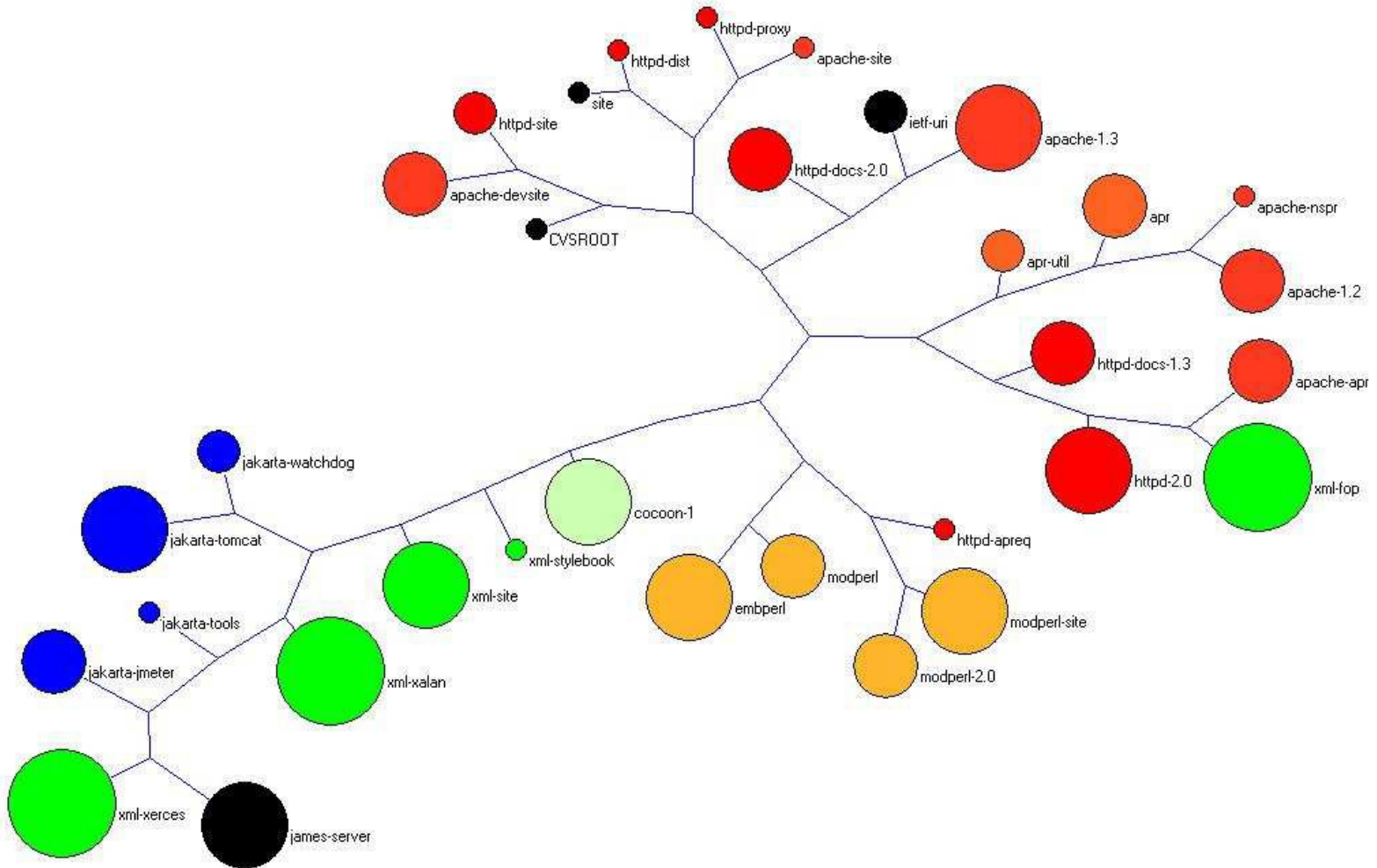


Reshaped with Girvan-Newman algorithm
(source [Robles])

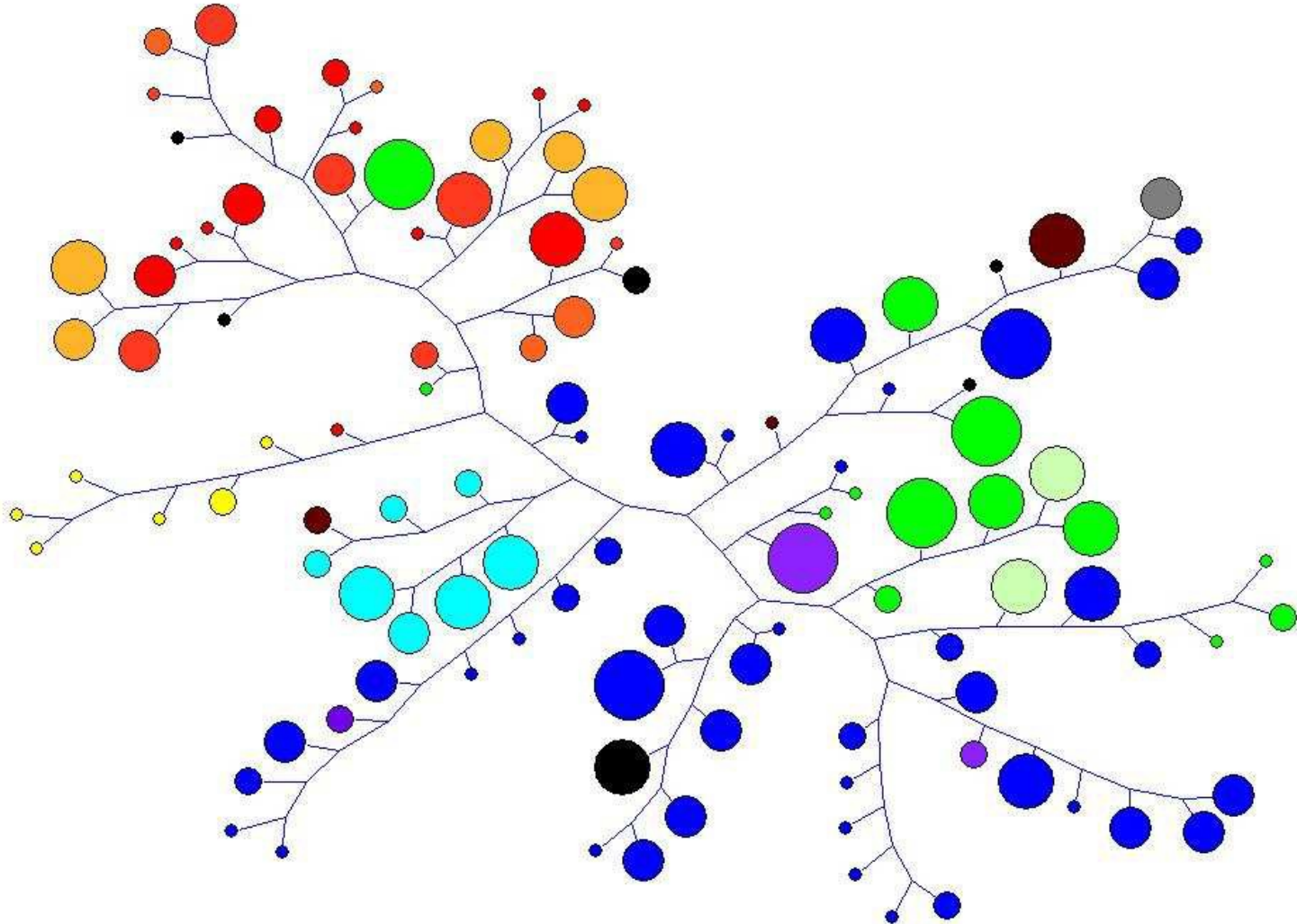
Apache 01/01/1999



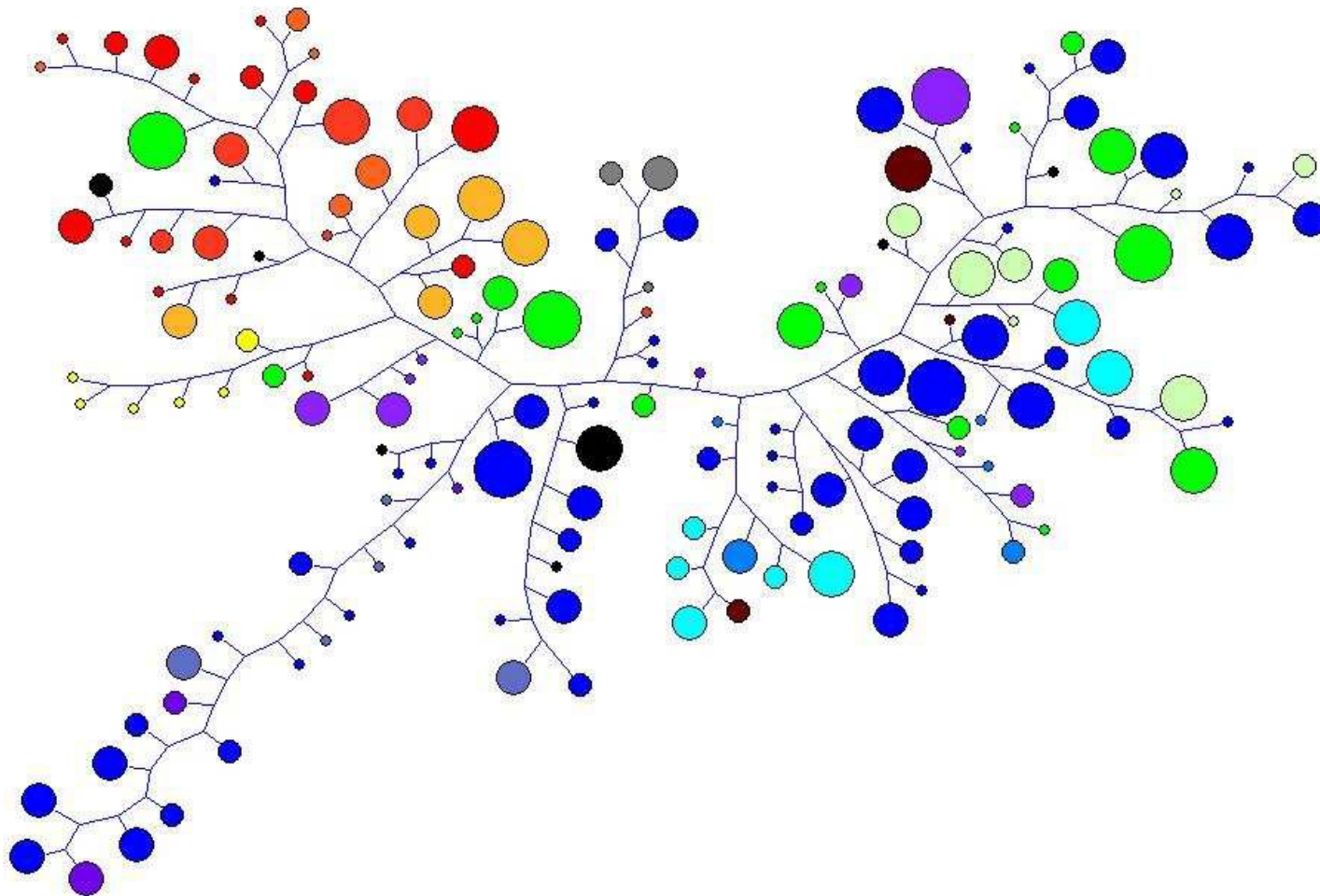
Apache 01/01/2000



Apache 01/01/2002



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



Institut National des
Télécommunications

CVSAnaly & phpGW

<http://localhost/~olivier/cvsanaly-web-phpgroupware/>

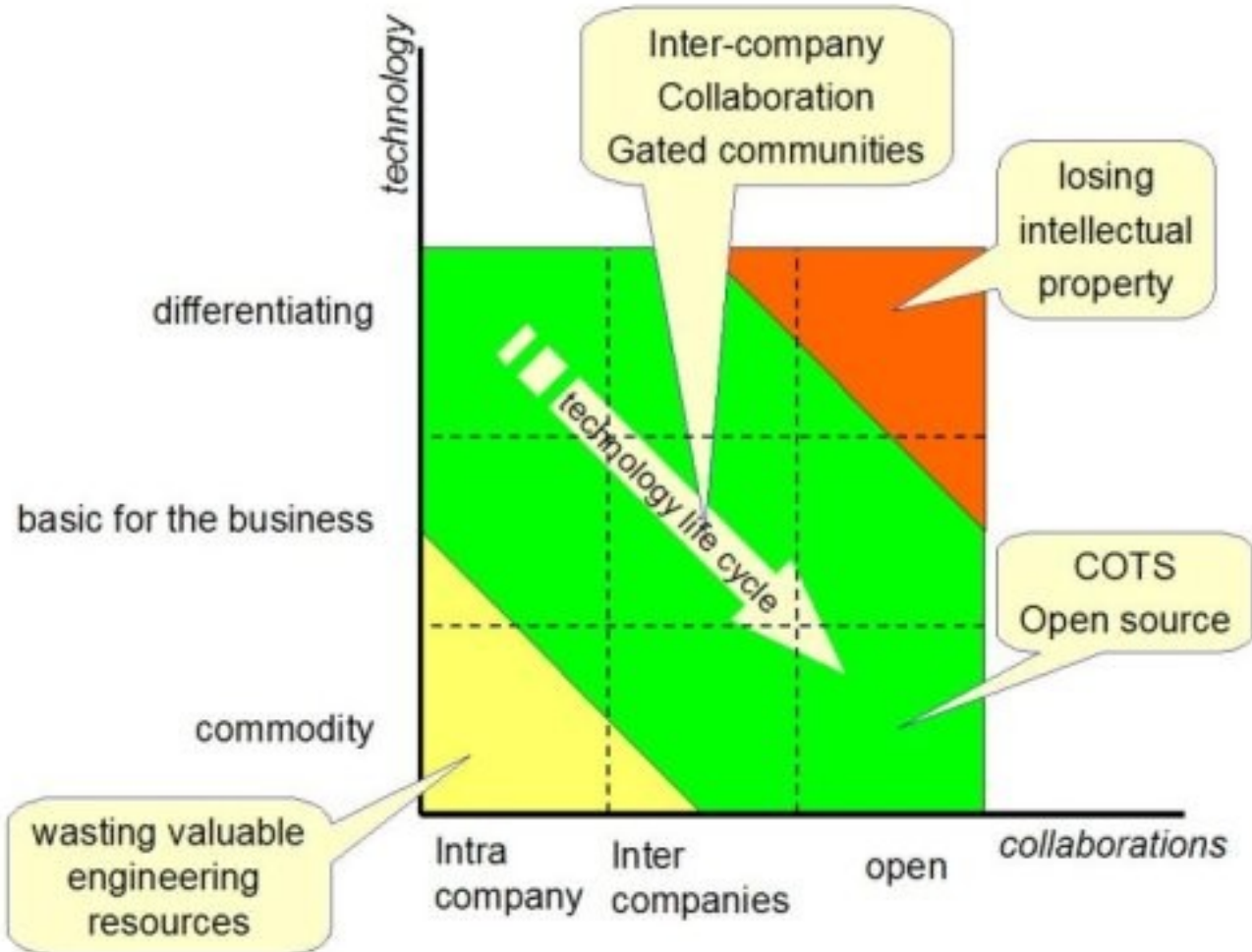


OSS & Industry

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

Past

- Driven by individual developer needs (an itch worth scratching).
- Generally, horizontal infrastructure (operating systems, utilities, compilers, DBMS, web & print servers).

Present

- Driven by purposive strategies by major players trying to gain competitive advantage.
- More visible IS applications in vertical domains.

Future Challenges

- Balancing organisational & individual efforts & rewards.
- How to stimulate development in vertical domains not immediately attractive to global development community.

Analysis & Design

Past

- Part of conventional agreed-upon knowledge in software development.
- Firmly based on principles of modularity to accomplish separation of concerns.
- Often done by one person/ core group as 'a tail-light to follow' in the bazaar.

Present

- More complex in spread to vertical domains where business requirements are not universally agreed upon
- More formalized software development processes.

Future Challenges

- Managing requirements elicitation and specification in open software networks.
- Organisational and network aspects of ensuring OSS quality – e.g. parallel distributed development leads to excessive modularity which potentially creates maintainability problems.

Implementation, Acquisition & Exploitation

Past

- 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

Present

- 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.

Future Challenges

- 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).

Productisation & Business strategies

Past

- Horizontal infrastructure (operating systems, utilities, compilers, DBMS, web & print servers)
- Primary Business Strategies
 - Value-added service enabling
 - Loss-leader/market-creating
- Haphazard Product Support - much customer reliance on email lists/bulletin boards, or on support provided by specialized software firms
- Licensing: GPL, LGPL, Artistic License, BSD & commercially-oriented MPL
- Key Tension: Achieving balance between collectivist

Present

- More visible IS applications in vertical domains
- Value-added service enabling Bootstrapping
- Market-creating
 - Loss-leader
 - Dual product/ licensing
 - Cost reduction
 - Accessorising
- Leverage community development
- Leverage OSS brand
- 'Whole Product' approach
- Customers willing to pay for a professional 'whole product' approach
- Plethora of licenses (85 to date validated by OSI or FSF)

Future Challenges

- How to stimulate development in vertical domains not attractive to global development community
- Further exploration of hybrid business models
- Deriving appropriate total cost of ownership (TCO) measures for open source
- Effecting the 'whole product' approach
- Safeguarding against IPR infringement
- Achieving balance between 'value for money' v

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