Libre software research in big projects: Some examples from research done on Linux, GNOME, KDE and Apache

Juan José Amor Iglesias, Israel Herraiz Tabernero
Universidad Rey Juan Carlos
{jjamor,herraiz} _at_ gsyc.escet.urjc.es

Brussels, February 26th 2005

Summary

- Free/Libre Open Source
- Advantages: public data
- Studying Source Code
- Studying Version Control data
- Social Networks

Free/Libre Open Source

OSD Model. Main features:

- License for use, distribution, modification.
- Source code available
  - One can access source code, to see it, to improve it and also take part in "official" development.
- Collaborative and geographically distributed development
  - Needed tools for:
    - Remote access to source code
    - Version control
    - Bug tracking
    - Developer discussion (mail lists, forums) ...
    - (generally) **all publicly available**

Lots of data publicly available

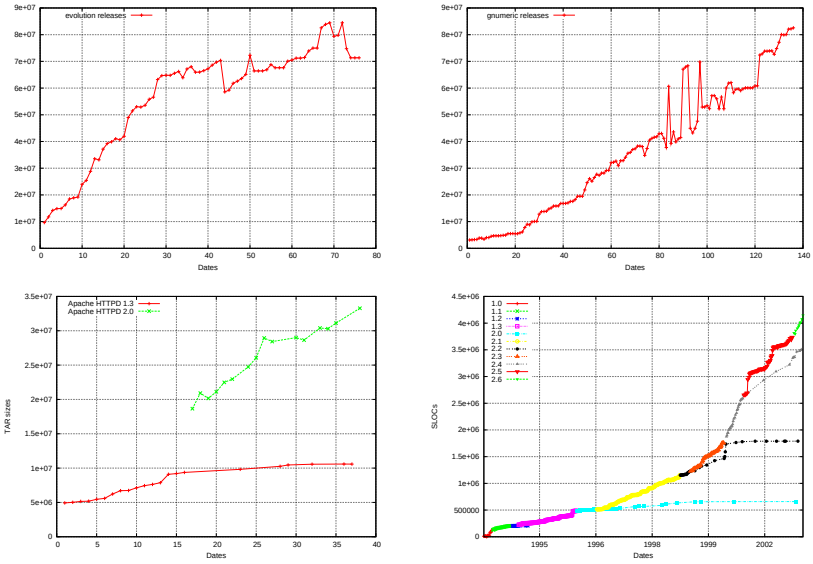
- Software releases (both binary and source)
- Version control: CVS, Subversion, etc
- Project documentation: man, info, Docbook, etc.
- Bug tracking: Bugzilla, Debian, Sourceforge/GForge sites, etc
- Mailing lists: Mailman / list archives
- Forums
- Usage stats (Debian Popularity contests)
- User and Developer Surveys, such as FLOSS
- and more...

Studying a project from a data source

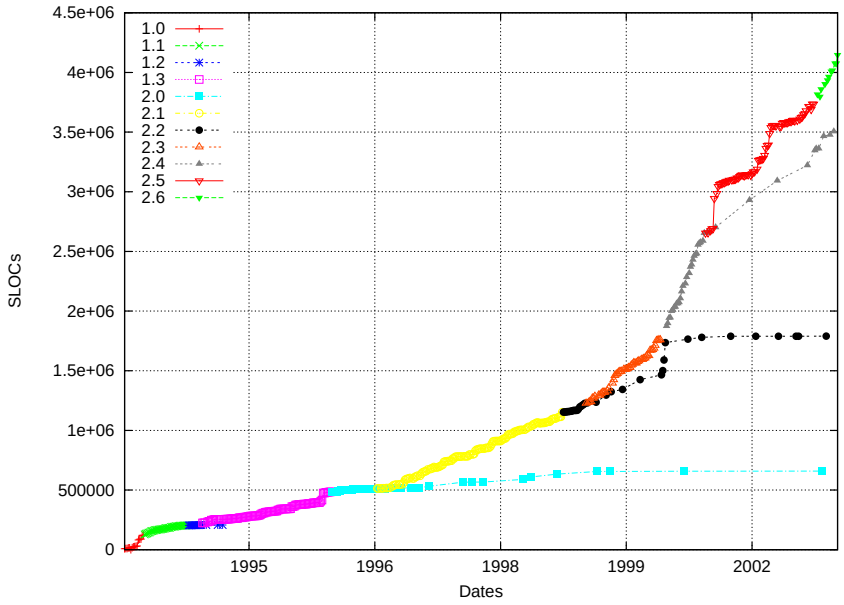
Example: source code analysis

- Data source: Version control system or source code official tarballs
- Metrics: SLOC, McCabe, ...)
- Classification of code: by language,...
- Contribution (eg. by author)
- Evolution of releases
- Learnt: structure of source code, languages used, developer activity

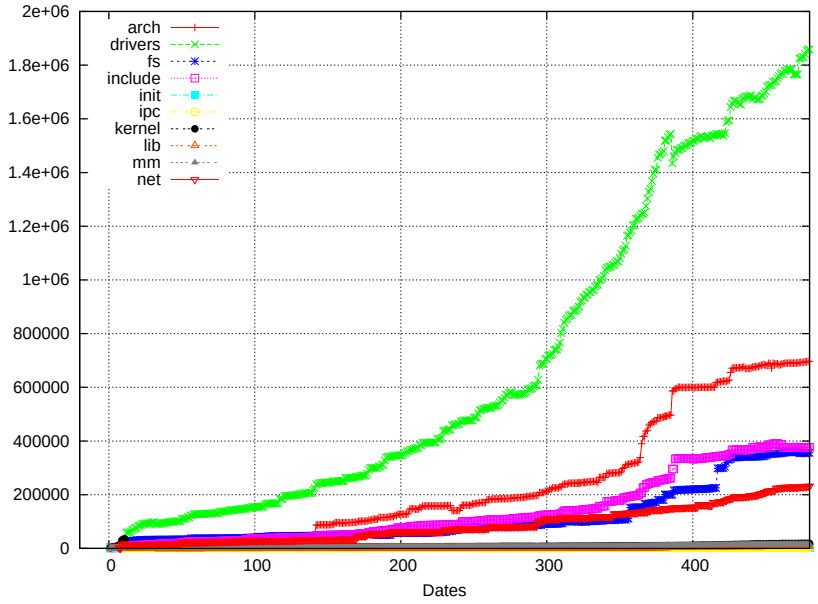
Example: Evolution on some projects



Linux: example of project growth



Linux: example of project growth

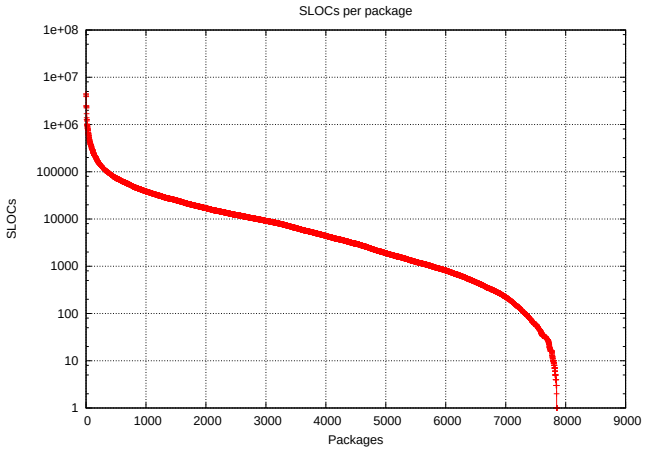


Studying data about a set of projects

Example: source code analysis from a GNU Distribution

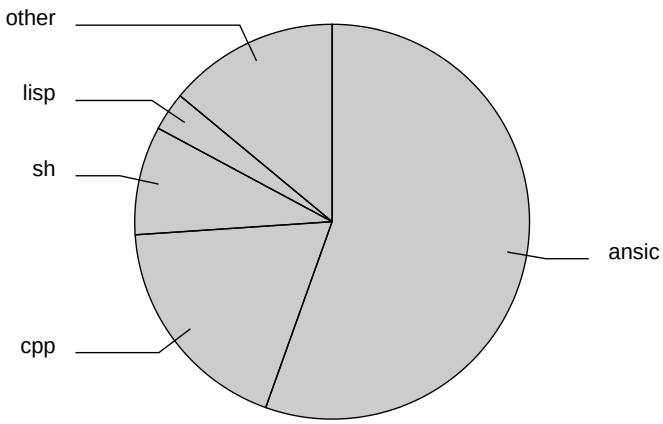
- Data source: Source packages of a GNU distro
- Packages are much more heterogeneous, so we can study:
  - Size distribution of packages.
  - Distribution on programming languages used.
  - Obtain some interesting evidences on evolution of GNU distros.

Size distribution of Debian Sarge Packages



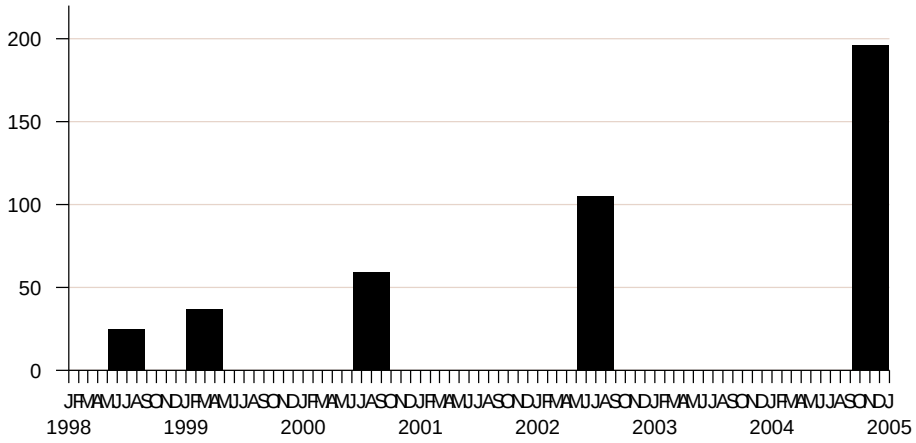
- More than 8000 packages.
- Mean package size: about 26 Kbytes.

Languages used on Debian Sarge



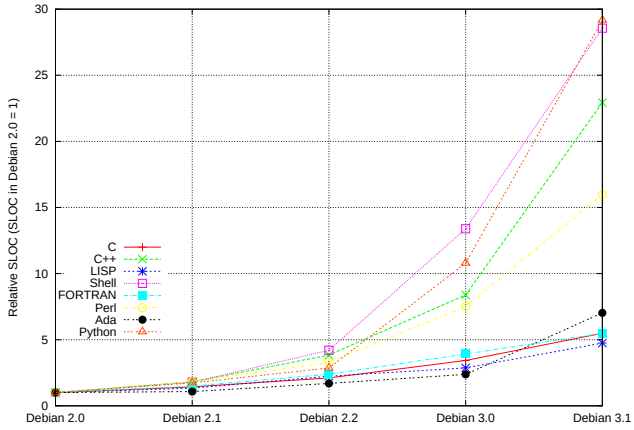
- Other languages used: python, java, assembler, fortran, ada, php... more than 20 programming languages detected.

Evolution of Debian releases



- Debian duplicates its size every two years.

Evolution of Languages used



- Although C is the most used language, its growth is not so important.
- Scripting languages such as Python grows quickly.

Another data source: Version control

Version control systems, such as CVS, are extensively used by developers and advanced users:

- Easy to use
- Access control (writable for developers, read-only for users)
- Code revision organized: revisions, branchs...
- For advanced users: access to latest (i.e. experimental) releases.

For us, CVS is an interesting data source:

- Access to source files of any release.
- Access to commit logs.
- This is a valuable information, not only about source code history, but also about developer activity.

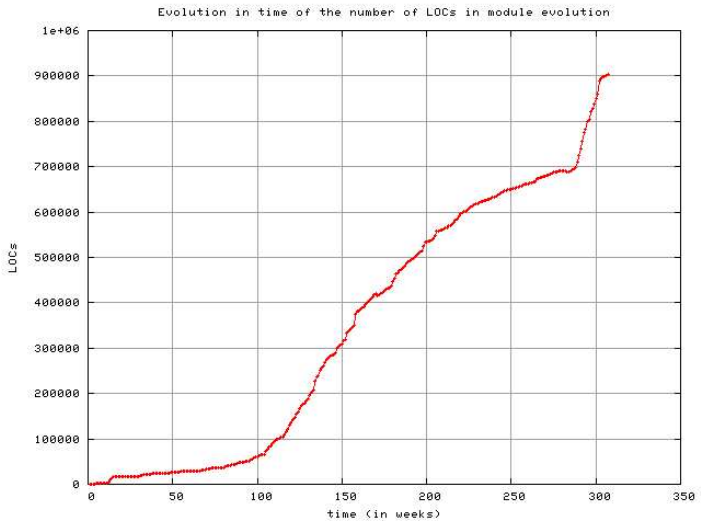
CVSAnaly sample results

CVSAnalY is a tool developed by URJC Libre Software team, which analyzes CVS logs and sources from a project.

As a sample, a summary result from Ximian Evolution.

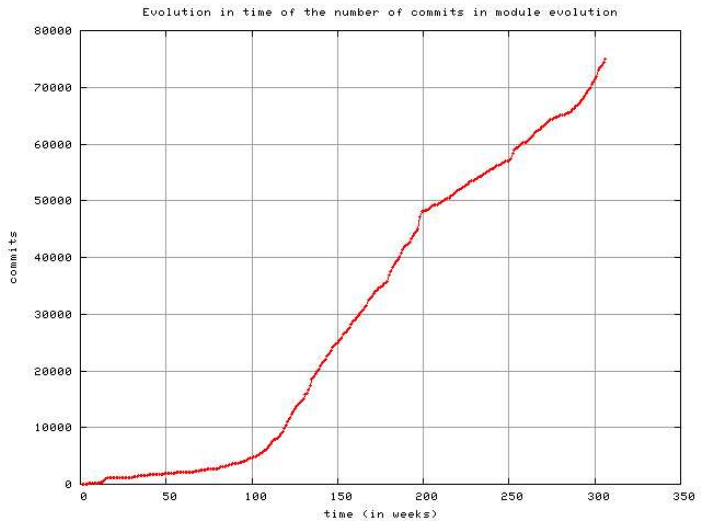
Commiters	190
Commits	88,157
Files	5,238
Lines Changed	16,411,471
Lines Added	9,360,719
Lines Removed	7,050,752
First Commit	1998-01-12
Last commit considered	2003-12-05

Software Growth in CVS releases



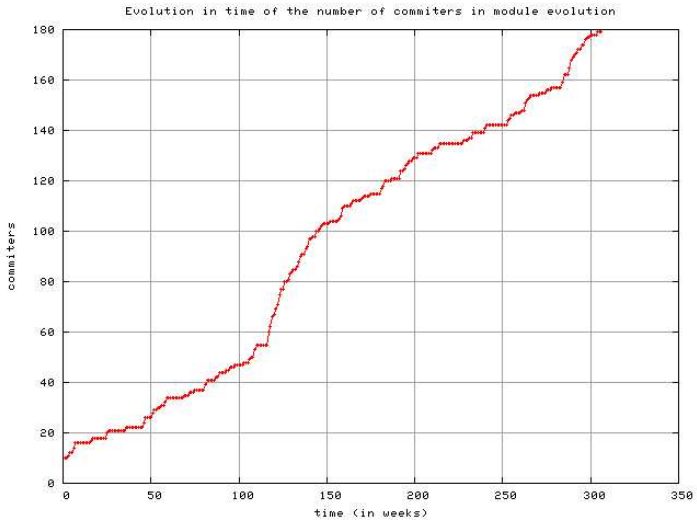
Data taken in years 1998-2003 from Ximian Evolution

CVS: Evolution in participation



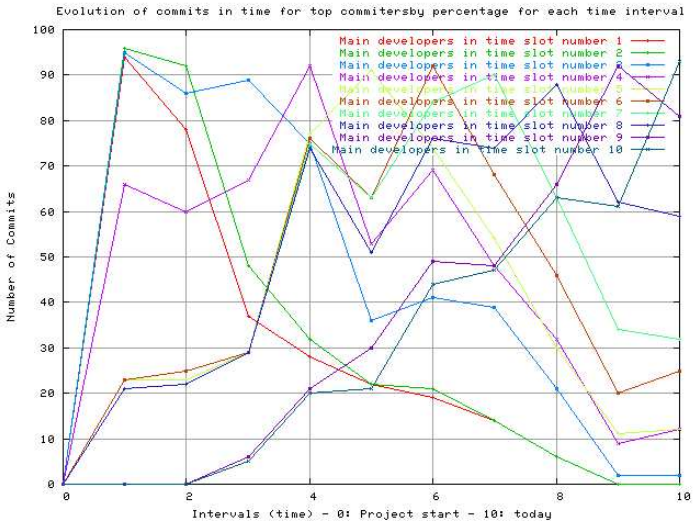
Data taken in years 1998-2003 from Ximian Evolution

CVS: Developer evolution



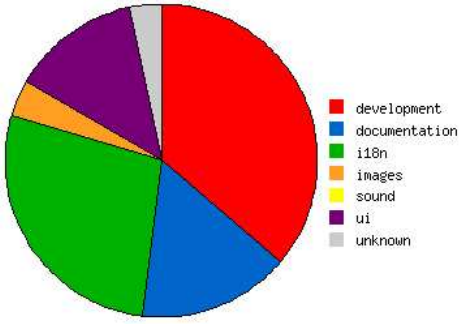
Data taken in years 1998-2003 from Ximian Evolution

CVS: Developer "generations"



Data taken in years 1997-2004 from KDE KDevelop

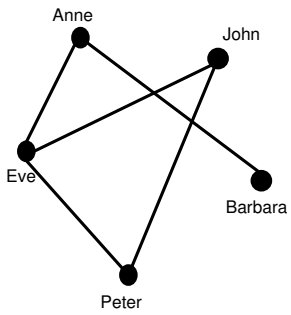
CVS: File discrimination



Data taken in years 1997-2004 from KDE CVS

Different contribution "classes". Can be analyzed for each commiter.

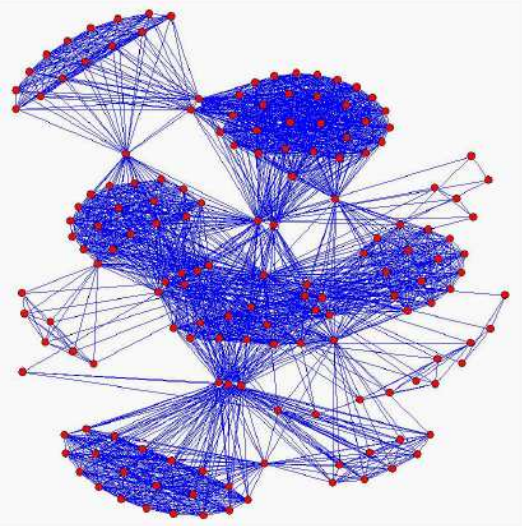
Social Network Analysis



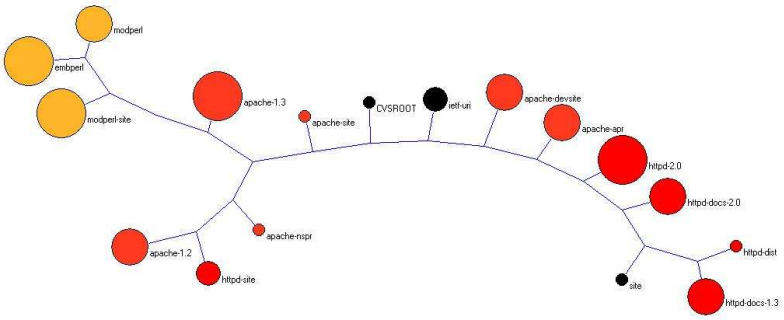
Social networks:

- Modelization: Vertices and relationships.
- Directed and undirected relationships.
- Study the network using well-known techniques.

Linux 1.0: personal relationships

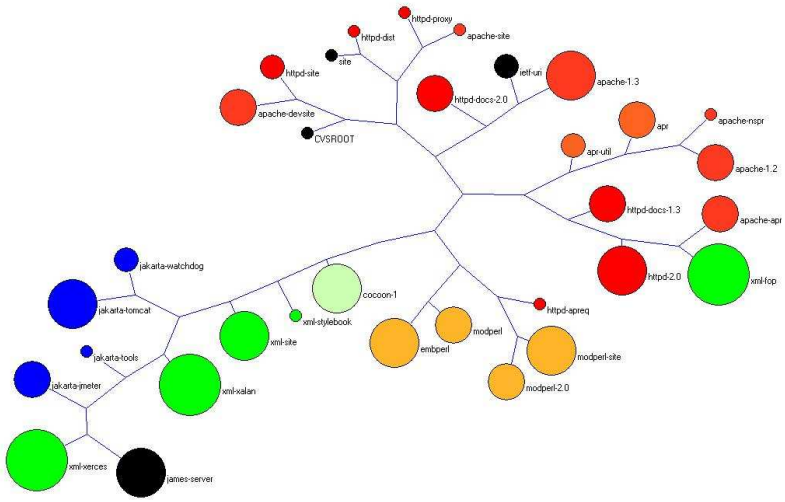


Community structure of the Apache project



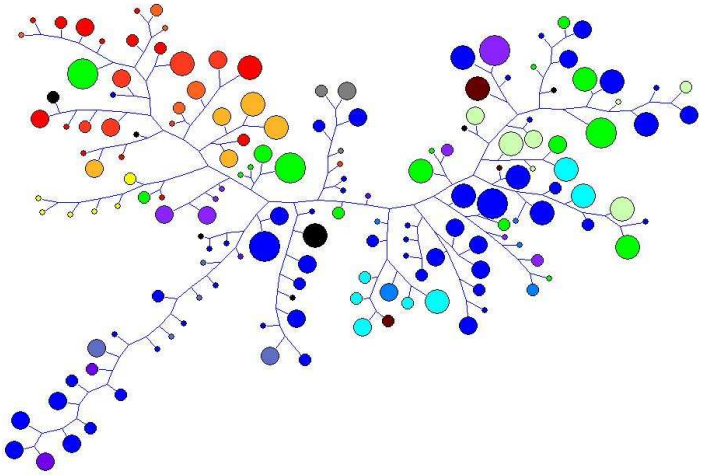
January 1st 1999

Community structure of the Apache project



January 1st 2000

Community structure of the Apache project



February 1st 2004

Conclusions

- Software Engineering in a quantitative manner is very interesting in Libre/Free Software projects, because of availability of lots of public data.
  - These data (usually) are not available when studying closed source projects.
- Libre/Free software projects appears to be behaviours different than traditionally projects of closed source/developed "in house".
  - Are the (classic) evolution laws valid for Open Source?

Learn more...

- Amor J., Robles G., Barahona J., López L.: *Toy Story: an analysis of the evolution of Debian GNU/Linux*
- Barahona J., López L., Robles G.: *Community structure of modules in the Apache project*
- Barahona J., Robles G.: *Free Software Engineering: A Field to Explore*
- López L., Robles G., Barahona J.: *Applying Social Network Analysis to the Information in CVS Repositories*
- Robles G., Koch S., Barahona J.: *Remote analysis and measurement of libre software systems by means of the CVSAnalY tool*
- Robles G., Barahona J.: *Results on Libre Software Engineering Research.* `http://libresoft.dat.escet.urjc.es/index.php?menu=Results`
- Robles G., Barahona J., Ghosh R.A.: *GlueTheos: Automating the Retrieval and Analysis of Data from Publicly Available Repositories*
- Robles G., Barahona J., Centeno J., Matellán V., Rodero L.: *Studying the evolution of libre software projects using publicly available data*

Most of these references available at:

`http://libresoft.dat.escet.urjc.es/`