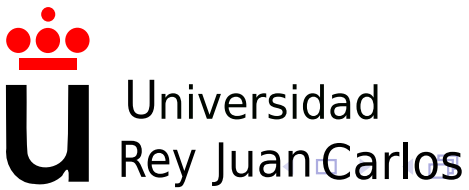
Using Social Network Analysis Techniques to Study Collaboration between a FLOSS Community and a Company

Juan Martínez-Romo, **Gregorio Robles**,
Jesús M. González-Barahona, Miguel Ortuño-Pérez

GSyC/Libresoft, Universidad Rey Juan Carlos – http://libresoft.es

OSS2008, Milan, September 8th 2008



Motivation

- High number of information from FLOSS projects available in the respositories
- Apply methodologies that highlight information for a given aspect
- In this presentation (paper):
  - Collaboration between FLOSS community and a company
  - Identification of leadership

FLOSS vs *corporate* software development

In corporate environments, we have:

- Formal organization, well defined (usually hierarchical) structure
- Clear guidelines about how to interact with each other
- Procedures and channels to use are known
- Each team of developers is assigned to certain modules of the system

All of this is not usually the case in FLOSS: large mount of spontaneous interaction structures arise, evolve and disappear without the intervention of a central control, yielding complex networks

Social Network Analysis (SNA)

- Apply classical social network concepts to data extracted from FLOSS projects, in particular to FLOSS projects that show a tight collaboration between a company and the FLOSS community
- Even in the presence of a company, FLOSS projects have a loose management style and are based heavily on third-party contributions
- Ad-hoc network of interpersonal dependencies that come from the interactions between the nodes that integrate it.
- Our work is mainly descriptive and has the purpose of showing how the application of techniques of social networks can be useful in scopes beyond the original ones

Methodology

- Retrieve data about the activity of developers from the source code management repository of the project
- For each commit (modification in a file in the repository): the date, the username of the developer (commiter), and the number of lines involved
- Developer network: Each vertex represents a particular developer; two vertices will be linked by an edge when both they have contributed to, at least, one common software artifact

Coordination degree

- Measures the ability of a vertex in a graph to interchange information
- Usefulness: shows the ability of a certain node to receive information of the network and capture the activity in a project precisely
- The total coordination degree of a vertex is a measure of the amount of information that the vertex is able to receive belonging to that particular network

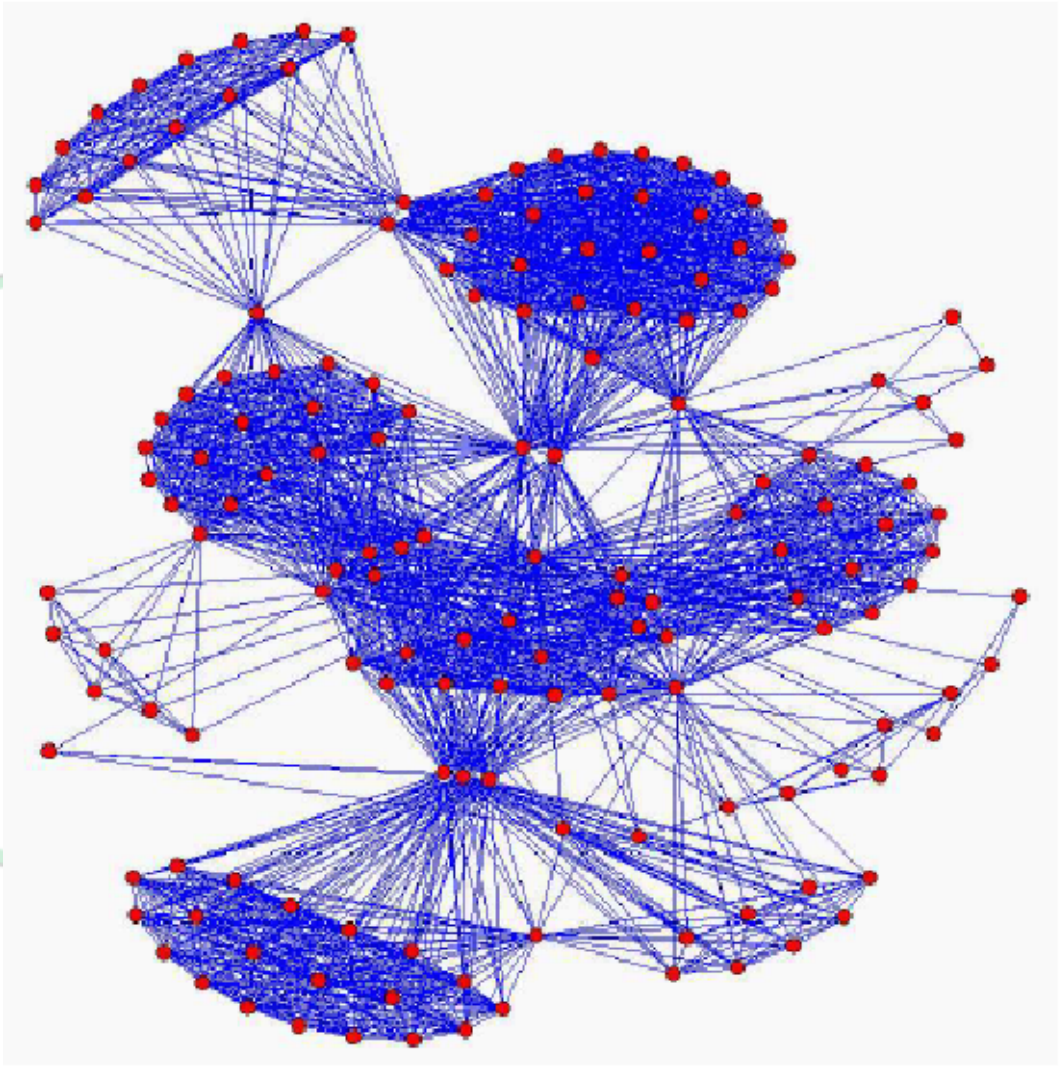
Betweenness centrality

- Measurement of the number of shortest paths traversing that particular vertex
- Usefulness: Information control that a developer can perform on a graph, in the sense that vertices with a high value are intermediate nodes for the communication of the rest
- In the SNA literature vertices with high betweenness centrality are known to cover *structural holes*. That is, those vertices glue together parts of the organisation that would be otherwise far away from each other
- They receive a diverse combination of information available to no one else in the network and have therefore a higher probability of being involved in the knowledge generation processes

Centrality Eigenvector

- Measurement of the importance of a node in a network
- Usefulness: assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes
- Even if a node influences just one other node, who subsequently influences many other nodes, then the first node in that chain is highly influential
- Good advice about leadership

An example (Linux 1.0)

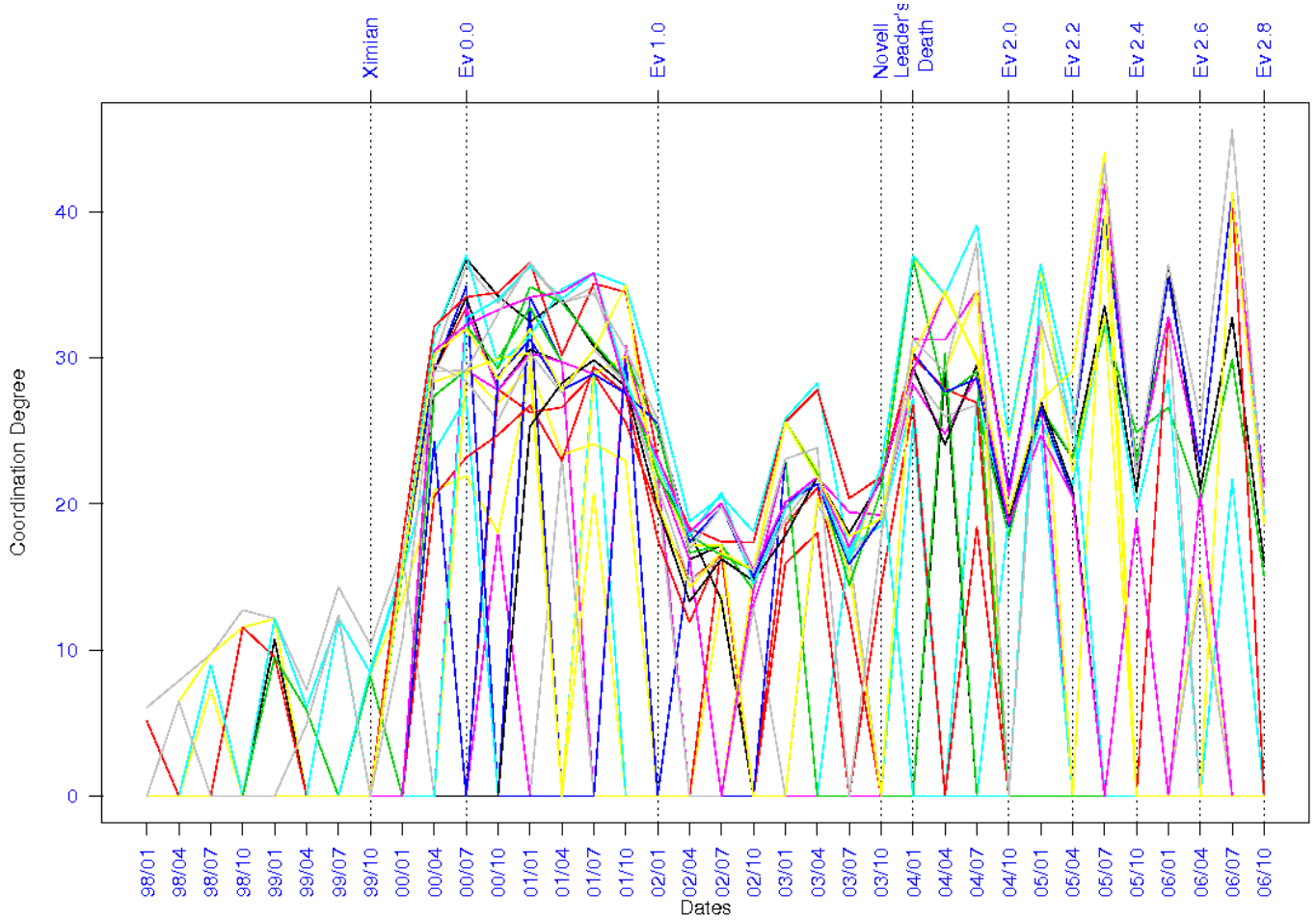


Evolution

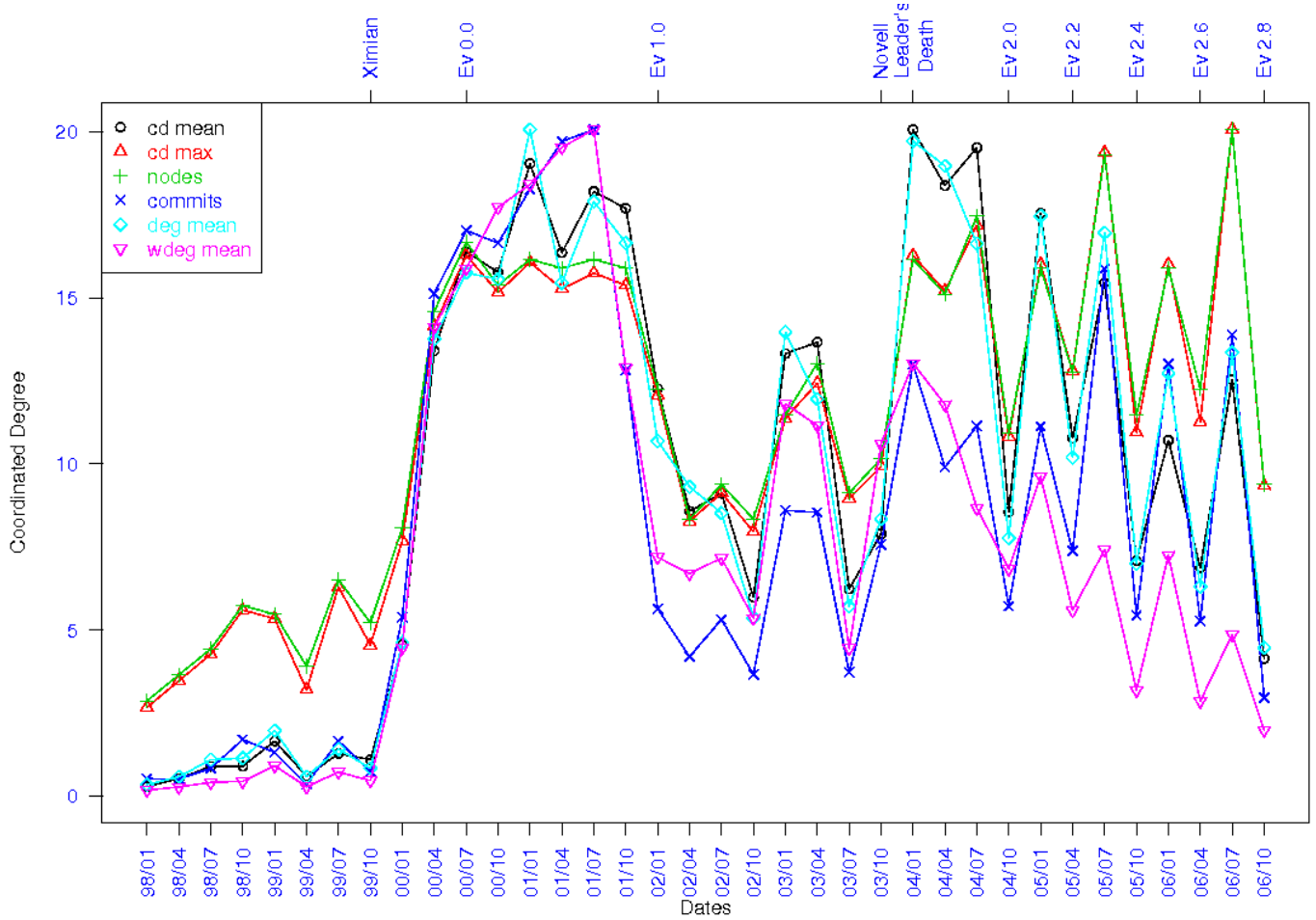
Event	Date
First commits	early 1998
Ximian takeover	October 1999
Version 0.0	July 2000
Version 1.0	February 2001
Takeover by Novell	October 2003
Death of main developer	January 2004
Version 2.0	October 2004
Version 2.2	April 2005
Version 2.4	October 2005
Version 2.6	April 2006
Version 2.8	October 2006

Cuadro: Most important events in the history of Evolution.

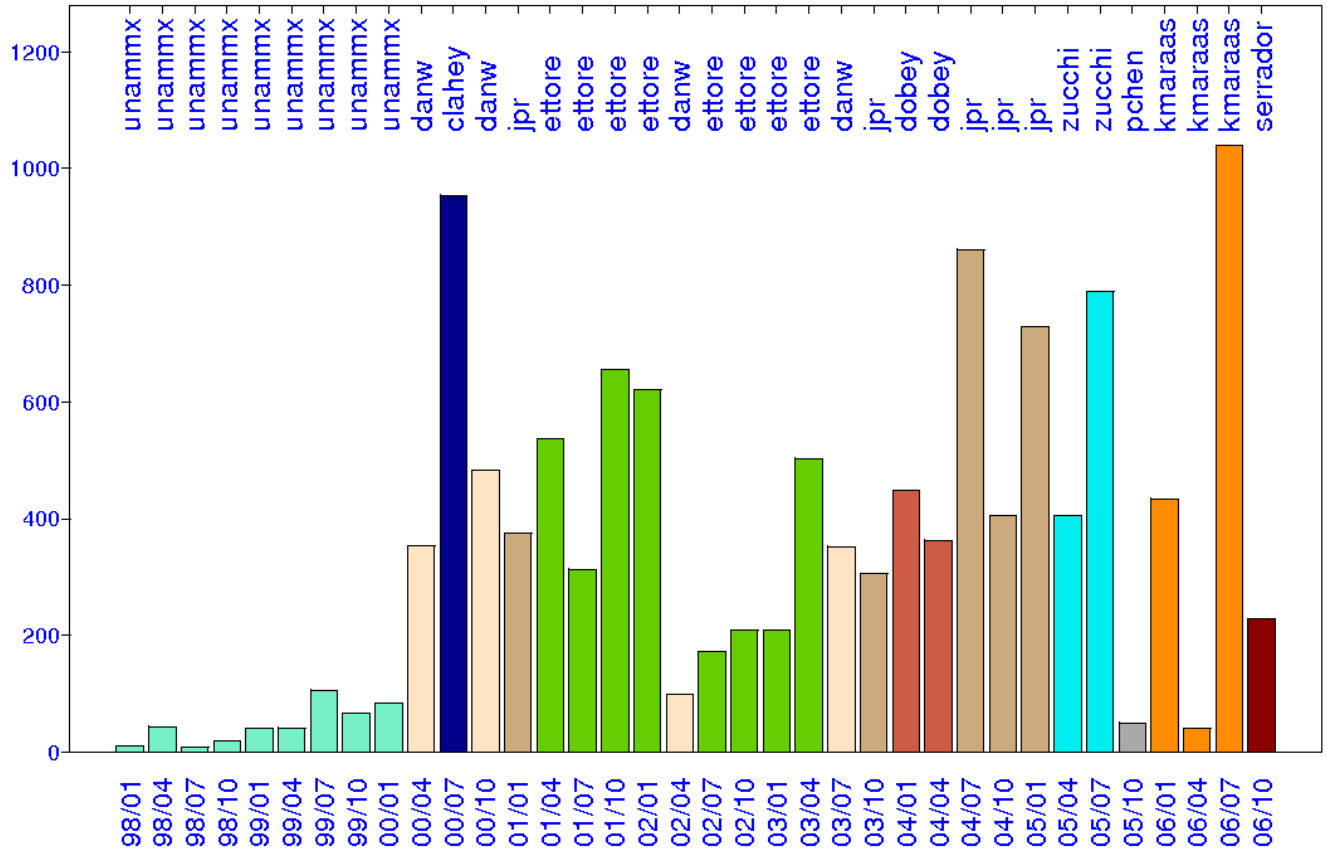
Coordination degree (top) (Evolution)



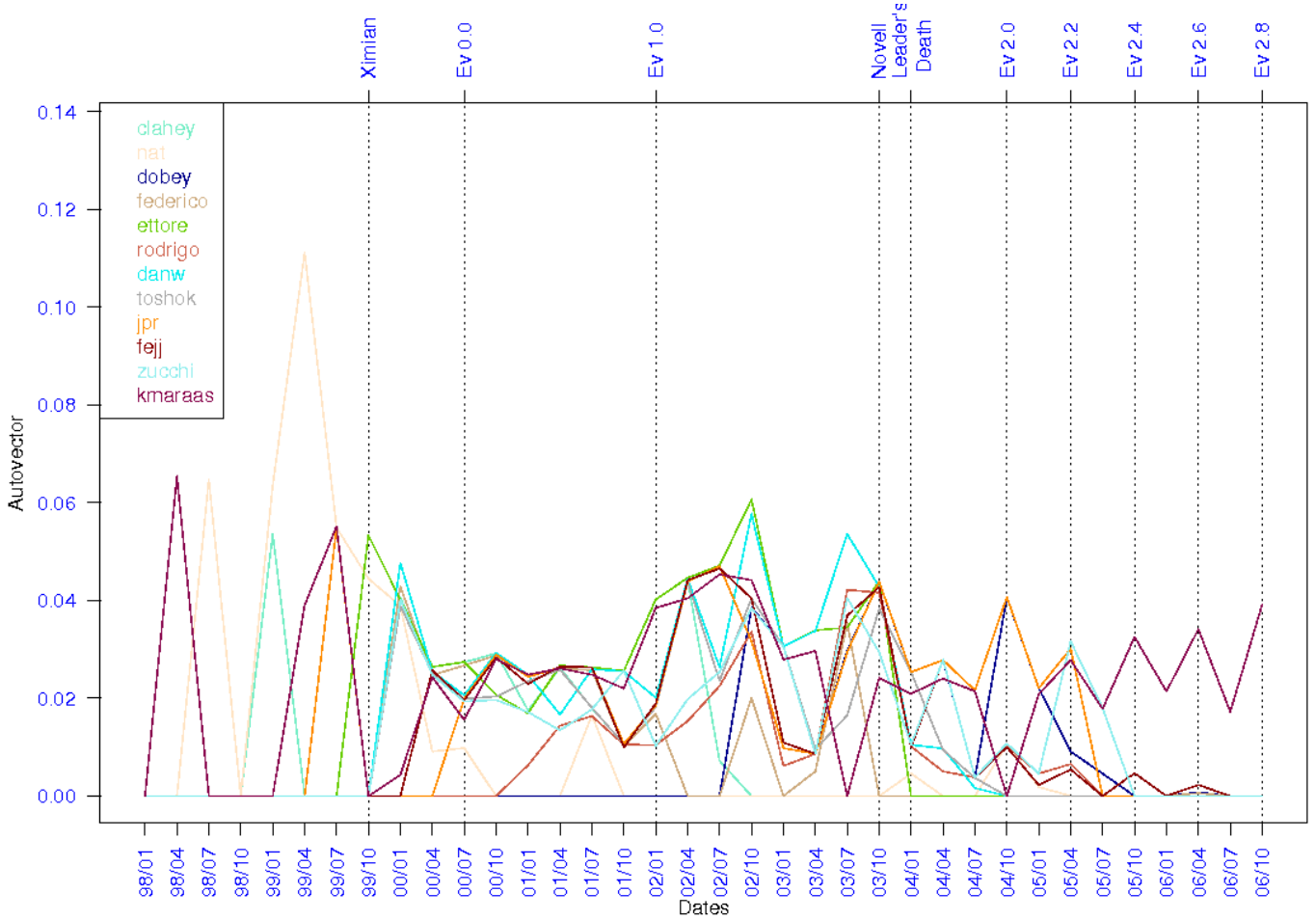
Coordination degree (mean vs active nodes) (Evolution)



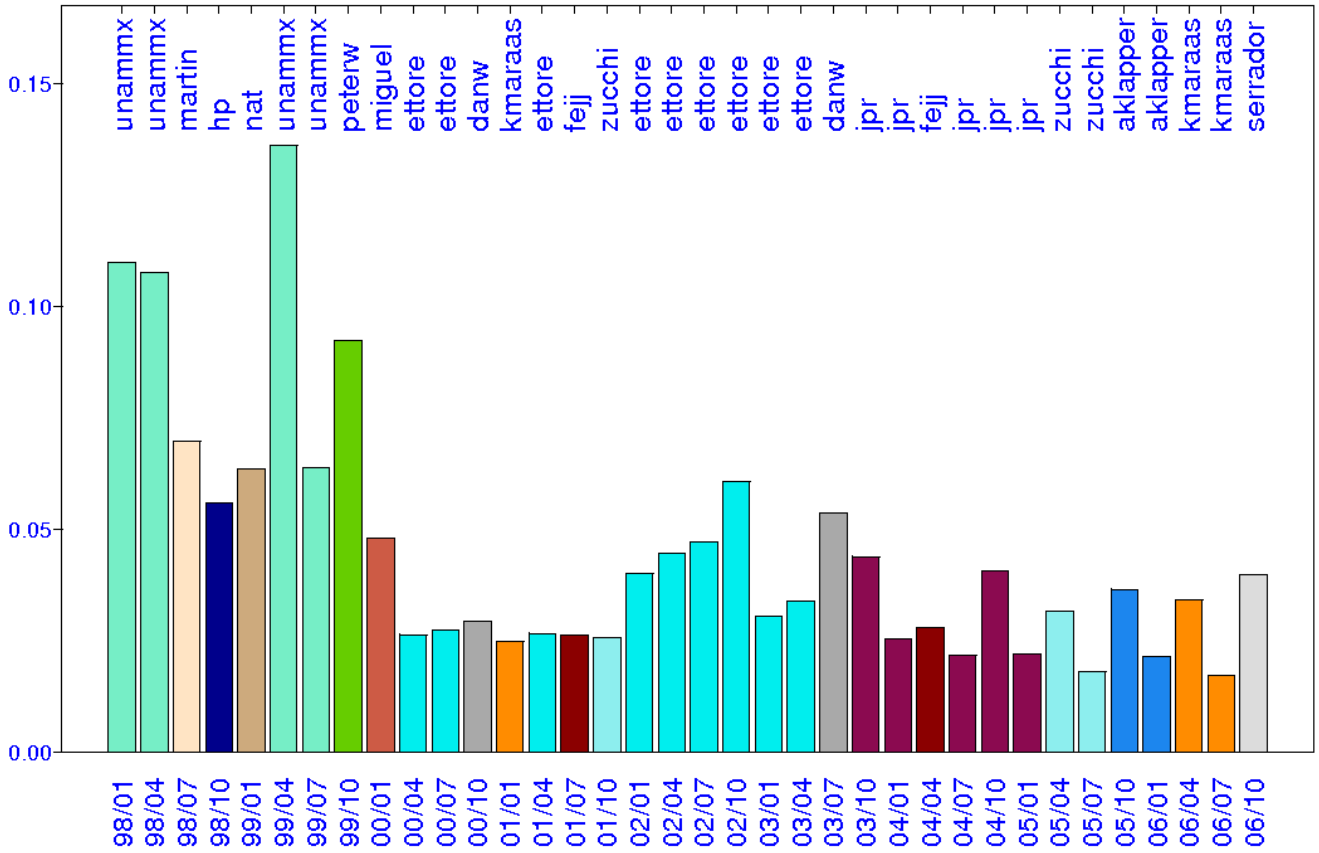
Weighted Betweenness comparative (Evolution)



Evolution of Eigenvalue (Evolution)



Evolution of Eigenvalue - Barchart (Evolution)

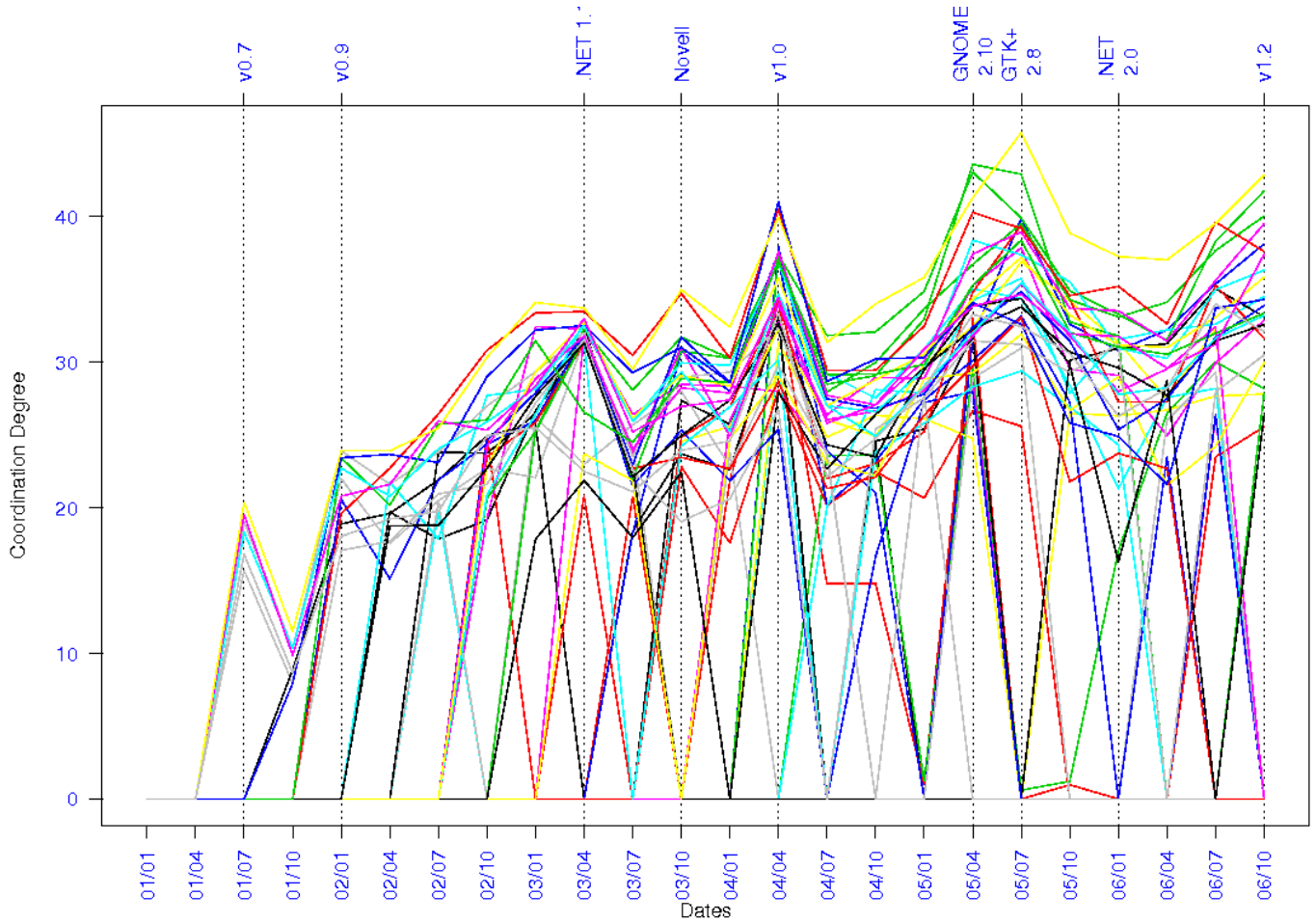


Mono

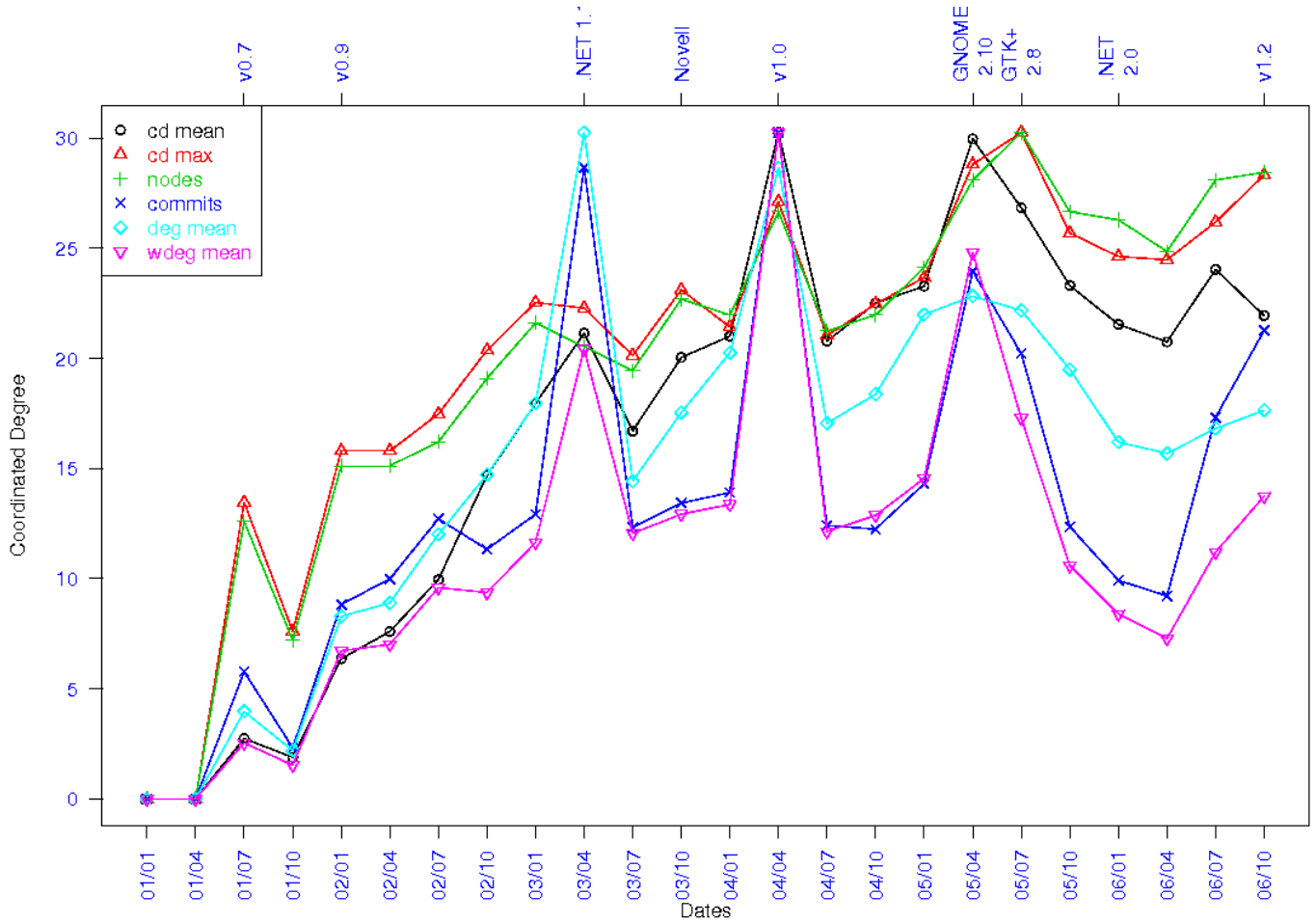
Event	Date
First commits	mid-2001
First version of Gtk# checked into CVS	September 2001
Version 0.7	September 2001
Version 0.9	February 2002
GTK+ 2.2	December 2002
.NET Framework 1.1	April 2003
Version 1.0	June 2004
GNOME 2.10	March 2005
GTK+ 2.8	August 2005
.NET Framework 2.0	November 2005
Version 1.2	November 2006

Cuadro: Most important events in the history of Mono.

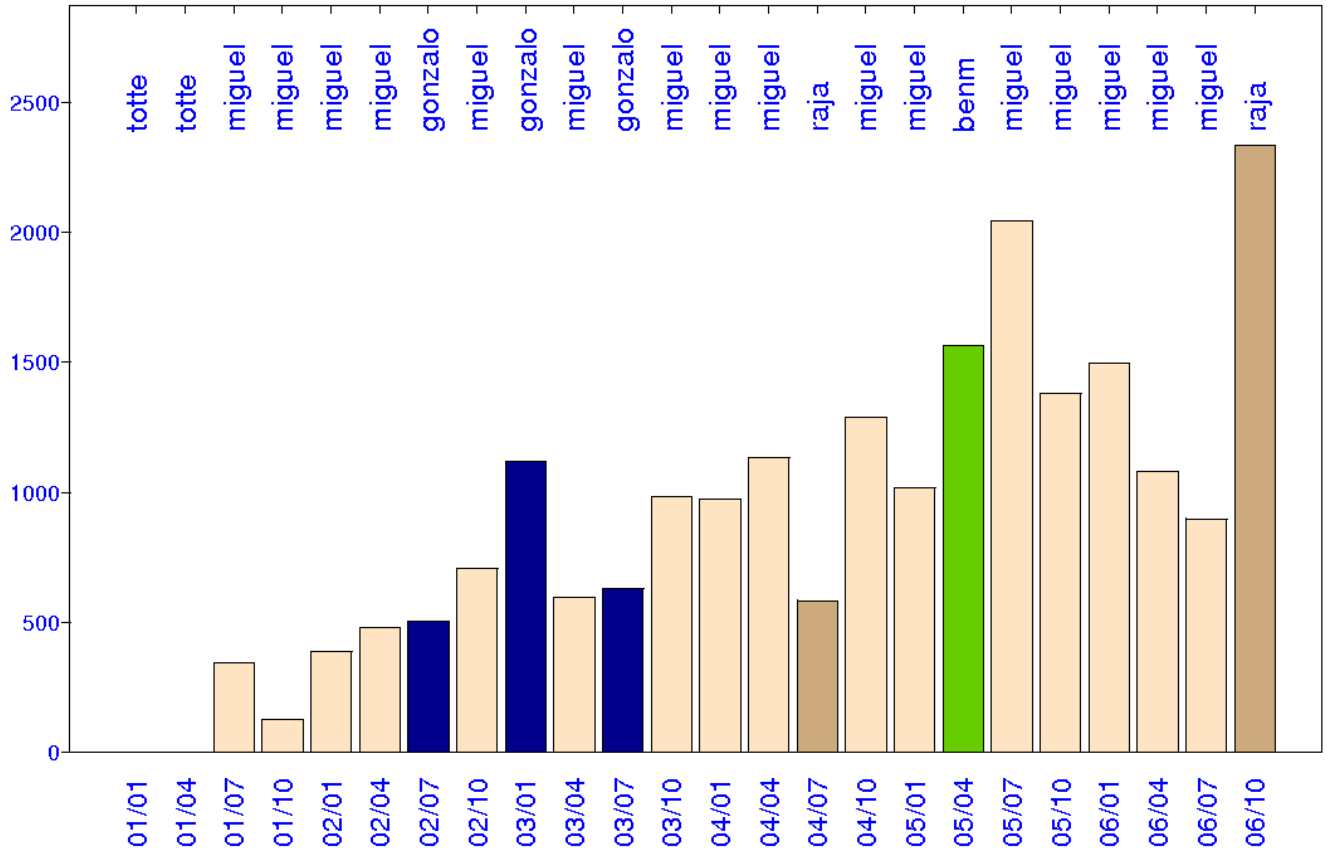
Coordination degree (top) (Mono)



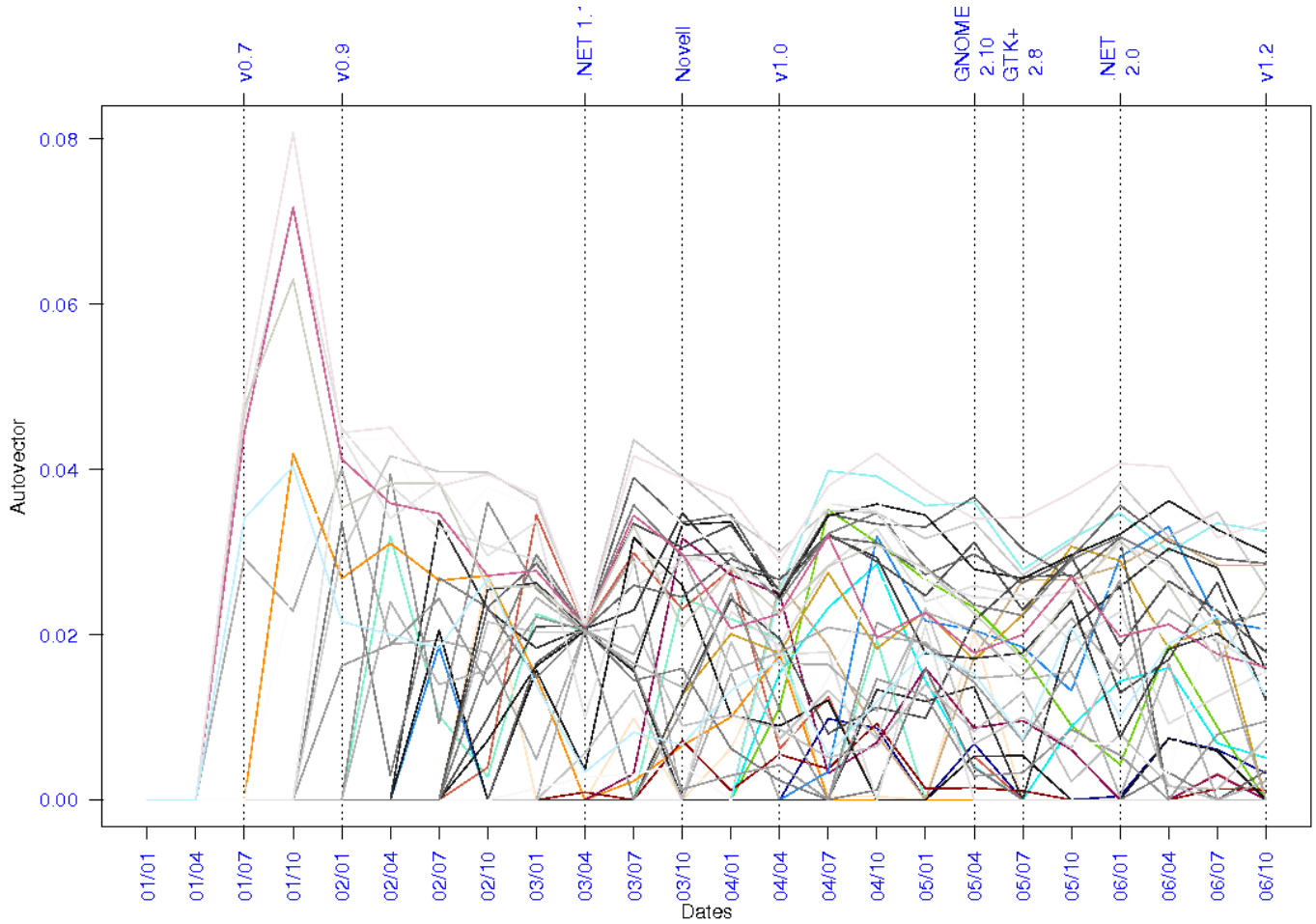
Coordination degree (mean vs active nodes) (Mono)

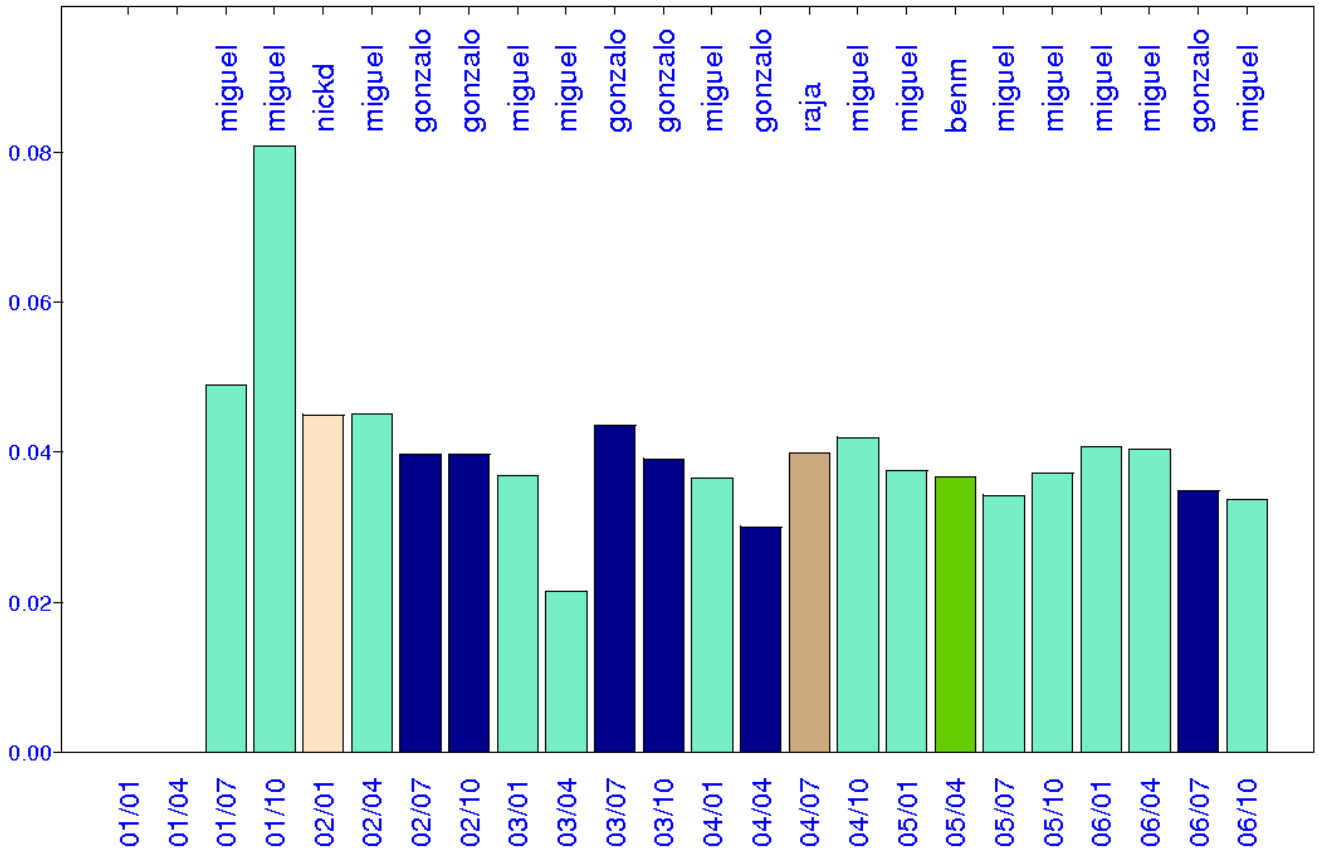


Weighted Betweenness comparative (Mono)



Evolution of Eigenvalue (Mono)




0.00
0.02
0.04
0.06
0.08
miguel
miguel
nickd
miguel
gonzalo
gonzalo
miguel
miguel
gonzalo
gonzalo
miguel
gonzalo
raja
miguel
miguel
benm
miguel
miguel
miguel
miguel
gonzalo
miguel
01/01
01/04
01/07
01/10
02/01
02/04
02/07
02/10
03/01
03/04
03/07
03/10
04/01
04/04
04/07
04/10
05/01
05/04
05/07
05/10
06/01
06/04
06/07
06/10

Conclusions and further work

- Descriptive study of several social networks of FLOSS projects where a company and the community collaborate
- We have identified the most important nodes (developers) in the project and have observed them from from several points of view (leadership, information control flows, etc.) and how they evolve over time
- Collaboration between the company and the community can, if properly handled, drive a more efficient development
- Time-based release management seems to animate the development of projects

Using Social Network Analysis Techniques to Study Collaboration between a FLOSS Community and a Company

Juan Martínez-Romo, **Gregorio Robles**,
Jesús M. González-Barahona, Miguel Ortuño-Pérez

GSyC/Libresoft, Universidad Rey Juan Carlos – http://libresoft.es

OSS2008, Milan, September 8th 2008