


 <p>Sponsored through Framework Programme Sixth (Call 5) by</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>		<b>Document Information</b>	
		<b>Version:</b> 1.0 <b>Date :</b> 9 Dec 09 <b>Pages :</b> 11	
		<b>Owning Partner:</b> URJC	
		<b>Author(s):</b> Daniel Izquierdo-Cortazar	
		<b>Reviewer(s):</b> Paul Adams Xavier Heymans	
		<b>To:</b> CONSORTIUM	
		<b>Purpose of distribution:</b>	
The QUALOSS Consortium consists of: CETIC (BE), Facultés Notre Dame de la Paix à Namur (BE), Universidad Rey Juan Carlos (ES), Fraunhofer IESE (DE), ZEA Partners (BE), MERIT (NL), AdaCore (FR), PEPITe (BE)		<b>Printed on at</b>	
<b>Status:</b> <input type="checkbox"/> Draft <input type="checkbox"/> To be reviewed <input type="checkbox"/> Proposal <input checked="" type="checkbox"/> Final/Released	<b>Confidentiality:</b> <input checked="" type="checkbox"/> Public - Intended for public use <input type="checkbox"/> Restricted - Intended for E.C. Program participants <input type="checkbox"/> Confidential - Intended for individual partner only		
<b>Deliverable ID:</b> D6.4  <b>Title:</b>  <div style="text-align: center;">Extra Materials</div>			
<b>License for distribution:</b>  This work is licensed under a <a href="http://creativecommons.org/licenses/by-sa/2.5/">Creative Commons Attribution-Share Alike 2.5 License</a> . (The license can be found in <a href="http://creativecommons.org/licenses/by-sa/2.5/">http://creativecommons.org/licenses/by-sa/2.5/</a> )  The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.			


	Extra Materials  Deliverable ID: D6.4	Page : 2 of 11
		Version: 1.0 Date: Dec 9, 09
		Status : Proposal Confid : Public

**Deliverable: D6.4**

**Title: Extra Materials**

**Executive Summary:**

This report explores the different materials created during the QualOSS' life regarding to teaching assistance, specific courses related to quality of open source software.


	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	Page : 3 of 11
		Version: 1.0 Date: Dec 9, 09
		Status : Proposal Confid : Public

## CHANGE LOG

Ver.	Date	Author	Description
0.1	30.11.2009	Daniel Izquierdo-Cortazar	Initial Version
0.2	8.12.2009	Paul Adams	First Review
1.0	8.12.2009	Daniel Izquierdo-Cortazar	Final Version


## APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Deliverable Identification

	Extra Materials  Deliverable ID: D6.4	Page : 4 of 11
		Version: 1.0 Date: Dec 9, 09
		Status : Proposal Confid : Public


## TABLE OF CONTENTS

<b>1. Introduction.....</b>	<b>5</b>
<b>2. Materials.....</b>	<b>6</b>
2.1 Master on Free Software.....	6
2.2 "Libresoft Tools Tutorial" Slides.....	7
2.3 "QualOSS" Slides.....	8
2.4 "Introduction to OpenBRR" Slides.....	9
2.5 "Comparison QsoS and OpenBRR" Slides.....	10
2.6 "Qualité Logicielle: Introduction" Slides.....	11

	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 5 of 11</p>
		<p>Version: 1.0 Date: Dec 9, 09</p>
		<p>Status : Proposal Confid : Public</p>

## 1. INTRODUCTION


There are several partners which are directly related to education. This provides a new point of view in terms of possible extra materials indirectly related to the main purpose of QualOSS. This is for instance those slides or tutorials which aim to explain the QualOSS methodology (or indirectly use it). However they are not included as a main part of the QualOSS consortium final deliverables, databases or technical reports. One of the main areas is the education at university level. Some of the partners have a main research and educational side, which helps to retrieve all the materials at this level.

	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 6 of 11</p> <hr/> <p>Version: 1.0</p> <p>Date: Dec 9, 09</p> <hr/> <p>Status : Proposal</p> <p>Confid : Public</p>
---	--	--

## 2.MATERIALS

### 2.1 MASTER ON FREE SOFTWARE

This master is lead by the Universidad Rey Juan Carlos in Madrid. It aims to provide a general vision about the libre software world, and foster the skills, from an industrial point of view, of the potential students regarding their interactivity between the industry and the libre software communities.

	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 7 of 11</p> <hr/> <p>Version: 1.0 Date: Dec 9, 09</p> <hr/> <p>Status : Proposal Confid : Public</p>
---	--	--

## 2.2“LIBRESOFT TOOLS TUTORIAL” SLIDES



---

## MASTER ON FREE SOFTWARE 2009/2010

### PRACTICAL APPROACH:

### ANALYSING LIBRE SOFTWARE COMMUNITIES (v0.3)

(CC) 2009 DANIEL IZQUIERDO CORTÁZAR,  
TEÓFILO ROMERA OTERO

SOME RIGHTS RESERVED. THIS DOCUMENT IS DISTRIBUTED UNDER THE CREATIVE COMMONS ATTRIBUTION  
SHAREALIKE 2.5 LICENCE, AVAILABLE IN  
[HTTP://CREATIVECOMMONS.ORG/LICENSES/BY-SA/2.5/](http://creativecommons.org/licenses/by-sa/2.5/)

---

#### TABLE OF CONTENTS

<b>SUMMARY.....</b>	<b>3</b>
<b>REQUIREMENTS:.....</b>	<b>3</b>
<b>EXPERTISE ACQUIRED AFTER THE TUTORIAL.....</b>	<b>3</b>
<b>MANDATORY PACKAGES.....</b>	<b>3</b>
<b>INTRODUCTION.....</b>	<b>5</b>
<b>CVSANALY.....</b>	<b>6</b>
Options:.....	6
Basic Usage.....	7
Database Schema.....	8
Basic Queries.....	10
<b>MAILINGLISTSTATS.....</b>	<b>12</b>
Options:.....	12
Basic Usage.....	13
Database Schema.....	17
Basic Queries.....	19
<b>BICHO.....</b>	<b>20</b>
Options.....	20
Basic Usage.....	21



*FLOSSMetrics: Creating the Database*.....21

*Basic Queries*.....24

***SLOCCOUNT***.....25

***GUILTY***.....27

## ***SUMMARY***

In this exercise, we are going to learn how use some useful tools in order to study libre software communities. We will use some tools such as CVSanaly or MailingListStats from the LibreSoft-tools suite, and some others that are available as libre software and made by the community such as SLOCCount by David Wheeler.

## ***REQUIREMENTS:***

Prior to following this tutorial. It is advisable that the student manages the following skills:

- Shell environment
- Being able to import MySQL databases
- Basic SQL for queries and management of the databases
- Additional packages installation (manual installation from sources)

## ***EXPERTISE ACQUIRED AFTER THE TUTORIAL***

The main objective of this tutorial is to introduce the students briefly to the tools and mechanisms that they can use to further explore about libre software communities. They should learn about the following topics:

- General data available in the publicly available repositories found in libre software projects.
- The basic usage of the tools.
- Libre software community data which allows to have a first glimpse about their composition and evolution.

## ***MANDATORY PACKAGES***

The following packages are needed. Most of them are included in almost any GNU/Linux distribution, for example in a Debian-based distribution you should be able to *apt-get install* all of them:

- mysql-server
- python
- python-dev
- python-mysqldb
- python-beautifulsoup
- sloccount
- automake
- autoconf
- git-core

The following package is an exception and it should be installed manually:

- Storm

You can get it with wget:

```
wget http://launchpad.net/storm/trunk/0.15/+download/storm-0.15.tar.bz2
```

But the hassle should not be more than doing something like:

```
python setup.py install
```

## Installing The Tools

CVSAnalY (including repositoryhandler) is one of the tools in the LibreSoft tools suite. It should be installed manually from the GIT repository it is being developed on.

Obtain the code:

```
git clone git://git.libresoft.es/git/cvsanaly
git clone git://git.libresoft.es/git/repositoryhandler
```

Install Repository Handler

```
./autogen.sh && make && sudo make install
```

Install CVSAnalY2

```
sudo python ./setup.py install
```

You can check that everything is working fine with

```
cvsanaly2 --help
```

Mailing List Stats, also from the LibreSoft suite will help you analysing mailing lists.

Obtain MLStats

```
wget http://forge.morfeo-project.org/frs/download.php/260/mlstats-0.3.3.tar.gz
```

Uncompress it

```
tar -xzf (or use ump and forget about formats forever)
```

Install it manually

```
sudo python ./setup.py install
```

Bicho is yet other of the LibreSoft tools. It works on BTS data mining.

Obtain Bicho

```
svn co https://svn.forge.morfeo-project.org/libresoft-tools/bicho/
```

Install Bicho

```
sudo python ./setup.py install
```

**Note:** Bicho is heavily depending on html parsing and also has a slow pace for retrieving web pages without causing DoS to the servers. So it can take very long to run the data gathering process. In shake of agility we will use previously generated FLOSSMetrics databases instead of running Bicho.

At last, we may have a short introduction to the newest tool in the LibreSoft suite, Guilty.

As above, obtain

```
git clone git://git.libresoft.es/git/guilty
```

Install Guilty

```
./autogen.sh && make && sudo make install
```

## ***INTRODUCTION***

FLOSSMetrics (fm3) is a research project funded by the European Commission which aims to recollect data from thousands of libre software projects. Data is collected in a periodic way, and it contains information from the source code management system (SCM), bug tracking systems (BTS), source code metrics and mailing lists. There are other projects which also provide this kind of information like FLOSSMole, but fm3 contains more low level data (for instance, commits made in each project). More information can be found at <http://flossmetrics.org> and databases can be found at <http://melquiades.flossmetrics.org>.

The usual way to work for fm3 is based on the process of data retrieval. To achieve this, several tools are used. In order to centralize all the process the Retrieval System tool was created. It controls from the URLs detection, the retrieval of the data and the final storage in a MySQL database.

In this practical approach we are going to work with the tools in order to analyse the structure and evolution of different libre software communities.

## CVSAnalY

CVSAnalY is a tool which stores the log from a SCM (CVS, SVN or Git) in a MySQL database.

### Options:

```
dizquierdo@rupia:~$ cvsanaly2 --help
cvsanaly2 2.0.0 - An analysis tool for your source code repository
Copyright (C) 2004-2009 LibreSoft <libresoft-tools-devel@lists.morfeo-project.org>
```

Usage: cvsanaly2 [options] [URI]

Analyze the given URI. An URI can be a checked out directory, or a remote URL pointing to a repository. If URI is omitted, the current working directory will be used as a checked out directory.

### Options:

-h, --help	Print this usage message.
-V, --version	Show version
-g, --debug	Enable debug mode
-q, --quiet	Run silently, only print error messages
--profile	Enable profiling mode
-f, --config-file	Use a custom configuration file
-l, --repo-logfile=path repository	Logfile to use instead of getting log from the repository
-s, --save-logfile[=path]	Save the repository log to the given path
-n, --no-parse	Skip the parsing process. It only makes sense in conjunction with --extensions
--extensions=ext1,ext2,	List of extensions to run

### Database:

--db-driver	Output database driver [mysql sqlite] (mysql)
-u, --db-user	Database user name (operator)
-p, --db-password	Database user password
-d, --db-database	Database name (cvsanaly)

`-H, --db-hostname` Name of the host where database server is running  
(localhost)

Metrics Options:

`--metrics-all` Get metrics for every revision, not only for HEAD

If you can run the option `--help`, it means that you correctly installed it. Once here, we need to stress several interesting options. On the one hand, the basic usage and data storage and on the other hand the use of extensions.

### ***Basic Usage***

We will need firstly a database to hold the data obtained with CVSanaly.

Create the database:

```
mysqladmin -u root -proot create cvsanaly
```

Next, we initiate the analysis, for instance, with the GNOME document viewer, Evince:

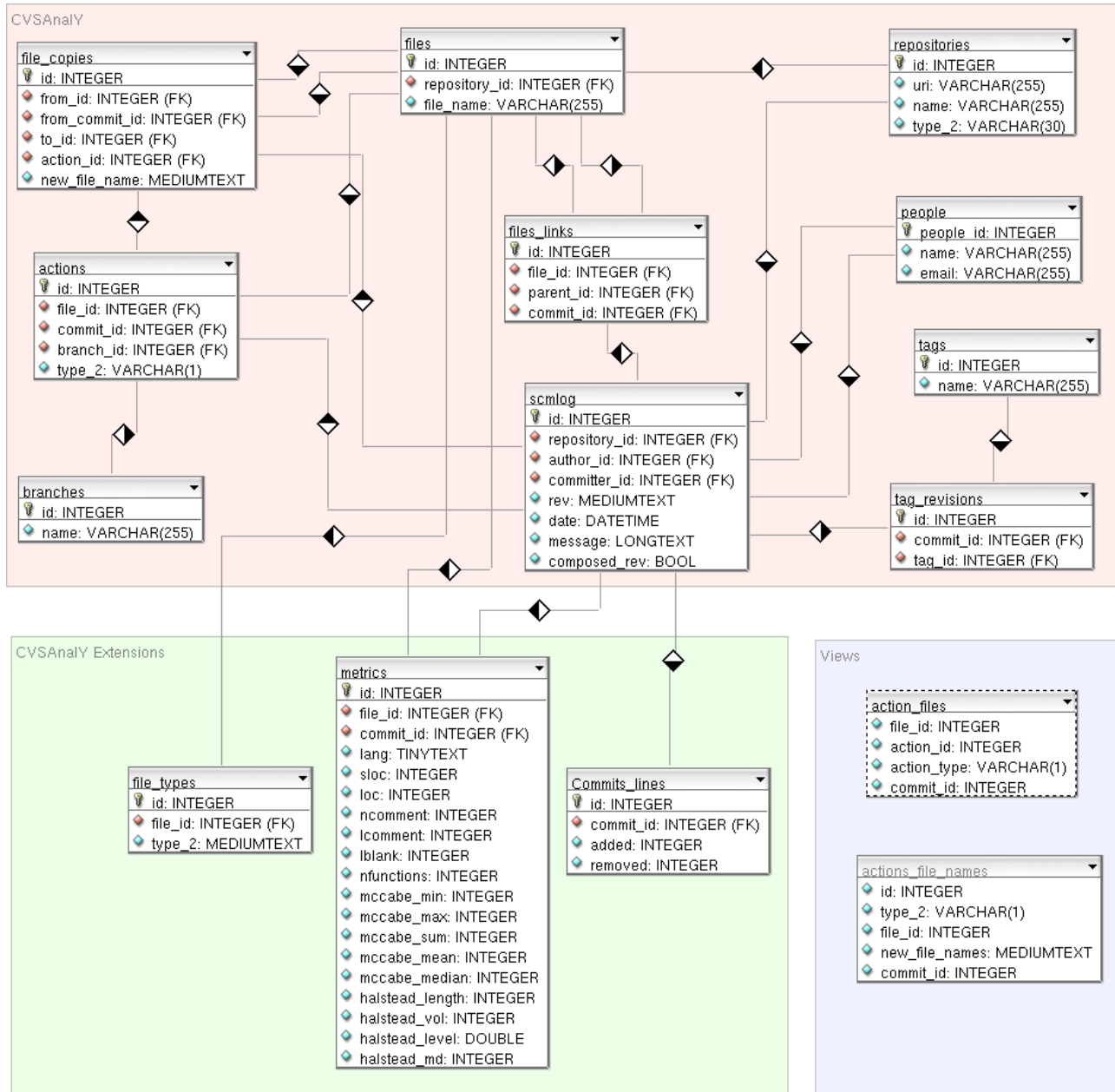
```
cvsanaly2 --db-user root --db-password root http://svn.gnome.org/svn/evince/trunk/
```

**Note:** Please, take into account that by default, the password is root, however you have to introduce yours.

**Note:** Just in case you want to add the data to some other database different from the chosen one, you have to specify this in the options: `--db-database` and next, the name, for instance: `cvsanaly2_evince`

## Database Schema

So far, we have prepared the database to work with it. Now, we are going to have a look at the schema.



Accessing the MySQL database:

```
mysql -u root -proot -A cvsanaly
```

**Note:** With the -A option, we specify which database we want to access directly.

Now we see the tables found in the schema:

```
mysql> show tables;
+-----+
| Tables_in_cvsanaly |
+-----+
| action_files      |
| actions           |
| actions_file_names |
| branches          |
| file_copies       |
| file_links        |
| files             |
| people            |
| repositories      |
| scmlog            |
| tag_revisions     |
| tags              |
+-----+
12 rows in set (0.00 sec)
```

We want to focus on the tables *actions*, *files*, *people* and *scmlog*. Lets have a closer look at them.

Table actions:

```
mysql> desc actions;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | 0        |       |
| type       | varchar(1)| YES  |     | NULL     |       |
| file_id    | int(11)   | YES  | MUL | NULL     |       |
| commit_id  | int(11)   | YES  | MUL | NULL     |       |
| branch_id  | int(11)   | YES  | MUL | NULL     |       |
+-----+-----+-----+-----+-----+-----+
```

Table files:

```
mysql> desc files;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
```



id	int(11)	NO	PRI	NULL		
file_name	varchar(255)	YES	MUL	NULL		
repository_id	int(11)	YES	MUL	NULL		

### Table people:

```
mysql> desc people;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
name	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	

### Table scmlog:

```
mysql> desc scmlog;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
rev	mediumtext	YES		NULL	
committer_id	int(11)	YES	MUL	NULL	
author_id	int(11)	YES	MUL	NULL	
date	datetime	YES		NULL	
message	longtext	YES		NULL	
composed_rev	tinyint(1)	YES		NULL	
repository_id	int(11)	YES	MUL	NULL	

## Basic Queries

### Total Number of Commits

```
mysql> select count(*) from scmlog;
```

count(*)
----------

```
|      3214 |
+-----+
```

### Total Number of Committers

```
mysql> select count(distinct(committer_id)) from scmlog;
```

```
+-----+
| count(distinct(committer_id)) |
+-----+
|                        203 |
+-----+
```

### Total Number of Commits group by year and month

```
mysql> select year(date), month(date), count(*) from scmlog group by year(date),
month(date) order by year(date), month(date);
```

```
+-----+-----+-----+
| year(date) | month(date) | count(*) |
+-----+-----+-----+
|      1999 |      4 |      2 |
|      1999 |      5 |      6 |
|      1999 |      6 |      3 |
|      1999 |      7 |      5 |
|      1999 |      8 |     39 |
|      1999 |      9 |      5 |
|      1999 |     10 |      6 |
|      1999 |     11 |      1 |
|      1999 |     12 |      1 |
|      2000 |      2 |      6 |
|      2000 |      3 |      1 |
|      2000 |      5 |      4 |
```

```
.....
.....
.....
```

```
|      2008 |      6 |     13 |
|      2008 |      7 |     29 |
|      2008 |      8 |     41 |
|      2008 |      9 |     75 |
|      2008 |     10 |     22 |
```

```

|      2008 |      11 |      30 |
|      2008 |      12 |      35 |
|      2009 |       1 |      74 |
|      2009 |       2 |     104 |
|      2009 |       3 |      60 |
|      2009 |       4 |      30 |
+-----+-----+-----+
93 rows in set (0.01 sec)

```

## ***MAILINGLISTSTATS***

This tool stores mbox-formated mailing lists information in a MySQL database.

### ***Options:***

```

mlstats 0.3.3 - Libresoft Research Group http://libresoft.urjc.es
(C) 2007-2009 Libresoft <libresoft@gsyc.escet.urjc.es>

```

```
Usage: ./mlstats [options] [URL1] [URL2] ... [URLn]
```

where URL1, URL2, ....., URLn are the urls of the archive web pages of the mailing list.

If they are a local dir instead of a remote url,

the directory will be recursively scanned for mbox files.

If the option "-" is passed instead of a URL(s), the URLs will be read from the standard input.

### General options:

```

-h, --help          Print this usage message.
-q, --quiet          Do not show messages about the progress in the retrieval and analysis
                     of the archives.
--version            Show the version number and exit.
-                    Read URLs from the standard input. This will ignore all the URLs
                     passed via the command line.

```

### Report options:

```

--report-file        Filename for the report generated after the analysis (default is
                     standard output)
                     WARNING: The report file will be overwritten if already exists.
--no-report           Do not generate report after the retrieval and parsing of the
                     archives.

```

## Private archives options:

```
--web-user      If the archives of the mailing list are private, use this username to
                  login in order to retrieve the files.
--web-password  If the archives of the mailing list are private, use this password to
                  login in order to retrieve the files.
```

## MySQL database options:

```
--db-user          Username to connect to the database (default is operator)
--db-password      Password to connect to the database (default is operator)
--db-name          Name of the database that contains data previously analyzed
                  (default is mlstats)
--db-hostname      Name of the host with a database server running (default is
                  localhost)
--db-admin-user    Username to create the mlstats database (default is root)
--db-admin-password Password to create the mlstats database (default is empty)
```

**Basic Usage**

Just like previously with CVSSaNaLY, a database is provided by default, but in this case, if we want the tool to create the database, we need to add the options *--db-admin-user* and *--db-admin-password*.

```
mlstats --db-user root --db-password root --db-name mlstats_evince --db-admin-user
root --db-admin-password root http://mail.gnome.org/archives/evince-list/
```

In this case, MLStats provides information for each month that is analysed.

```
Retrieving http://mail.gnome.org/archives/evince-list/2009-June.txt.gz...
```

```
Retrieving http://mail.gnome.org/archives/evince-list/2009-May.txt.gz...
```

```
Retrieving http://mail.gnome.org/archives/evince-list/2009-April.txt.gz...
```

```
.....
.....
```

```
Analyzing      /home/dizquierdo/.mlstats/mbox/mail.gnome.org/archives/evince-list/2005-
January.txt
```

```
***WARNING: 189 messages (out of 252) parsed but not stored***
```

```
Analyzing      /home/dizquierdo/.mlstats/mbox/mail.gnome.org/archives/evince-list/2005-
February.txt
```

```
***WARNING: 106 messages (out of 157) parsed but not stored***
```

```
Analyzing      /home/dizquierdo/.mlstats/mbox/mail.gnome.org/archives/evince-list/2005-
March.txt
```

\*\*\*WARNING: 90 messages (out of 135) parsed but not stored\*\*\*

In the end, the tool will provide a small summary:

```
3392 messages analyzed
1635 messages stored in database mlstats_evince
0 messages ignored by the parser
WARNING: Some messages were parsed but not stored
MLStats report
-----
```

Total messages by domain name (only top 10 per list):

Mailing list	Domain name	#
-----	-----	----
evince-list	gmail.com	323
evince-list	yandex.ru	237
evince-list	redhat.com	155
evince-list	gnome.org	151
evince-list	xs4all.nl	67
evince-list	suwalski.net	23
evince-list	gmx.net	20
evince-list	hotmail.com	18
evince-list	web.de	17
evince-list	newcon.de	16

Total people posting by domain name (only top 10 per list):

Mailing list	Domain name	#
-----	-----	----
evince-list	gmail.com	148
evince-list	gnome.org	19
evince-list	yahoo.com	10
evince-list	gmx.de	9
evince-list	redhat.com	8
evince-list	web.de	6
evince-list	hotmail.com	4
evince-list	comcast.net	4
evince-list	64studio.com	3
evince-list	gmx.net	3

Total messages by top level domain(only top 10 per list):

Mailing list	TLD	#
-----	-----	----
evince-list com	657	
evince-list ru	243	
evince-list org	223	
evince-list net	90	
evince-list nl	75	
evince-list de	71	
evince-list edu	24	
evince-list it	23	
evince-list fr	21	
evince-list ca	20	

Total people posting by top level domain(only top 10 per list):

Mailing list	TLD	#
-----	-----	----
evince-list com	251	
evince-list org	56	
evince-list net	35	
evince-list de	33	
evince-list edu	17	
evince-list fr	13	
evince-list uk	12	
evince-list it	9	
evince-list au	8	
evince-list nl	7	

Total messages by year:

Mailing list	Year	#
-----	-----	----
evince-list 2005	536	
evince-list 2006	213	
evince-list 2007	300	
evince-list 2008	314	
evince-list 2009	272	

## Total people posting by year:

Mailing list	Year	#
-----	-----	----
evince-list	2005	139
evince-list	2006	77
evince-list	2007	128
evince-list	2008	143
evince-list	2009	87

## Total messages by email address (only top 10 per list):

Mailing list	Email	#
-----	-----	----
evince-list	nshmyrev@yandex.ru	237
evince-list	carlosgc@gnome.org	81
evince-list	jrb@redhat.com	65
evince-list	uws+gnome@xs4all.nl	58
evince-list	wbolster@gnome.org	48
evince-list	bclark@redhat.com	41
evince-list	mpgritti@gmail.com	28
evince-list	pat@suwalski.net	23
evince-list	mpg@redhat.com	19
evince-list	jani.monoses@gmail.com	17

## Total people posting in each list:

Mailing list	#
-----	----
evince-list	485

## Total messages in each list:

Mailing list	#
-----	----
evince-list	1635

MLStats, Copyright (C) 2007-2009 Libresoft Research Group

MLStats is Open Source Software/Free Software, licensed under the GNU GPL.

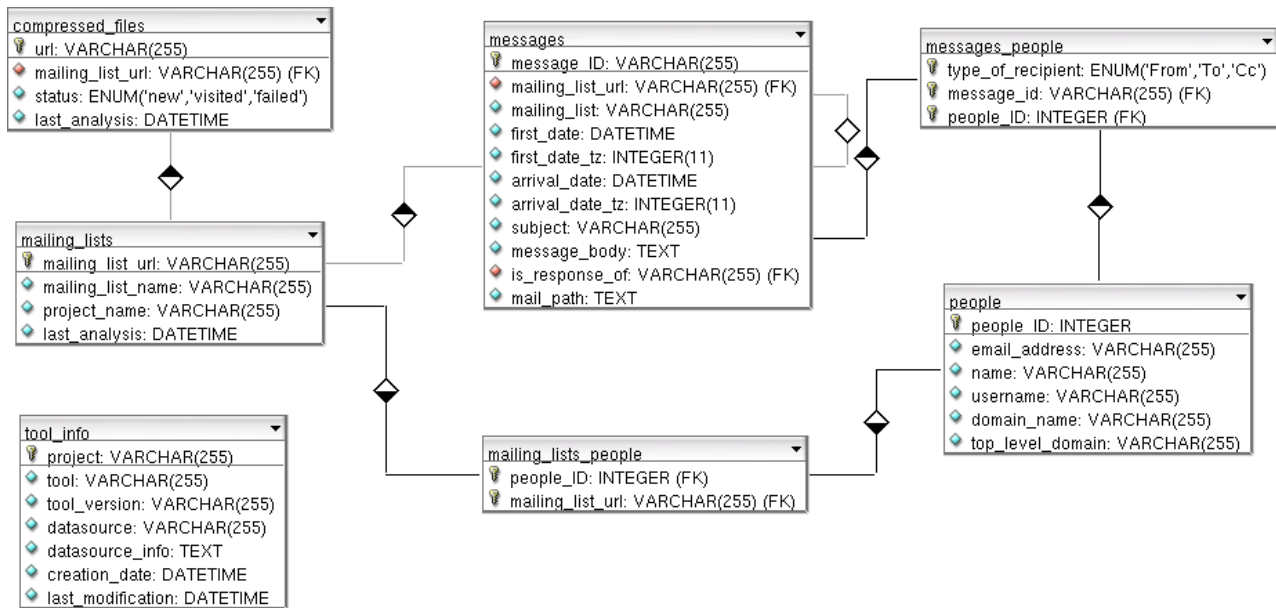
MLStats comes with ABSOLUTELY NO WARRANTY, and you are welcome to

redistribute it under certain conditions as specified by the GNU GPL license;  
see the documentation for details.

Please credit this data as "generated using Libresoft's 'MLStats'."

## Database Schema

The following is the database schema for MLStats:



Lets have a closer look at the tables.

```
mysql> show tables;
```

```

+-----+
| Tables_in_mlstats_evince |
+-----+
| compressed_files         |
| mailing_lists             |
| mailing_lists_people      |
| messages                  |
| messages_people           |
| people                    |
+-----+
  
```



Most interesting tables are *mailing\_lists*, *messages* and *messages\_people*.

Table mailing\_lists:

```
mysql> desc mailing_lists;
```

Field	Type	Null	Key	Default	Extra
mailing_list_url	varchar(255)	NO	PRI	NULL	
mailing_list_name	varchar(255)	YES		NULL	
project_name	varchar(255)	YES		NULL	
last_analysis	datetime	YES		NULL	

Table messages:

```
mysql> desc messages;
```

Field	Type	Null	Key	Default	Extra
message_ID	varchar(255)	NO	PRI	NULL	
mailing_list_url	varchar(255)	NO	MUL	NULL	
mailing_list	varchar(255)	YES		NULL	
first_date	datetime	YES		NULL	
first_date_tz	int(11)	YES		NULL	
arrival_date	datetime	YES		NULL	
arrival_date_tz	int(11)	YES		NULL	
subject	varchar(255)	YES		NULL	
message_body	text	YES		NULL	
is_response_of	varchar(255)	YES	MUL	NULL	
mail_path	text	YES		NULL	

Table messages\_people:

```
mysql> desc messages_people;
```

Field	Type	Null	Key	Default	Extra
type_of_recipient	enum('From','To','Cc')	NO	PRI	From	
message_id	varchar(255)	NO	PRI	NULL	
people_ID	int(10) unsigned	NO	PRI	NULL	

## Basic Queries

### Total Number of Sent Emails

```
mysql> select count(*) from messages;
```

count(*)
1635

### Total Number of Participants

```
mysql> select count(distinct(people_ID)) from messages_people;
```

count(distinct(people_ID))
504

### Total Number of Emails grouped by year and month

```
mysql> select year(first_date), month(first_date), count(*) from messages group by year(first_date), month(first_date) order by year(first_date), month(first_date);
```

year(first_date)	month(first_date)	count(*)
2005	1	64
2005	2	50
2005	3	45

	2005		4		88	
	2005		5		70	
	2005		6		36	
	2005		7		41	
	2005		8		38	
	2005		9		39	
	2005		10		33	
	2005		11		12	
	2005		12		20	

.....  
.....  
.....

	2009		3		71	
	2009		4		30	
	2009		5		40	
	2009		6		45	

+-----+-----+-----+

53 rows in set (0.01 sec)

## ***BICHO***

This tool retrieves data from the bug tracking systems and stores it in a MySQL database. Currently, this is able to work with SourceForge (parsing HTML) and GNOME, KDE and Apache Bugzillas.

### ***Options***

Usage: bicho [options] [URL]

It extracts data from bug tracking systems from a project given

Options:

-h, --help	Print this usage message.
-t, --type	Type of bug tracking system (sf bg) SourceForge or Bugzilla
-p, --path	Path where downloaded URLs will be stored (/tmp/bicho/)

Database input specific options:

--db-driver_in	Input database driver [sqlite mysql postgres] (None)
--db-user_in	Database user name (None)
--db-password_in	Database user password (None)

```
--db-database_in Database name (None)
--db-hostname_in Name of the host where database server is running (None)
--db-port_in      Port where the database server is running (None)
```

Database output specific options:

```
--db-driver_out   Output database driver [sqlite|mysql|postgres] (mysql)
--db-user_out     Database user name (None)
--db-password_out Database user password (None)
--db-database_out Database name (None)
--db-hostname_out Name of the host where database server is running (localhost)
--db-port_out     Port where the database is (3306)
```

Values found in config file 'bicho.conf' are used as default values

If config file is not found all parameters are required except:

```
if url is given:
    Database input parameters are not required
else:
    Database input parameteres are required.
```

This tool also works with a configuration file with the same options as seen above. The most important are `--type`, which indicates the kind of repository to be analysed (SourceForge or Bugzilla) and specifications for the database.

### ***Basic Usage***

For this tutorial, and as a piece of advice, we recommend to download a database previously generated by Bicho from the FLOSSMetrics project. This will avoid the inevitable wait for the time to retrieve all the bugs. Specifically there is a sleep of five seconds between each HTTP access. Generally speaking, the BTSs are overloaded most of the time due to the heavy load of activity in the servers. Thus, in order to avoid this undesirable situation, one can download a fm3 database SQL *dump* created by Bicho. A database for each repository can be found at <http://melquiades.flossmetrics.org> which is the initial step to access all the data. Besides, there is a search box which allows to filter by name.

### ***FLOSSMetrics: Creating the Database***

A good example project is Findbugs, which is hosted in SourceForge

(<http://melquiades.flossmetrics.org/project/findbugs>).

Wget

[http://melquiades.flossmetrics.org/data/projects/findbugs/fm3\\_findbugs\\_bts\\_3067\\_20081127T23:37:50.122782.sql.gz](http://melquiades.flossmetrics.org/data/projects/findbugs/fm3_findbugs_bts_3067_20081127T23:37:50.122782.sql.gz)

Once we have downloaded the file, it is necessary to extract it and charge it in the database server. To uncompress the file, we need to execute this shell command:

```
gunzip *.sql.gz (or use unq and forget forever)
```

It is not mandatory to create the database. In this case, Bicho provides this information. So, we need to run the following commands:

```
mysqladmin -u root -proot create bicho_findbugs  
mysql -u root -proot bicho_findbugs < *.sql
```

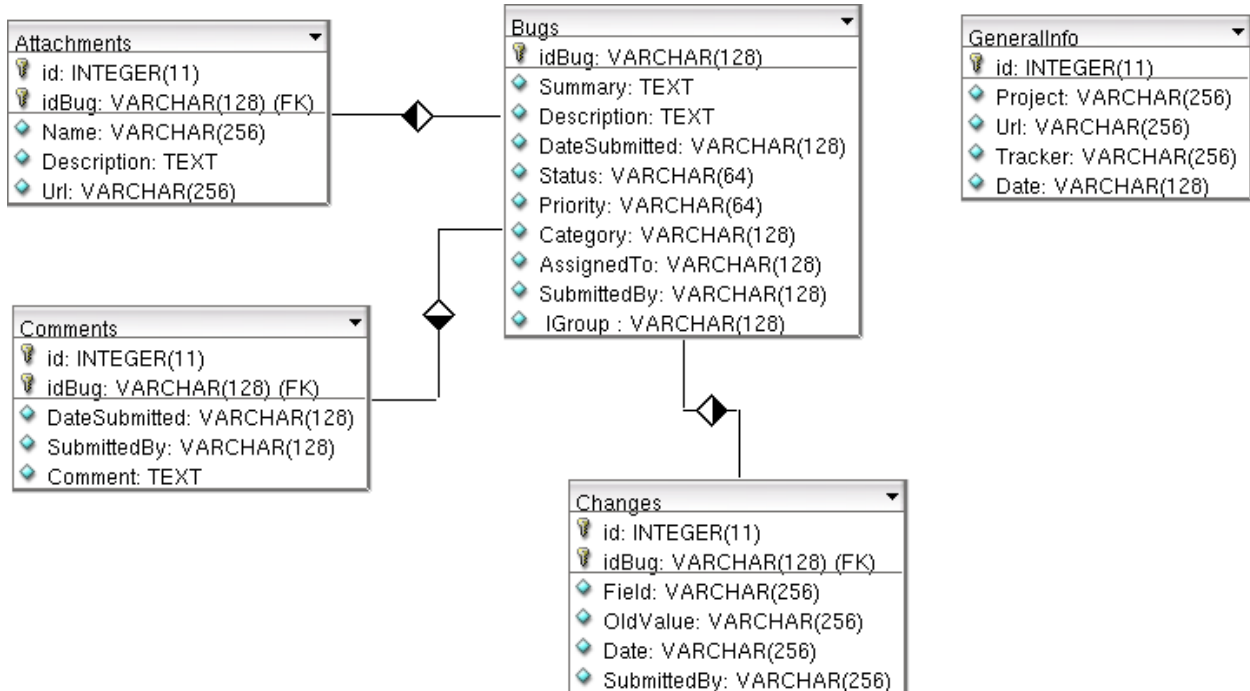
Please, notice that this command is using root as user and also as password. If your user or password is different, use yours.

Once it is done, lets check that everything works.

```
$ mysql -u root -proot  
mysql> show databases;
```

## Database Schema

The following is the database schema for Bicho:



Lets have a closer look to the tables.

```

mysql> show tables;
+-----+
| Tables_in_bicho_findbugs |
+-----+
| Attachments              |
| Bugs                     |
| Changes                  |
| Comments                 |
| GeneralInfo              |
+-----+
  
```

The table *Attachments* provides the URL to the different files found and they are related to their comments. The table *Comments* is formed of all the comments written down by users related to a bug. Finally, table *Changes* provides information about each change ever related to a given bug.

The main table of the schema is the table named as *Bugs*:

```
mysql> desc Bugs;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
idBug	varchar(128)	YES		NULL	
Summary	text	YES		NULL	
Description	text	YES		NULL	
DateSubmitted	varchar(128)	YES		NULL	
Status	varchar(64)	YES		NULL	
Priority	varchar(64)	YES		NULL	
Category	varchar(128)	YES		NULL	
IGroup	varchar(128)	YES		NULL	
AssignedTo	varchar(128)	YES		NULL	
SubmittedBy	varchar(128)	YES		NULL	

## Basic Queries

### Total Number of Bugs.

```
mysql> select count(*) from Bugs;
```

count(*)
658

### Total Number of People submitting Bugs.

```
select count(distinct(SubmittedBy)) from Bugs;
```

count(distinct(SubmittedBy))
303

**Total Number of Bugs grouped by year and month**

```
mysql> select year(DateSubmitted), month(DateSubmitted), count(*) from Bugs group by
year(DateSubmitted), month(DateSubmitted) order by year(DateSubmitted),
month(DateSubmitted);
```

year(DateSubmitted)	month(DateSubmitted)	count(*)
2004	5	4
2004	6	10
2004	7	7
2004	8	11
2004	9	1
2004	10	14
2004	11	9
.....		
.....		
.....		
2008	2	14
2008	3	30
2008	4	20
2008	5	16
2008	6	17
2008	7	20
2008	8	8
2008	9	1

***SLOCCOUNT***

This tool studies the source code lines. It provides information related to the number of SLOC (Source Lines of Code), the kind of programming language used and a COCOMO 85 basic study is launched to determine cost and effort.

The results of this tool is provided in the shell.

Here is an example analysing the Evince repository:

```
svn co http://svn.gnome.org/svn/evince/trunk/
cd trunk
sloccount ./
```



Have a non-directory at the top, so creating directory top\_dir

Adding /tmp/trunk/./AUTHORS to top\_dir

Adding /tmp/trunk/./COPYING to top\_dir

.....  
 .....  
 .....

Found a working MD5 command.

Computing results.

SLOC	Directory	SLOC-by-Language (Sorted)
21621	backend	ansic=19577,cpp=2044
13293	shell	ansic=13293
9466	libview	ansic=9466
7524	cut-n-paste	ansic=7468,sh=56
5465	libdocument	ansic=5465
689	libmisc	ansic=689
595	previewer	ansic=595
438	properties	ansic=438
174	thumbnailer	ansic=174
110	test	python=110
59	top_dir	ansic=46,sh=13
0	data	(none)
0	help	(none)
0	po	(none)

Totals grouped by language (dominant language first):

ansic: 57211 (96.26%)  
 cpp: 2044 (3.44%)  
 python: 110 (0.19%)  
 sh: 69 (0.12%)

Total Physical Source Lines of Code (SLOC) = 59,434  
 Development Effort Estimate, Person-Years (Person-Months) = 14.58 (174.96)  
 (Basic COCOMO model, Person-Months = 2.4 \* (KSLOC\*\*1.05))  
 Schedule Estimate, Years (Months) = 1.48 (17.79)

```
(Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule) = 9.83
Total Estimated Cost to Develop = $ 1,969,601
(average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
SLOCCount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOCCount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOCCount'."
```

## ***GUILTY***

Guilty is a file-oriented tool that retrieves information about line changes. Probably too new to explain it properly ;-)

```
teo@limon:~/Desktop/20091125_master/guilty $ guilty --help
Usage: guilty [ options ... ] URI [ FILES ]
```

Analyze repository modifications

### Options:

```
--version          show program's version number and exit
-h, --help         show this help message and exit
-g, --debug        Run in debug mode
-c FILE, --config-file=FILE
                   Use a custom configuration file
-r REV, --revision=REV
                   Revision to analyze (HEAD)
-f, --fast         Run faster but moves and copies are not detected
-o OUTPUT, --output=OUTPUT
                   Output type [text|db|xml|csv] (text)
```

### Database Options:

```
--db-driver=driver Output database driver [mysql|sqlite] (mysql)
-u user, --db-user=user
                   Database user name (operator)
-p password, --db-password=password
                   Database user password
-d database, --db-database=database
                   Database name (guilty)
```

```
-H host, --db-hostname=host
```

```
      Name of the host where database server is running
      (localhost)
```

Create database, run Guilty for CVSanaly and a small example query.

```
guilty -o db --db-user=root --db-password=root --db-database=guilty ../cvsanaly/
```

```
select a.name, count(b.id) from blame b, authors a where a.id=b.author_id group by
author_id order by 2 desc;
```

## Database Schema

Have a look at it for yourself.

```
mysql> show tables;
```

```
+-----+
| Tables_in_guilty |
+-----+
| authors          |
| blame            |
| files            |
| revisions        |
+-----+
4 rows in set (0.62 sec)
```


```
mysql> desc blame;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    |       |
| file_id    | int(11)       | YES  | MUL | NULL    |       |
| revision_id | int(11)       | YES  | MUL | NULL    |       |
| author_id  | int(11)       | YES  | MUL | NULL    |       |
| line       | int(11)       | YES  |     | NULL    |       |
| date       | datetime      | YES  |     | NULL    |       |
| orig_path  | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

## ***REFERENCES***

Everything you need to know about the tools (or at least everything that is available about the tools ;-), can be found here:

<http://melquiades.flossmetrics.org/wiki/doku.php>

	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 8 of 11</p> <hr/> <p>Version: 1.0 Date: Dec 9, 09</p> <hr/> <p>Status : Proposal Confid : Public</p>
---	--	--

## 2.3“QUALOSS” SLIDES

# QualOSS

## QUALity of Open Source Software

<http://qualoss.org>

Master on Free Software  
Madrid, 10<sup>th</sup> January 2009

Presented by **Daniel Izquierdo-Cortazar**  
[dizquierdo@gsyc.escet.urjc.es](mailto:dizquierdo@gsyc.escet.urjc.es)  
Universidad Rey Juan Carlos

**Jean-Christophe Deprez**  
QualOSSProject Coordinator  
[jean-christophe.deprez@cetic.be](mailto:jean-christophe.deprez@cetic.be)  
CETIC



# Outline

- Introduction
- Motivation
- QualOSS Methodology
- Current Status



# Introduction

- Funded by the Sixth Framework Program
- Duration: 2,5 years
- September 2006 – February 2009





# Introduction

- Consortium:
  - CETIC, Belgium
  - Faculté Universitaire Notre Dame de la Paix, Belgium
  - Universidad Rey Juan Carlos (GSyC/Libresoft), Spain
  - Fraunhofer IESE, Germany
  - ZEA Partners, Belgium.
  - UNU-MERIT, The Nederland
  - AdaCore, France,
  - PePiTe, Belgium

# Motivation

- The strategic objective of the QualOSS project is to enhance the competitive position of the European software industry by providing a methodology and tools for improving productivity and the quality of software products

# Motivation

- FLOSS endeavor definition
- This is a set of:
  - Community members
  - FLOSS components
  - Software process support tools
  - Software libraries
  - ...
- A FLOSS endeavor in FLOSS world  $\approx$  an enterprise in proprietary world

# Motivation

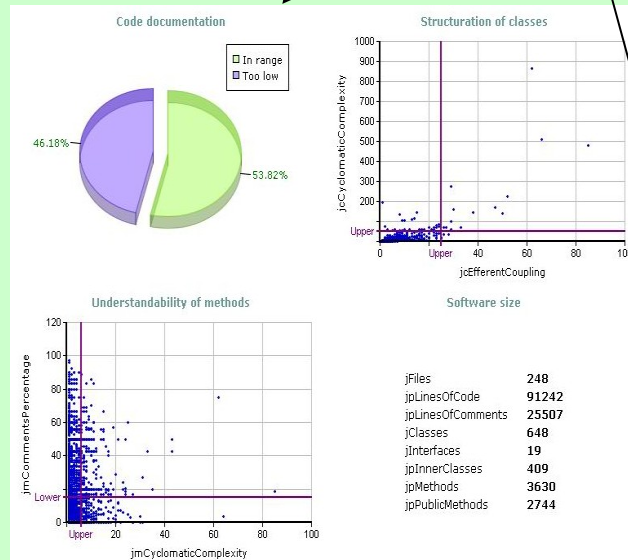
**FLOSS Integrator:**  
*Now I know what  
FLOSS endeavours  
best fits my quality  
needs*



**FLOSS community leader:**  
*Now I know how to  
monitor my FLOSS  
endeavor*

QUALOSS  
Platform

QUALOSS  
repository



## FLOSS datasets

FLOSS own repositories  
(eg in a forge)

Filtered FLOSS data

- FLOSSMETRICS
- FlossMole
- SQO-OSS
- Ohloh

Other related repositories

- Com. Vul. DB
- Amazon
- Interviews

# Motivation

- QualOSS methodology application:
  - Select the best FLOSS component/sub-component which best fits with my requirements
  - Decide when to migrate from one version of a FLOSS component to a newer one

# QualOSS Methodology

- Robustness
  - Capability to keep functioning when mishaps occur
    - External: the appearance of a new competing FLOSS endeavor
    - Internal: Discussion between two community members
- Evolveability
  - Capability to remain viable in the long future

# QualOSS Methodology

It is a generic F/OSS assessment process composed of 5 tasks:

- Initiating an Assessment
  - Identify the business reason for an assessment
- Setting up an Assessment
  - Identify data, tools, people, workflows involved in an assessment

# QualOSS Methodology

- Collect and Analyse Data
  - Follow and adjust the outcome of setting-up task to collect and analyze the data identify
- Interpret the results
  - Follow and adjust the outcome of setting-up task to interpret the results
- Supervise an Assessment
  - Collect issues raised during the 4 other tasks



# Current Status

- Fifty FLOSS endeavors were selected
  - Previously, some of them were manually analysed.
  - Java, C, C++, Python, Ada
- Working on the QualOSS platform
  - First alpha version is almost ready.
  - Some data have been retrieved from projects.


# Current Status

- Retrieving data from FLOSSMetrics or FLOSSMole
  - Projects selected were included in FLOSSMetrics analysis
- Creating indicators
  - Indicators can have four colors, black, red, yellow and green.
  - They will help to provide a general overview of a set of metrics

# Thanks for your attention



Any questions?  
Comments?  
Concerns?

	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 9 of 11</p> <hr/> <p>Version: 1.0 Date: Dec 9, 09</p> <hr/> <p>Status : Proposal Confid : Public</p>
---	--	--

## 2.4“INTRODUCTION TO OPENBRR” SLIDES

# OpenBRR: Business Readiness Rating

Master on Free Software

Gregorio Robles

greg@gsync.escet.urjc.es

GSyC/Libresoft

9-10 November 2007

LibreSoft  
we study libre software

**GSyC**



Universidad  
Rey Juan Carlos

GSyC

LibreSoft

(cc) 2007 Gregorio Robles

Some rights reserved. This work licensed under Creative Commons Attribution-ShareAlike License. To view a copy of full license, see <http://creativecommons.org/licenses/by-sa/2.0/> or write to Creative Commons, 559 Nathan

Abbott Way, Stanford, California 94305, USA.

# Introduction

- Evaluating software is a critical task for IT managers responsible for the selection of software components
- There is currently no easy, effective and trustworthy element for assessing free software
- The Business Readiness Rating OpenBRR is being proposed as an open and standard framework to quicklyl make informed and educated decisions on free software
- Lead by the CarnegieMellon West University, Spike Source, Intel and O'Reilly's Code Zoo

# The challenge

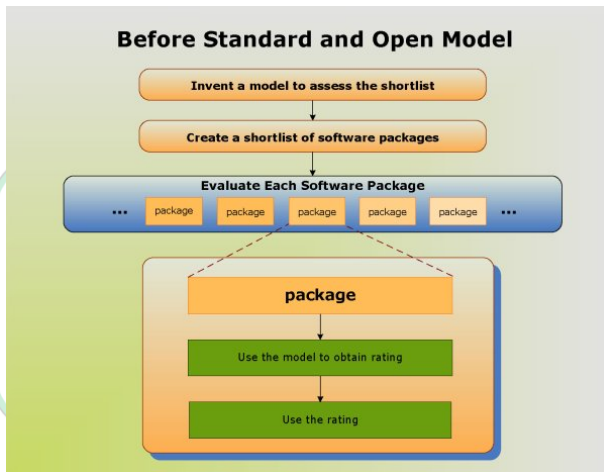
Users and potential adopters of free software face following challenges:

- Selection
- Support
- Longevity
- Volatility
- Low quality





# Before Standard and Open Model



# Looking for an Open and Standard Model

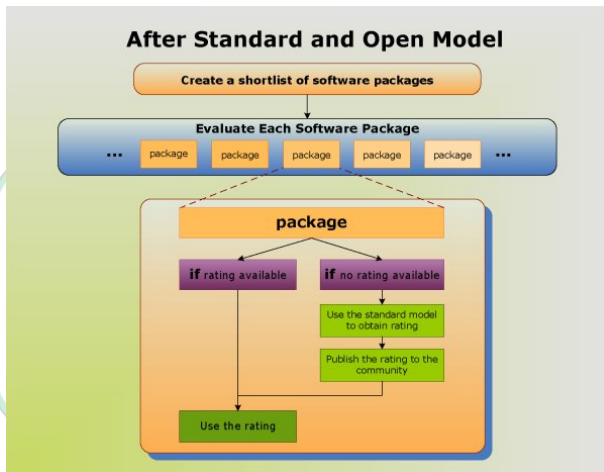
## Characteristics of the model (CSAC)

- Complete (every prominent characteristic)
- Simple
- Adaptable
- Consistent

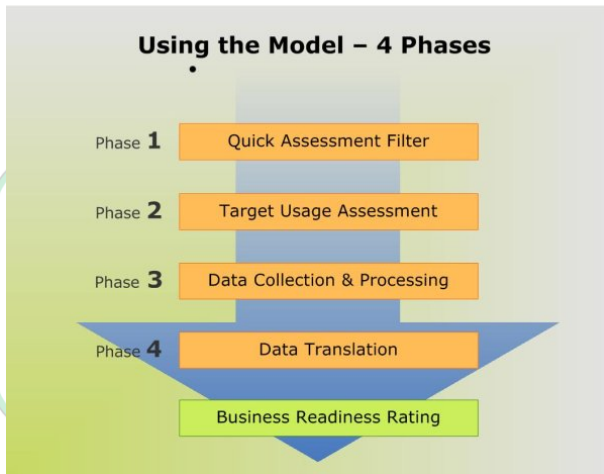
Allows to share assessment results, promotes trust.



# After a Standard and Open Model



# The Four Phases of Software Assessment



# Initial Filtering

## Quick Assessment Phase:

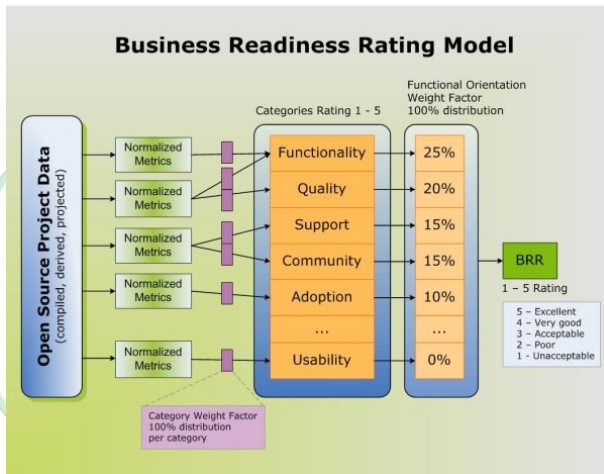
- What is the licensing/legal situation of the software?
- Does it comply with standards?
- Are there referenceable adopters or users for it?
- Is a supporting or stable organization associated with the development efforts?
- What is its implementation language?
- Does it support internationalization and localization in your desired language?
- Are there third-party reviews of the software?
- Have books been published about the software?
- Is it being followed by industry analysts, such as Gartner or IDC?


# Metrics and Categories

- Functionality
- Usability
- Quality
- Security
- Performance
- Scalability
- Architecture
- Support
- Documentation
- Adoption
- Community
- Professionalism



# OpenBRR in detail: Weightings



	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 10 of 11</p> <hr/> <p>Version: 1.0</p> <p>Date: Dec 9, 09</p> <hr/> <p>Status : Proposal</p> <p>Confid : Public</p>
---	--	---

## 2.5“COMPARISON Q<sub>SoS</sub> AND OPENBRR” SLIDES



# OpenBRR and QSoS

## Master on Free Software

Daniel Izquierdo Cortazar

dizquierdo@gsyc.es  
GSyC/Libresoft

10th January, 2009

**GSyC**



Universidad  
Rey Juan Carlos

GSyC

LibreSoft

(cc) 2009 Daniel Izquierdo Cortazar

Some rights reserved. This work licensed under Creative Commons Attribution-ShareAlike License. To view a copy of full license, see <http://creativecommons.org/licenses/by-sa/2.0/> or write to Creative Commons, 559 Nathan

Abbott Way, Stanford, California 94305, USA.

# Index

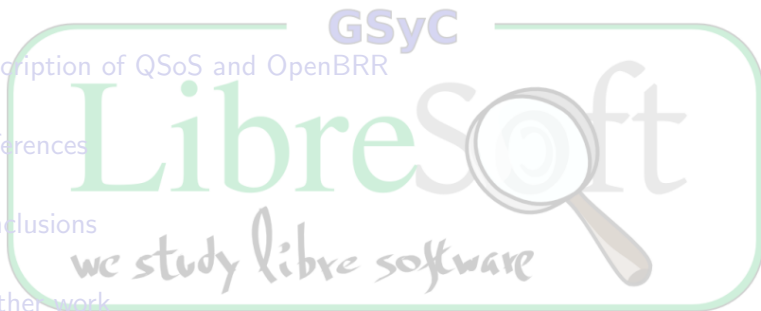
## 1 Introduction

## 2 Description of QSoS and OpenBRR

## 3 Differences

## 4 Conclusions

## 5 Further work



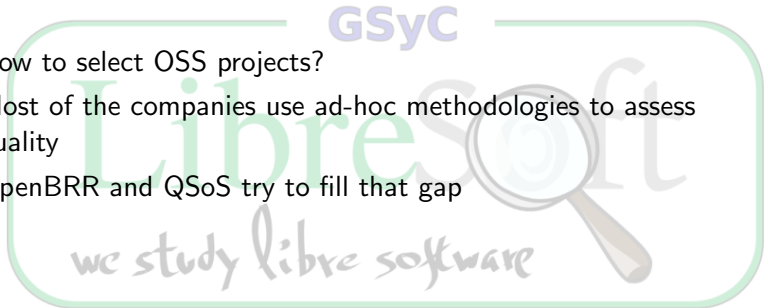
Based on:

- **Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSoS**
- *Jean-Christophe Deprez and Simon Alexandre*
- *CETIC, Charleroi, Belgium*

we study libre software

# Introduction

- How to select OSS projects?
- Most of the companies use ad-hoc methodologies to assess quality
- OpenBRR and QSoS try to fill that gap



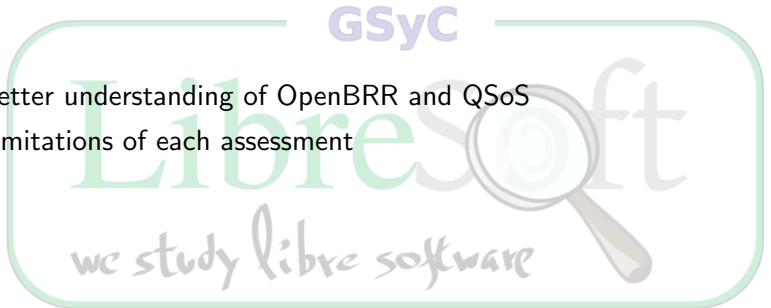
# Introduction

- Light weight methodologies.
- QSoS (Atos Origin)
- OpenBRR (Carnegie Mellon West and Intel)



# Motivation

- Better understanding of OpenBRR and QSoS
- Limitations of each assessment



# Motivation

- Comparison of:
- Overall approach
- Scoring Procedures
- Evaluation Criteria





# Index

1 Introduction

2 Description of QSoS and OpenBRR

3 Differences

4 Conclusions

5 Further work



# QSoS and OpenBRR

- Start from a list of projects given by the FLOSS integrator
- QSoS provides a list of criteria and "quality attributes"
- A score is provided by the criteria given by QSoS

*we study libre software*

# QSoS: Main Criteria

- Intrinsic Durability
- Industrialized Solution
- Technical Adaptability
- Strategy



# OpenBRR: Main Criteria

- Usability
- Quality
- Security
- Performance
- Scalability
- Architecture
- Support
- Documentation
- Adoption
- Community
- Professionalism



# Index

1 Introduction

2 Description of QSoS and OpenBRR

3 Differences

4 Conclusions

5 Further work



# OpenBRR

- Each user may have a different sight of the product
- Main criteria could be less important depending on the assigned role
- "Usability" may not mean the same for a developer than for a user
- "Support" may not mean the same for a company than for a user
- OpenBRR provides that flexibility

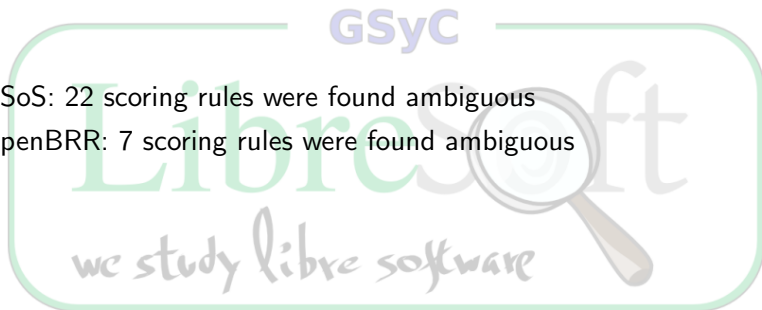
# QSoS

- QSoS provides an absolute score
- In this way, every analysis should give the same score
- There are no roles, and no different points of view

*we study libre software*

# Ambiguity

- QSoS: 22 scoring rules were found ambiguous
- OpenBRR: 7 scoring rules were found ambiguous





## Web sites and activity

- QSoS: It is registered more activity (9th of January)
- OpenBRR: No activity for at least one year (9th of January)

we study libre software

# Index

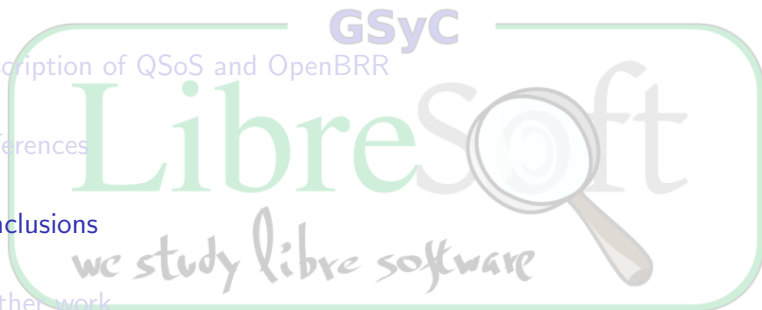
1 Introduction

2 Description of QSoS and OpenBRR

3 Differences

4 **Conclusions**

5 Further work



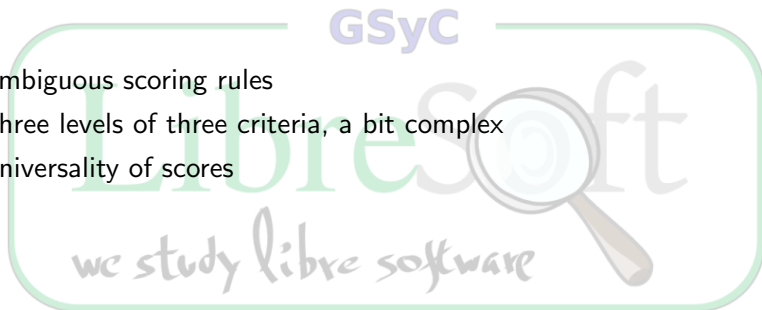
## QSoS: advantages

- Clear list of criteria
- Versioned QSoS methodology
- Extensive list of criteria



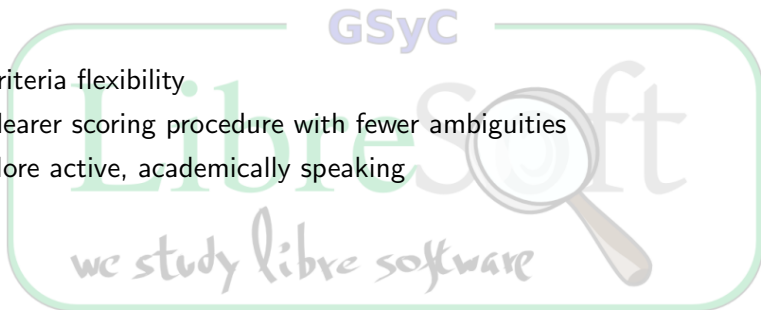
## QSoS: disadvantages

- Ambiguous scoring rules
- Three levels of three criteria, a bit complex
- Universality of scores



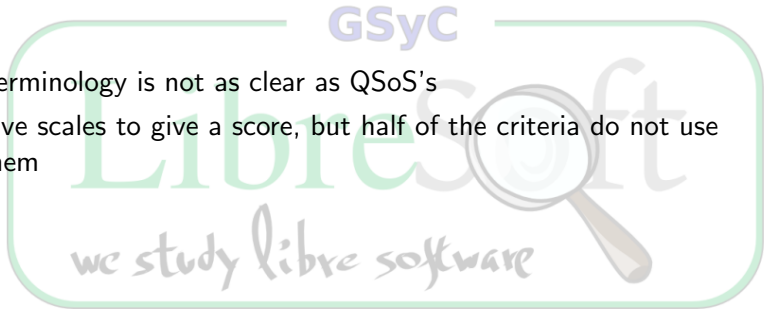
# OpenBR: advantages

- Criteria flexibility
- Clearer scoring procedure with fewer ambiguities
- More active, academically speaking



# OpenBRR: disadvantages

- Terminology is not as clear as QSoS's
- Five scales to give a score, but half of the criteria do not use them



# Index

1 Introduction

2 Description of QSoS and OpenBRR

3 Differences

4 Conclusions

5 Further work




# QualOSS

- Quality in Open Source Software





	<p>Extra Materials</p> <p>Deliverable ID: D6.4</p>	<p>Page : 11 of 11</p> <hr/> <p>Version: 1.0 Date: Dec 9, 09</p> <hr/> <p>Status : Proposal Confid : Public</p>
---	--	---

## 2.6“QUALITÉ LOGICIELLE: INTRODUCTION” SLIDES



# Qualité Logicielle

*Introduction*

Naji HABRA



# Plan 1 : Qualité

- Introduction à la qualité
  - Présentation du contexte
  - Notion et coût
  - Caractéristiques de qualité
  - Définitions
- Modèles Qualité produit
  - Pourquoi un modèle ?
  - Quelques Modèles Produit
- Ex Modèle Produit (Normatif) : ISO 9126
  - Introduction aux normes et à la normalisations
  - Développer la structure de la norme
  - Idée des mesures proposées
  - Liens autres normes ISO 15939 - ISO 14598( + évolutio, Square)
- Modèles Qualité Processus :
  - Qualité et Approche Processus
  - Quelques Modèles Processus
- Ex Modèle Processus : OWPL
- **Ex. Modèles spécifiques**
  - Modèle spécifique pour Open Source
  - Modèles spécifiques pour la Documentation



# QUALOSS Model context EU project

- Motivation

- Open-source development processes have emerged as an effective approach to reduce cycle-time and decrease design, implementation, and quality assurance costs for certain types of software.
- What are the prospects for advancing to much higher levels of quality in open source software?
- More specifically, what attributes must be possessed by quality-related interventions for them to be feasibly adoptable in open source practice?
- What key challenges of open-source software to control long term maintenance and evolution of costs ?



# QUALOSS Model

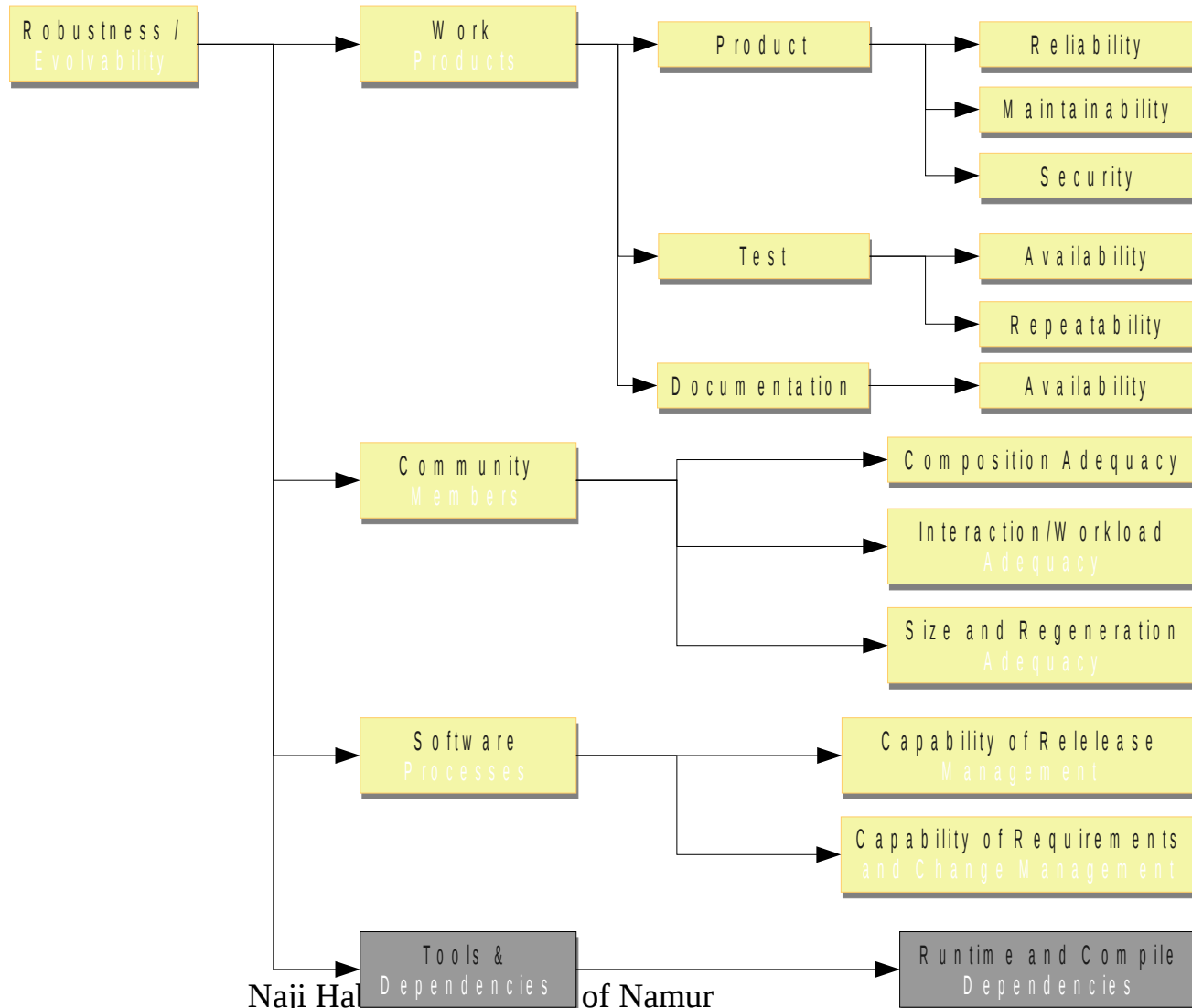
## context : EU project

- Motivation

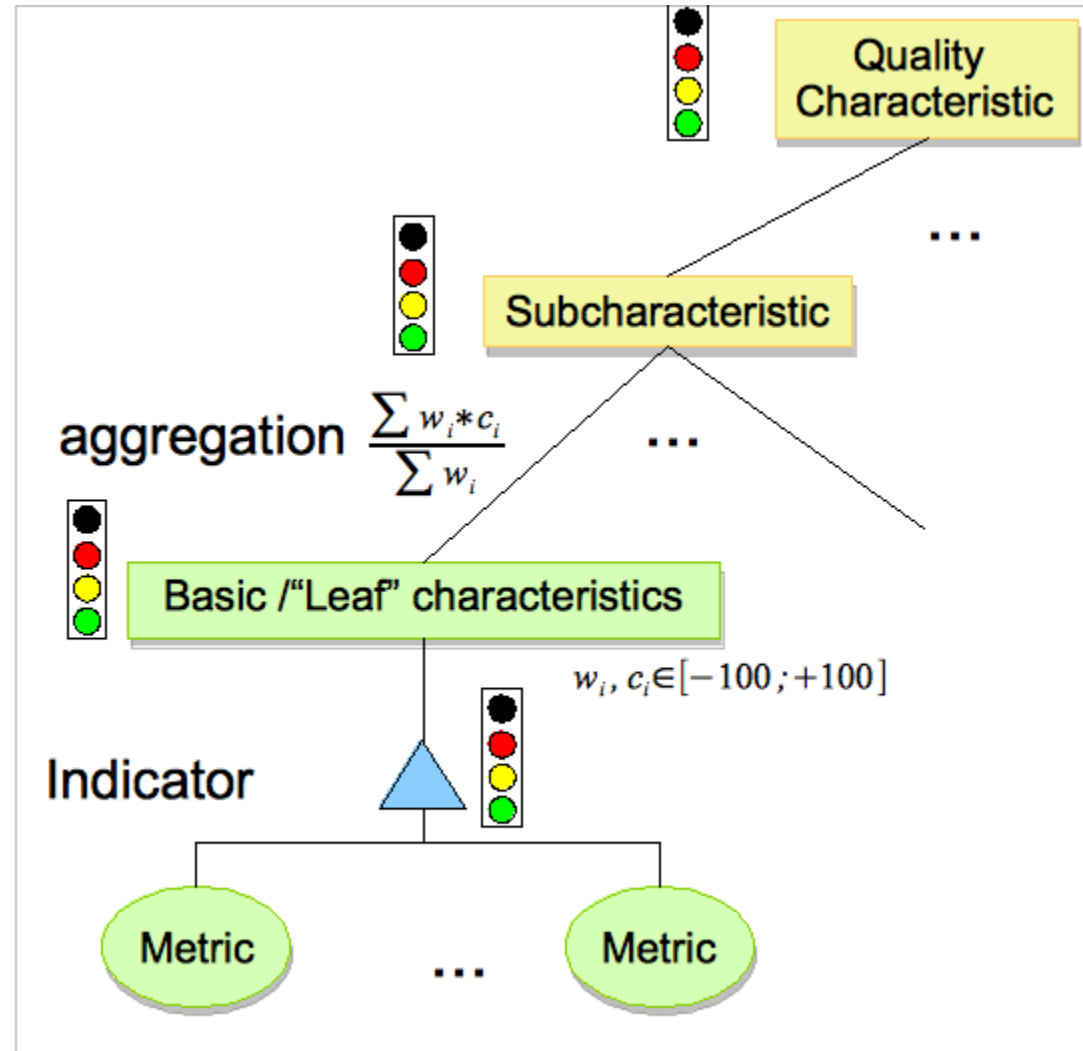
- Open-source development processes have emerged as an effective approach to reduce cycle-time and decrease design, implementation, and quality assurance costs for certain types of software.
- What are the prospects for advancing to much higher levels of quality in open source software?
- More specifically, what attributes must be possessed by quality-related interventions for them to be feasibly adoptable in open source practice?
- What key challenges of open-source software to control long term maintenance and evolution of costs ?



# QUALOSS Model structure



# QUALOSS Model structure





# QUALOSS Model Method

- Preparation Phase
  - Context :
    - Full integration - exploitation - fork ...
  - Scope :
    - project /sub project / collection of projects
  - Usage :
    - integration in product / in service
  - Mode :
    - Comparison - introspection - ...
- Measurement
  - Data collection
  - Manual + automated measurement
  - Parameterization (weight) and indicator calculus
- Interpretation
  - Connection with management / risk





# Documentation Quality model context

- OSS context
  - Generalisable
- New investigation
  - Norms ??
  - Documentation
    - "availability" (syntactical)
    - "<>" correctness" (semantical)



# Documentation Quality model

## Documentation Taxonomy

1. Product  
advertising and  
contact  
document(s)

2. Product  
installation and  
application  
documents

OSS  
documentation  
availability

3. Documents of  
product  
development and  
process

4. Management  
and copyright  
documents



# Documentation Quality model

## Document organisation

Content is arranged so that information can be *easily* located

No:	Question	Metric	Range
1	Is document divided to <b>chapters</b> ?	Chapter	Yes / No
2	Is document organised to <b>sections</b> and/or <b>subsections</b> ?	Section / subsection	Yes / No
3	Is there a <b>table of content</b> in the document?	Table of content	Yes / No
4	Is there a <b>term index</b> in the document?	Term index	Yes / No
5	Is there an <b>acronym glossary</b> in the document?	Acronym glossary	Yes / No
6	Is there a <b>bibliography</b> (cited <b>literature references</b> ) in the document?	Bibliography (literature references)	Yes / No / NA
7	Are all the <b>tables</b> annotated with a unique <b>identification number</b> ?	Table identification numbers	Yes / No / NA

# Documentation Quality model

## Document organisation

$$g = \frac{\sum_{i=1}^N r_i}{N}$$

- **$\{ r1 \dots rN \}$**  : answers to each question about document organisation
- **$N$**  : number of questions about document organisation
- **$g$**  : organisation of a single document

$$dor_j = \frac{\sum_{i=1}^K g_i}{K}$$

- **$K$**  : number of documents belonging to a document type  $j$
- **$dor_j$**  : organisation of documents belonging to document type  $j$



# Documentation Quality model

## Content completeness

- Information necessary for the stakeholder

Content of the document	Description
2. Overall description	Describe the general factors that affect the product and its requirements. Provides a background for those requirements
<b>2.1. Product perspective</b>	Put the product into perspective with other related products or clearly state that it is independent.
<b>2.2. Product function</b>	Provide a summary of the major functions that the software will perform
<b>2.3. User characteristics</b>	Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
<b>2.4. Constraints</b>	Provide a general description of any other items that will limit the developer's options
<b>2.5. Assumptions and Dependencies</b>	List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements
<b>2.6. Apportioning of requirements</b>	Identify requirements that may be delayed until future versions of the system



# Documentation Quality model

## Content completeness

- Information necessary for the stakeholder
- All information units are defined at the high level of detail

No:	Question	Metric	Range
1	What is the degree of information presentation in a document?	Information presentation degree	0 – when question is not analysed; 1 – low presentation degree; 2 – average presentation degree; 3 – high presentation degree



# Documentation Quality model

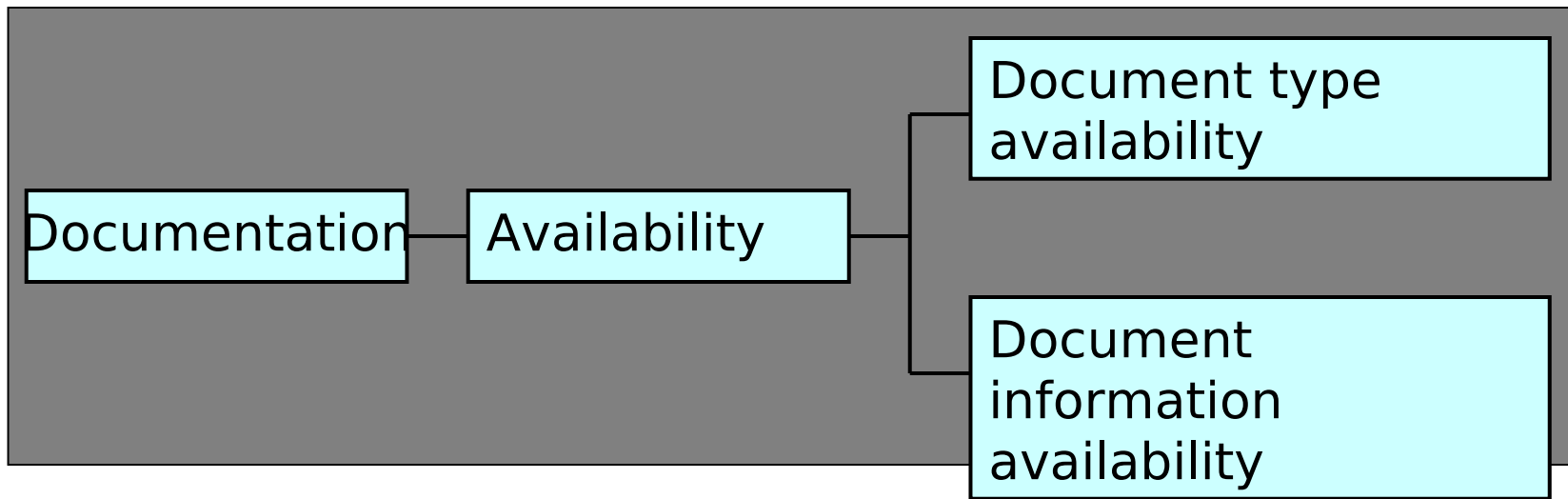
## Document completeness

$$dco_j = \frac{\sum_{i=1}^M c_i d_i}{3M}$$

- **{ c1 ... cM }** : estimates of *content completeness*
- **{ d1 ... dM }** : estimates of *information completeness*
- **M**: number of questions
- **dcoj** : completeness of documents belonging to document type *j*



# Documentation Quality model Indicators







# Documentation Quality model

Indicators *DTA*: Document type availability

- Availability of documents belonging to a certain document type

$$DTA = \frac{DF}{DN}$$

***DF*** : number of documents types for which documents are found

***DN*** : number of considered document types



# Documentation Quality model

Indicators *DIA*: Document information availability

- Organisation and information completeness of the document content

$$DIA = \frac{\sum_{i=1}^{DN} (dor_i + dco_i)}{2DN}$$

*dor<sub>i</sub>* : organisation of documents belonging to document type *i*

*dco<sub>i</sub>* : completeness of documents belonging to document type *i*

*DN* : number of considered document types



# Documentation Quality model

## Indicator interpretation

How can indicators be interpreted?

Documentation  
is not  
available

Documentatio  
n availability  
is limited

Documentatio  
n availability is  
average

Documentatio  
n availability  
is high

How can interpretation of the  
documentation availability (DA) indicators be  
estimated?



	Projects	Document type availability
1	Thunderbird (Mozilla)	100.00%
2	Hadoop	100.00%
3	Xemacs	100.00%
4	Python	92.31%
5	Evolution	92.31%
6	Jmeter	92.31%
7	PLONE	84.62%
8	Zope	84.62%
9	FreeBSD	84.62%
10	NetBSD	84.62%
11	Eclipse	84.62%
12	Jetspeed	84.62%
13	Evince	84.62%
14	PhPMyAdmin	76.92%
15	Writer (OpenOffice)	76.92%
16	VLC	76.92%
17	Jakarta Struts	76.92%
18	Nautilus	76.92%
19	k3b	69.23%
20	xPDF	69.23%
21	Findbugs	69.23%
22	CVSanaly	69.23%
23	Galeon	61.54%
24	Sup	61.54%
25	Omnitux	53.85%
26	yanolc	46.15%

Document type  
availability



	Projects	Document type availability
1	Thunderbird (Mozilla)	100.00%
2	Hadoop	100.00%
3	Xemacs	100.00%
4	Python	92.31%
5	Evolution	92.31%
6	Jmeter	92.31%
7	PLONE	84.62%
8	Zope	84.62%
9	FreeBSD	84.62%
10	NetBSD	84.62%
11	Eclipse	84.62%
12	Jetspeed	84.62%
13	Evince	84.62%
14	PhPMyAdmin	76.92%
15	Writer (OpenOffice)	76.92%
16	VLC	76.92%
17	Jakarta Struts	76.92%
18	Nautilus	76.92%
19	k3b	69.23%
20	xPDF	69.23%
21	Findbugs	69.23%
22	CVSanaly	69.23%
23	Galeon	61.54%
24	Sup	61.54%
25	Omnitux	53.85%
26	yanolc	46.15%

## Document type availability

- Green
  - 88,5 - 100 %
- Yellow
  - 73 - 88,49 %
- Red
  - 57,5 - 72,99 %
- Black
  - Less than 57,5%



	Projects	Document information availability
1	Python	52.45%
2	PLONE	52.08%
3	Thunderbird (Mozilla)	50.46%
4	Zope	49.47%
5	FreeBSD	47.31%
6	PhPMyAdmin	46.56%
7	NetBSD	43.25%
8	Eclipse	42.98%
9	Writer (OpenOffice)	42.20%
10	Hadoop	41.72%
11	Xemacs	41.28%
12	VLC	38.74%
13	Jetspeed	34.83%
14	Jakarta Struts	34.27%
15	Evolution	32.72%
16	Jmeter	32.08%
17	k3b	31.52%
18	Evince	30.85%
19	xPDF	28.96%
20	Findbugs	27.62%
21	Nautilus	25.44%
22	CVSanaly	24.54%
23	yanolc	18.41%
24	Galeon	17.01%
25	Sup	16.15%
26	Omnitux	15.56%

Document information  
availability

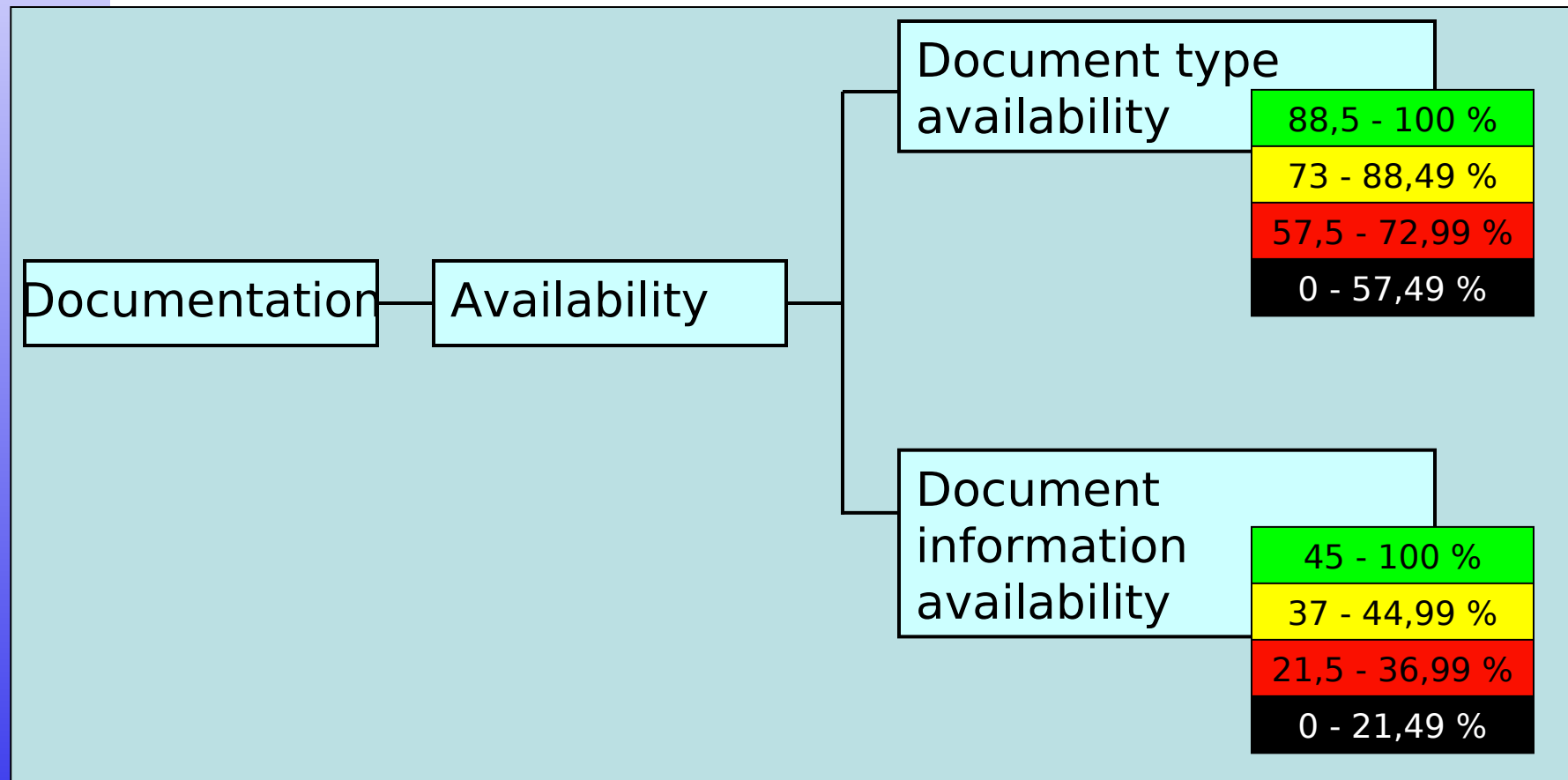
	Projects	Document information availability
1	Python	52.45%
2	PLONE	52.08%
3	Thunderbird (Mozilla)	50.46%
4	Zope	49.47%
5	FreeBSD	47.31%
6	PhPMyAdmin	46.56%
7	NetBSD	43.25%
8	Eclipse	42.98%
9	Writer (OpenOffice)	42.20%
10	Hadoop	41.72%
11	Xemacs	41.28%
12	VLC	38.74%
13	Jetspeed	34.83%
14	Jakarta Struts	34.27%
15	Evolution	32.72%
16	Jmeter	32.08%
17	k3b	31.52%
18	Evince	30.85%
19	xPDF	28.96%
20	Findbugs	27.62%
21	Nautilus	25.44%
22	CVSanaly	24.54%
23	yanolc	18.41%
24	Galeon	17.01%
25	Sup	16.15%
26	OmniTux	15.56%

## Document information availability

- Green
  - 45 - 100 %
- Yellow
  - 37 - 44,99 %
- Red
  - 21,5 - 36,99 %
- Black
  - Less than 21,5%

# Documentation Quality model

## Indicator interpretation







# Documentation Quality model

## Indicator interpretation

	Projects	Document type availability	Document information availability
1	Thunderbird (Mozilla)	100.00%	50.46%
2	Hadoop	100.00%	41.72%
3	Xemacs	100.00%	41.28%
4	Python	92.31%	52.45%
5	Evolution	92.31%	32.72%
6	Jmeter	92.31%	32.08%
7	PLONE	84.62%	52.08%
8	Zope	84.62%	49.47%
9	FreeBSD	84.62%	47.31%
10	NetBSD	84.62%	43.25%
11	Eclipse	84.62%	42.98%
12	Jetspeed	84.62%	34.83%
13	Evince	84.62%	30.85%
14	PhPMYAdmin	76.92%	46.56%
15	Writer (OpenOffice)	76.92%	42.20%
16	VLC	76.92%	38.74%
17	Jakarta Struts	76.92%	34.27%
18	Nautilus	76.92%	25.44%
19	k3b	69.23%	31.52%
20	xPDF	69.23%	28.96%
21	Findbugs	69.23%	27.62%
22	CVSanaly	69.23%	24.54%
23	Galeon	61.54%	17.01%
24	Sup	61.54%	16.15%
25	Omnitux	53.85%	15.56%
26	yanolc	46.15%	18.41%

- Availability of documents of a certain type does not guarantee availability of the information
- A document can contain high availability degree of information, although the overall project documentation availability might be of a low degree