



Free/Libre/Open Source Metrics and Benchmarking

FP6-033982

FLOSSMetrics Final Report

First Edition

March 29, 2010



European Commission



Information Society
Technologies



©FLOSSMetrics consortium
This work is licensed under a Creative
Commons Attribution-Share Alike 3.0
License

Contents

1 The Challenge	2
2 Addressing the Challenge: The Project's Proposition	3
3 Who Benefits From FLOSSMetrics?	5
3.1 Software industry	5
3.2 Libre software projects and communities	6
3.3 Academia and research community	6
4 Highlights of Achievements	7
5 The results	9
5.1 FLOSSMetrics infrastructure	9
5.1.1 Overview	9
5.1.2 Retrieval System	9
5.1.3 Database	11
5.1.4 Melquiades website	13
5.2 High Level Studies	18
5.3 Guide for SMEs	18
6 Pilots and studies	23
7 Recommendations and guidelines on how to use FLOSSMetrics' results	27
8 Availability of Results	31
9 Partners	32

1 The Challenge

Libre (free, open source) software¹ has shown since its origins new possibilities in the area of collaborative production of software products. After almost 30 years of history, it has demonstrated in many cases tremendous success, both in terms of commercial and technical strengths of delivered products, but also as a model of organisation and development.

Libre software is arguably one of the best examples of open, distributed models for production and development that exists today. What is more important, from the point of view of the classical approaches to development methodologies by groups of professionals (and specifically from the point of view of the classical concepts of software engineering), the models used in libre software development are innovative in several ways, to the point that they are only recognized as valid models at all since they have actually produced mature and stable software. Any previous “theoretical” analysis would have probably concluded libre software development was not capable of producing any sustained, useful output.

In this context, industry, SMEs, public administrations and individuals are increasingly relying on libre software as a competitive advantage in the globalising, service-oriented software economy. But they need detailed, reliable and complete information about libre software, specifically about its development process, its productivity and the quality of its results. They need to know how to benchmark individual projects against the general level. And they need to know how to learn from, and adapt, the methods of collaborative, distributed, agile development found in libre software to their own development processes, especially within industry.

Some studies about libre software development try to answer the previous questions, but they are usually focused only on handful of projects, such as the Linux kernel, GNOME, Apache, Mozilla or Eclipse. Although they miss the larger landscape of the whole world of libre software development, they do show interesting details of the development model of the specific projects studied, and of the resulting software products. However, only with reliable data about a large quantity of projects would it be possible to start venturing sound theories about libre software development models, their advantages and problems, the identification of best practices and success strategies. In other words, until data is available about the evolution over time of a large number of projects, with enough detail, and with a quality suitable for being the basis of scientific research, little advance can be expected on our understanding of this new way of developing software.

This lack of understanding is important not only to the research community, but also to any actor interested in participating in libre software development, either as an active contributor, or as a consumer of libre software products. In particular, SMEs and other companies considering to use or to produce libre software need a much better understanding of the inherent qualities of specific products and projects, and ways of interpreting their basic parameters.

¹Through this document, the term “libre software” is used to refer both to “free software”, as defined by the Free Software Foundation and “open source software”, as defined by the Open Source Initiative.

2 Addressing the Challenge: The Project's Proposition

FLOSSMetrics addresses these challenges by analysing in depth, from a quantitative point of view, a large quantity (thousands) of projects, using mainly publicly available data sources.

Libre software projects are known for their open development processes, during which huge quantities of information about the project are made available on the Internet, in many cases in data formats simple to retrieve and parse by using automated tools. For instance, source code management systems (extensively used in libre software projects since several years ago) provide very detailed information about who was doing what and when, and about the historic evolution of the source code itself. Public mailing lists and forums provide a lot of information about the communication channels used in the system, and about the decision making process. Bug tracking systems provide details about the problems found with the software, and the way they are solved. FLOSSMetrics retrieves these (and more) data, storing and later analysing it in an automated way, with frequent and continuous (also automated) monitoring and updates. Once the data for a large quantity of projects was available in comparable conditions, some studies and analyses have also been performed.

These analyses open new lines to better understand the landscape of libre software development, and to obtain factual data about it which can be used to improve libre software development itself and to identify interesting practices that could be used in other contexts. European industry, SMEs, as well as public administrations and individuals will be able to take informed decisions about how to benefit from the competitive advantage of libre software, either as a development process or in the evaluation and choosing of individual software applications.

In short, the main results that FLOSSMetrics has produced are:

- a huge database with factual details about all the studied projects
- some higher level analysis and studies which help to understand how libre software is actually developed
- a sustainable platform for continued, publicly available benchmarking and analysis beyond the lifetime of this project
- a guide with suggestions and guidelines for the adoption of FLOSS within SMEs.

In order to obtain the previous results, the project accomplished distinct tasks:

- Identified and evaluated sources of data and developed a comprehensive database structure
- Integrated already available tools to extract and process such data into a complete platform
- Built and maintained an updated empirical database applying extraction tools to thousands of open source projects

- Developed visualisation methods and analytical studies, especially relating to benchmarking, identification of best practices, measuring and predicting success and failure of projects, productivity measurement, simulation and cost/effort estimation
- Disseminated the results, including data, methods and software
- Provided for exploitation of the results by producing an exploitation plan, validated with the project participants from industry especially from an SME perspective

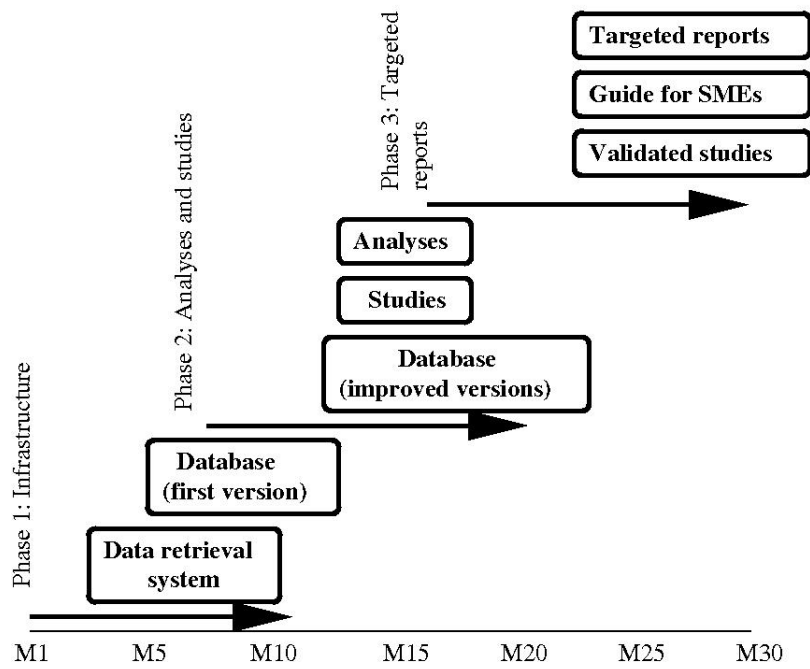


Figure 1: FLOSSMetrics roadmap

3 Who Benefits From FLOSSMetrics?

There are three main groups that can be benefited from FLOSSMetrics:

- Software industry
- FLOSS projects and communities
- Academia and research community

3.1 Software industry

Many large European industrial players expect that FLOSS will become increasingly important, not only in the primary but also in the secondary software sector, covering strategic areas such as telecommunication, automotive, aeronautics and aerospace, large equipment industries, energy, and health. This implies that the need for reliable and comprehensive information on existing FLOSS projects and products will also become more important.

FLOSSMetrics will be useful for industrial users, helping to answer questions or doubts in areas such as:

- Introduction of libre software and spreading it in the organization
- Selection of the right software package in the large search space of possible solutions
- Training of the people involved
- Quality assessment of libre software
- How to guarantee quality to clients and regulatory bodies
- Interoperability with legacy solutions
- Cohabitation of libre software and legacy software
- How to find a community for software that you want to produce, and initiate a libre software community around it
- Policies on company people to get involved in libre software communities
- What kind of company policies are sensible
- Ownership and control issues - how to ensure own requirements are met
- How to keep own differentiating (IPR-related) software separated from libre software

In general, knowledge all these areas will be meaningfully improved by the availability of the data produced by FLOSSMetrics, enabling studies that will help to clarify issues and provide new valuable knowledge.

3.2 Libre software projects and communities

For libre software developers public access to the integrated system and data generated by this project will provide a consistent framework for comparing and benchmarking their own projects, and also for understanding their own state of affairs from new points of view.

In general, libre software developers know a lot of details about the projects in which they work, but not always they have a complete, comprehensive view of how the whole project is performing. Even project leaders, especially in large projects, may lack information to understand the inner dynamics of their communities. In this area, FLOSSMetrics will provide for many projects enough data and analysis of that data to answer many questions, but also a set of tools that projects can use themselves if they prefer to include them in their infrastructure.

As an additional point, libre software projects will also have access to many of the results of the research studies produced (see below), that way helping them to better understand the consequences of their decisions, and their own evolution over time.

3.3 Academia and research community

For the academic community, the FLOSSMetrics results clearly form the basis for further research. The availability of data will enable researchers not used to the retrieval of data directly from libre software repositories to produce interesting research results, focusing on the analysis of the data, and not having to bother about how to efficiently retrieve them.

The availability not only of detailed data, but also of data for a large quantity of projects, is expected also to increase the number of projects considered in research studies, thus contributing to increase their statistical validity, by greatly increasing the size of the sample. This could have a large impact in the field of software engineering in general, since until now it has not been easy to perform large-scale studies, which on the other hand are essential to ensure that conclusions are general enough to be considered.

In particular, the database and studies developed within the project will provide an excellent basis for further research and policy studies, the main form of non-commercial research exploitation for research institutions.

The results are also relevant to education and especially higher education in software engineering, as they can further develop the field of libre software engineering which has the special feature that students can immediately have practical development experience, unlike with proprietary software.

4 Highlights of Achievements

The next list describes the main achievements obtained during the FLOSSMetrics project:

- **FLOSSMetrics tools.** Some tools were developed in order to build the infrastructure that allows to retrieve and analyse data, that later will be stored into the database. These tools are:
 - **Repository Finder:** system developed to automatically find the repositories with information related to the development for a given project.
 - **Retrieval System:** this system allow the researchers to retrieve and analyse data from a large quantity of libre software projects in an automatic way.
 - **Bicho:** bug-tracking retrieval tool that supports SourceForge and Bugzilla issue tracking systems.
- **FLOSSMetrics database.** This database has become one of largest databases about FLOSS, and probably, the largest one storing quantitative data. It currently contains information about the repositories of 3,150 projects (which quantitative data is available about 2,800), including those from forges such as GNOME, KDE, Apache, SourceForge, ObjectWeb or OSOR.
- **Melquiades website.** This website is the main interface to the FLOSSMetrics database. All the analysis and results obtained during project duration are available on this website and are accessible through a powerful API in different formats (i.e database dumps, charts, animations).
- **High level studies.** A total of 14 studies were undertaken using the FLOSSMetrics database. They do not intend to conform a comprehensive analysis of FLOSS development, but a set of examples of how the data collected by FLOSSMetrics can be used for research studies. The studies have in common some context related to software development covering some aspects such as: characterisation of FLOSS development, software evolution, quality, efficiency, maintainability, etc. Furthermore, two studies were performed as good examples of how the FLOSSMetrics data set could be helpful to industry: the first one investigates the productivity of FLOSS developers in socio-economics terms; the second one estimates the cost/estimation study based on substitution costs for FLOSS.
- **SMEs guide.** This guide presents a set of guidelines and suggestions for the adoption of FLOSS within SMEs, that will guide companies from the initial selection and adoption of FLOSS within the IT infrastructure up to the creation of suitable business models based on FLOSS. During this year, a fourth version was published including a software catalogue designed to provide an initial list of packages that are stable and relevant for several different industry sectors.

The guide received significant attention, with individual chapters published in high-level websites like GrokLaw and InformationWeek, CNET, LinuxToday, Linux Magazine, FSDaily, and around 550 additional sources; translated in other languages and

redistributed by libre software companies to their customers as reference material. The chapter on best practices has been turned into a chapter for an upcoming O'Reilly book on Open Government, in publication February 2010.

5 The results

This section describes the main results obtained along the project duration. For more information about these and other results, please read the FLOSSMetrics deliverables².

5.1 FLOSSMetrics infrastructure

5.1.1 Overview

The main aim of FLOSSMetrics was to construct, publish and analyse a large scale database with information and metrics about libre software development. Using existing methodologies and tools developed by members of the consortium, the project performed quantitative analysis of on several thousand software projects, allowing, for the first time, analysis and benchmarking based on robust large scale evidence.

To accomplish these tasks, FLOSSMetrics developed a powerful platform based on three components:

- **Retrieval System:** a system that downloads publicly available information from FLOSS projects that is available on the Internet and extracts facts and data from these sources using third party tools. Currently analyses data from three types of repositories: source code management systems, mailing lists and issue trackers.
- **FLOSSMetrics database:** the output given by the Retrieval System is stored in this database. The structure of the database has been designed to cover the various needs of the public, such as researchers or developers. For this reason the database is divided in several levels, according to the various studies that can be performed. The first level contains data extracted by the tools integrated into the retrieval system; the second level unifies the data of the previous level based on several criteria; the third level will contains analyses and statistics.
- **Melquiades website:** Melquiades is the name of the website interface designed with the purpose of making more visible and accessible the different results stored in the FLOSSMetrics database. In addition to a user interface designed for end-users, it also provides an extensive API for automated consultation of data.

The FLOSSMetrics infrastructure is available public under GPL license and included into the libresoft-tools package³.

5.1.2 Retrieval System

The Retrieval System is a software package that automates the retrieval (and partially, analysis) of data from public repositories about libre software development. It actually is mainly a front-end that organises and schedules the execution of a set of third party retrieval and analysis tools.

The high-level design of the retrieval system is shown in Figure 2.

²<http://flossmetrics.org/sections/deliverables>

³<http://tools.libresoft.es>

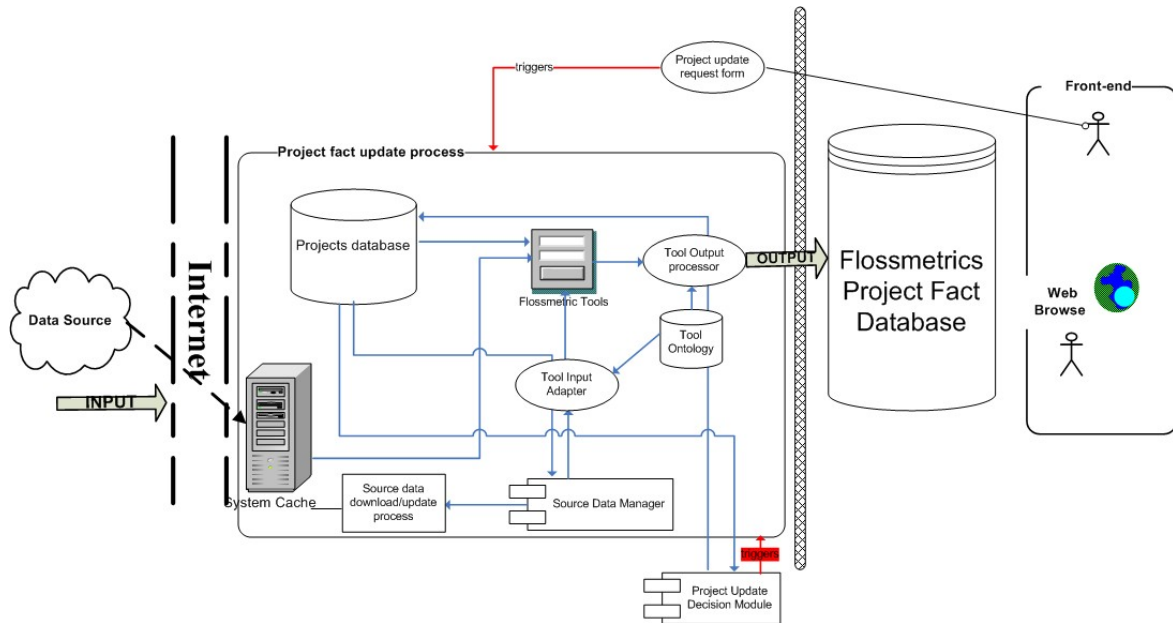


Figure 2: Retrieval System Design

As can be seen, the system is built with several modules that shared out the different tasks that the platform should accomplish. The system receives as input information about which projects will be retrieved and analysed. With the given data, the system search for the projects datas ource repositories. Then, it downloads the data and executes the available tools over them. Once the work has been finished, the results are stored into the FLOSSMetrics database.

During the data source search, the retrieval system is helped by the repository finder. Octopus - the code name of the repository finder - is a tool, also developed by the FLOSSMetrics team, that helps the retrieval system to find the usual repositories of information related to development tools for a given project. The tool is able to support different kinds of forges such as GForge and GForge-based sites and SourceForge, GNOME and Apache sites. The deliverable "D1.3 - Repository finder"⁴ describes this tool more deeply.

One of the main features of the system is that it has been designed to include external tools for the extraction of data and facts and the measurement of public repositories. The rationale for this is that there already exists a large number of tools that can be reused for this purpose. As many of them are released under a libre software license, embedding them in the system should be a characteristic to strive for. External tools should therefore be considered as plug-ins, that should be embedded into the system. This assumes a clear plug-in interface, so that new external tools can be easily included. The current version of the retrieval system supports CVSanaly (for the analysis of source code repositories), MLStats (for the analysis of mailing lists) and Bicho (for the analysis of issue tracking systems).

⁴<http://flossmetrics.org/sections/deliverables/docs/deliverables/WP1/D1.3-RepositoryFinder.pdf>

For more information about the retrieval system design and implementation, please read the documents "D2.1 - Retrieval System Design"⁵ and "D2.2 - Implementation of the Retrieval System"⁶.

5.1.3 Database

The database with information about projects is the central element of the FLOSSMetrics system. Data retrieved from public repositories via the Retrieval System are stored in it, and later, are publicly available in MySQL dump format through the Melquiades website.

The database currently⁷ contains information of the repositories from 3,150 projects, of which quantitative data is available about 2,800. The next tables summarize the status of the database:

Databases		Sources				Total
		SCM	MLS	TRK	Other	
First Level (Tools Level)		2311	623	4443	x	7377
Second Level (Unification Level)		1	1	1	x	3
Third Level (Analysis Level)	Variables	1374	470	2569	x	4413
	Studies	x	x	x	3119	3119
Total		3686	1094	7013	3119	14912

Figure 3: Databases per sources

Total of projects	3150
Projects with tool databases already available	2778
Analysed source code management	2013
Analysed mailing lists repositories	623
Analysed issue tracking systems	623
Projects with variables databases already available	2412
Projects with source code variables	1193
Projects with mailing lists variables	470
Projects with issue tracking variables	1476
Projects with studies databases already available	1639

Figure 4: Projects with databases

⁵http://flossmetrics.org/sections/deliverables/docs/deliverables/WP2/D2.1-Retrieval_System_Design.pdf

⁶http://flossmetrics.org/sections/deliverables/docs/deliverables/WP2/D2.2-Implementation_Retrieval_System.pdf

⁷November 20th, 2009

The database design and its implementation are fully described in the deliverables “D3.1 - Database specification”⁸ and “D3.2 - Database”⁹ or in the database page¹⁰ of the Melquiades’ wiki.

The structure of the FLOSSMetrics database has been designed to cover the various needs of the public, such as researchers or developers. For this reason the database is divided in several levels, according to the various studies that can be performed.

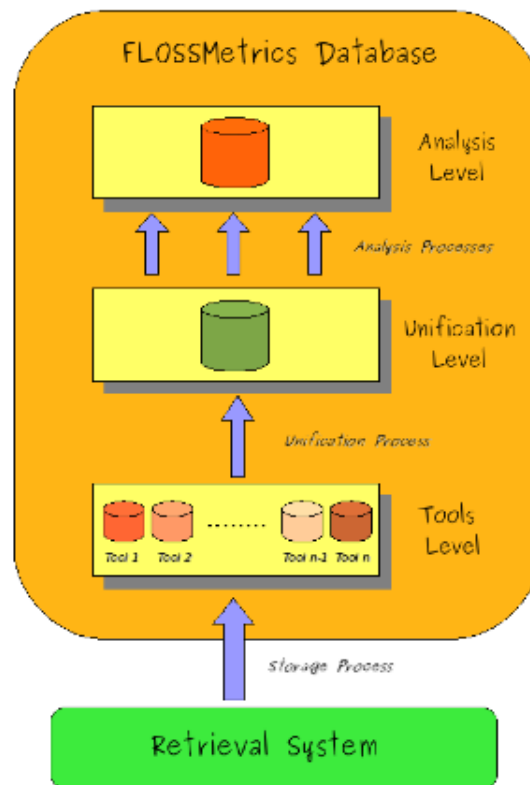


Figure 5: FLOSSMetrics Database Design

As it can be observed from Figure 5, the database is divided in three levels. The first (lower) level contains data extracted by the tools integrated into the retrieval system; the second (medium) level unifies the data of the previous level; the third (high) level stores analyses and statistics.

- **Tools Level.** This is the lower level of the database. As its name indicates, it stores data obtained directly from executing mining or extraction tools on projects’ data sources. The retrieval system is in charge of performing this work and managing the insertion of these data into the database.

⁸http://flossmetrics.org/sections/deliverables/docs/deliverables/WP3/D3.1-Database_Specification.pdf

⁹<http://flossmetrics.org/sections/deliverables/docs/deliverables/WP3/D3.2.-Database.pdf>

¹⁰<http://melquiades.flossmetrics.org/wiki/doku.php?id=database>

This level can be considered as a set of small and independent databases where each tool generates distinct data for its own database structure.

The main advantage of this level is that allows researchers to access specific data related to a specific project without managing large amount of data. For example, they can take the CVSAnALY database of the Apache httpd project to perform certain analyses about the committers of the project, without taking the database with the mailing lists from the same project, or without taking other CVSAnALY databases from other projects. The researchers will not have to manipulate data of projects that are not valuables for their studies, neither having to search nor filtering data in the database.

- **Unification Level.** The main objective of this level is to unify data from the same resource type of the previous level in just one database. To carry out this work of unification, a set of scripts have been developed. These scripts take the data from the tools level. Next, the data are mixed, and in a last step, after removing irrelevant, duplicated or incorrect data, the data are stored into the three databases of the second level, one for source code management systems (SCM), other for mailing lists (MLS) and finally another for issue tracking systems (TRK).
- **Analyses Level.** This level is fed by the analysis applications that performs their work over the data taken from the tools and the unification levels.

The level tries to minimized the time expended by researchers during their investigations. It stores a large set of distinct analyses, including common results and those who need long-time of process. If these analyses are available for the researches, they will not waste the time recalculating them.

5.1.4 Melquiades website

The Melquiades website¹¹ is the main interface to the FLOSSMetrics database. All the results and analysis from the analysed FLOSS projects are available on this website, through a powerful API and in different formats such as database dumps, charts, or even animations.

¹¹<http://melquiades.flossmetrics.org>



Figure 6: Melquiades website

The main features of the website are:

- **Melquiades API¹²**: an extensible API makes users to retrieve from projects, different types of data (descriptions, dumps, variables and metrics) based on several search criteria. Users can automatise their queries obtaining most of the data available in Melquiades in different formats (HTML, XML, JSON).
- **Dynamics charts**: this type of charts allows to display the information in a more useful way that the static one does. Users, interacting with the charts, can navigate along them knowing the value for every point in the chart, what happened in a specific point in time, etc.

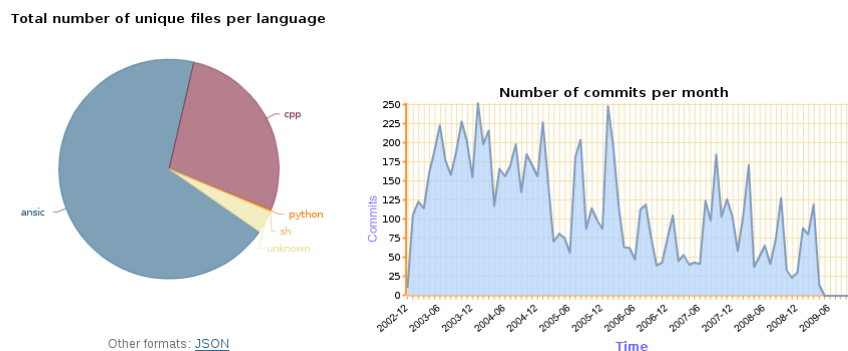


Figure 7: Dynamics Charts examples

¹²The API documentation is available on <http://melquiades.flossmetrics.org/wiki/doku.php?id=api>

- **Quality indicators:** community quality indicators are also available for some projects. They are based on the QUALOSS method v1.0 and allow to evaluate if the FLOSS community dynamics are at risk.
- **Melquiades Wiki¹³:** the wiki provides useful information about every detail of the FLOSSMetrics database and the Melquiades website. Database schemas, data dictionaries, SQL queries, APIs, frequently questions, tips and, definitely, all the knowledge that database users need is available here. In addition, this site has also been conceived as a research point, where every user can write her results and others can discuss, make comments or suggestions about them.

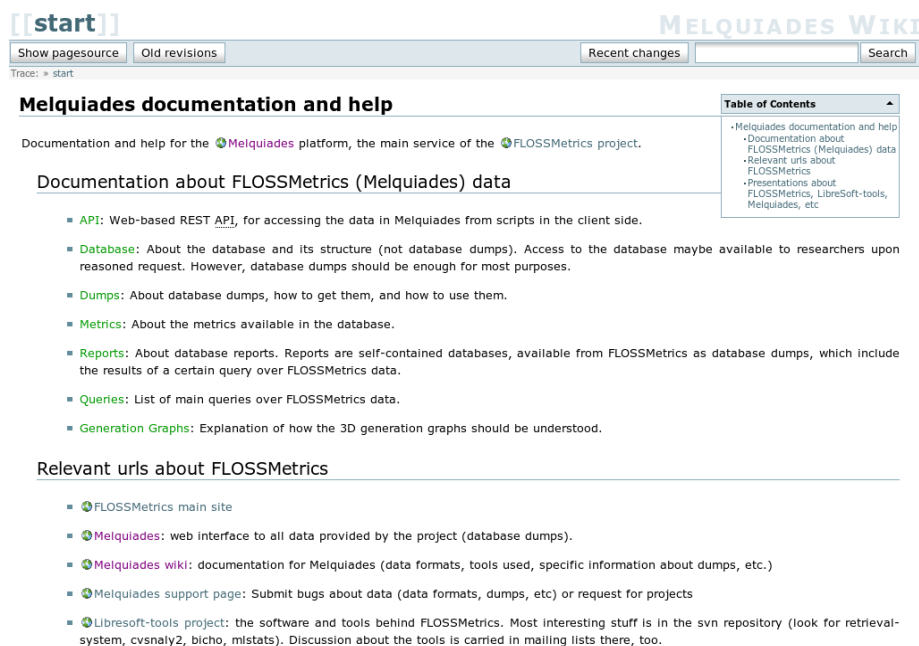


Figure 8: Melquiades Wiki main page

- **Melquiades support system¹⁴:** using this system, users can report about errors in the data sets; bugs in the website or request for new analyses of data sources.

The website offers a navigable list with the projects that are currently available on Melquiades. Each entry of the list contains the project name and a link to its web page on Melquiades, the forge that stores the project - if any - and can be followed by three symbols. These symbols represent different types of resources: a blue circle for source code management systems (SCM), a yellow diamond for mailing lists (MLS) and an orange square for issue tracking systems (TRK). If the symbol is present, there are analysed data from this type of resource.

¹³<http://melquiades.flossmetrics/wiki>

¹⁴<http://melquiades.flossmetrics/support>

From: To: [Get Projects](#)

Name	Forge	SCM	MLS	TRK
a3dx	SourceForge			
a8e	SourceForge			
aaf	SourceForge			
aai-portal	SourceForge			
abbot	SourceForge			
abills	SourceForge			
abiword	Unknown			
abl	SourceForge			
absolute-linux	SourceForge			
abstract	SourceForge			
ac3filter	SourceForge			
acbr	SourceForge			
acceleo	ObjectWeb			
accs	SourceForge			
ace	Unknown			

Figure 9: List of projects example

Each project in Melquiades has his own page where all its information is available. The project web page contains descriptions, links to the resources, database dumps, charts and quality indicators.

The screenshot shows the Melquiades Data website interface. At the top, there is a navigation menu with links for MAIN, NEWS, PROJECTS, DOCS, SUPPORT, and ABOUT US, along with a search bar. The main content area is titled 'gedit' and includes a description of the text editor, a 'Project Content' sidebar with links for Description, Results, Resources, Charts, and Quality Indicators, and a 'Dumps Corner' sidebar with links for Dumps Web Folder and Dumps WebDAV. Below the description, there is a 'Results' section with three tables of repository analyses:

SCM repository analyses			
Tool	Type	Date	Download
cvsanaly2	svn	2009-05-17	Download
cvsanaly2	svn	2009-02-28	Download
cvsanaly2	svn	2008-09-05	Download
cvsanaly2	svn	2008-08-31	Download
cvsanaly2	svn	2008-06-16	Download
cvsanaly2	svn	2008-05-14	Download








MLS repository analyses			
Tool	Type	Date	Download
MLStats	mls	2009-09-11	Download


TRK repository analyses			
Tool	Type	Date	Download
Bicho	trk	2009-09-20	Download

Figure 10: Project web page

The dumps are divided in the three categories described above and depending on the analysed repositories. A box for each category is presented and shows the different entries about the distinct available dumps. Each entry includes the name of tool used during the analysis, the type of the repository, the date when the analysis was performed and a link to the dump. These dumps are stored in a specific folder for each project that is accessible via HTTP or WebDAV protocols.

Results

SCM repository analyses			
Tool	Type	Date	Download
 cvsanaly2	svn	2009-05-16	Download
 cvsanaly2	svn	2009-02-26	Download
 cvsanaly2	svn	2008-09-05	Download
 cvsanaly2	svn	2008-08-30	Download
 cvsanaly2	svn	2008-08-26	Download
 cvsanaly2	svn	2008-06-13	Download
 cvsanaly2	svn	2008-05-13	Download

MLS repository analyses			
Tool	Type	Date	Download
 MLStats	mls	2009-09-11	Download


TRK repository analyses			
Tool	Type	Date	Download
 Bicho	trk	2009-09-19	Download

Figure 11: Dumps list

If any one of resources is analysed, then some dynamics charts are also provided. Depending on the resource type different data are shown. For instance: number of committers and commits per month, source lines of code per month, number of files per month, types of the files, for SCM; number of messages and authors per month, number of replies and threads per month, for MLS; number of reports and contributors per month, type of reports, for TRK. In addition, the page contains a table with a summary of the main results.

MLS Charts

These data are also available [here](#) (MySQL dump format)

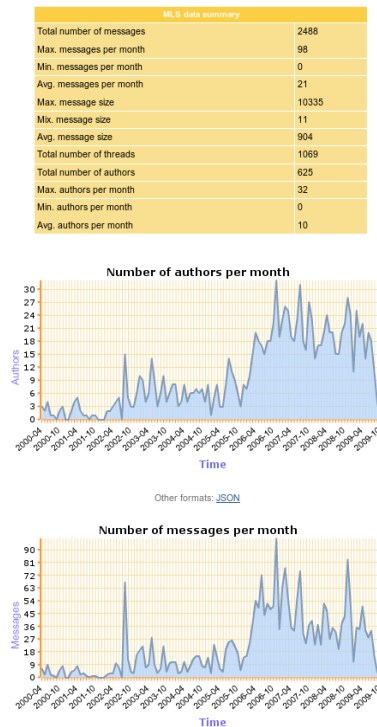


Figure 12: Project charts

5.2 High Level Studies

Note: The FLOSSMetrics consortium considers these studies as pilots. Therefore, the full description is included in the pilots section (section 6) of this document.

A total of 14 studies were undertaken using the FLOSSMetrics database. They do not intend to conform a comprehensive analysis of FLOSS development, but a set of examples of how the data collected by FLOSSMetrics can be used for research studies. The studies have in common some context related to software development covering some aspects such as: characterisation of FLOSS development, software evolution, quality, efficiency, maintainability, etc. Furthermore, two studies were performed as good examples of how the FLOSSMetrics data set could be helpful to industry: the first one investigates the productivity of FLOSS developers in socio-economics terms; the second one estimates the cost/estimation study based on substitution costs for FLOSS.

5.3 Guide for SMEs

There is no shortage of data and results that demonstrate that FLOSS, when adopted with appropriate best practices, can significantly lower costs and provide quality IT (information technology) solutions, especially for small and medium enterprises (SMEs). For

example, the Consortium for Open Source Software in Public Administration (COSPA) project demonstrated that by using best practices for FLOSS procurement, not only was software acquisition cheaper, but the evaluation of tangible and intangible costs over 5 years demonstrated a cost reduction ranging from 20% to 60%. The EU study on the impact of FLOSS indicates that OSS can reduce software research and development costs by 36%, while the INES project found that companies adopting OSS increased profits and reduced time to market and development costs in 80% of the trials.

If FLOSS is so advantageous, why is so little use of it perceived in the marketplace, especially among SMEs? Several groups studied the adoption barriers for libre software in companies and public administrations; among the examples we have selected, the 2008 CIO.com survey identified the following:

Adoption Barriers	Percent
Product support concerns	45%
Awareness/knowledge of available solutions	29%
Security concerns	26%
Lack of support by management	22%
Licensing or legal concerns	12%
Investment in architecture from other vendor(s)	20%
Software quality issues	20%
Customization concerns	15%
Not relevant to our product or service	7%
Pressure on open-source providers by commercial vendors	5%
Software cost allocation policies	2%
Other	9%

Table 1: Survey on FLOSS adoption barriers

Similar results were obtained by the tOSSad project¹⁵, where in the project deliverable D18 some of the barriers identified were:

- Reticence of users
- Resistance to change
- Lack of companies that migrate, combined with a lack of a local OS based economy creates a lack of confidence
- Perceived issue of “free” is not good does not offer quality
- Lack of business models that create confidence in FLOSS

¹⁵towards Open Source Software adoption and dissemination (tOSSad) <http://www.tossad.org/>

It is clear that what was missing in terms of adoption was a source of information that would had to cover not only technical aspects, but legal and business-related as well; in particular, the issue of long-term sustainability was deemed important by most of the analysts that were covering the field.

For this reason, the FLOSSMetrics project included a set of activities designed to spread information on OSS for small and medium enterprises, and to facilitate this adoption process. The activities were centered on a freely-licensed document, the FLOSSMetrics SME guide, designed to collect and provide the information in a way that was clear, transparent in terms of sources and data, unbiased and direct. The material was designed with two different audiences in mind: companies interested in using libre software, and companies interested in providing services based on libre software, or that were interested in integrating FLOSS components in their own offering. The guide ultimately integrated original research, and the work of several past EU projects, like SPIRIT, COSPA, CALIBRE, OpenTTT, INES and many others; as well as reputable research sources.

Within a company, the value that comes from FLOSS can derive from several different areas:

- Basic substitution/migration: the use of FLOSS in the IT infrastructure, frequently in substitution of a proprietary software.
- New deployment: the introduction of FLOSS for a new project internal to the company (adoption).
- Selling services based on FLOSS.
- Selling products that contain FLOSS as a significant component.

In this sense, a company may find useful FLOSS from a tactical point of view (FLOSS is cheaper to implement, with less constraint from a traditional vendor, or may help in introducing products in a reduced time to market) or a strategic point of view (creation of new markets, adoption of different business models). To be sustainable, a company must adopt a business model that provides a way to turn the FLOSS adoption into lower costs or increased revenues, and must also take into account the fact that at least a part of the participant community may be out of control of the company (as it commonly happens in large scale FLOSS projects, most contributors are not working for a single company). Our guide was designed to help companies in every step of the ladder of FLOSS adoption¹⁶.

¹⁶Figure adapted from Carbone P, "Value Derived from Open Source is a Function of Maturity Levels"

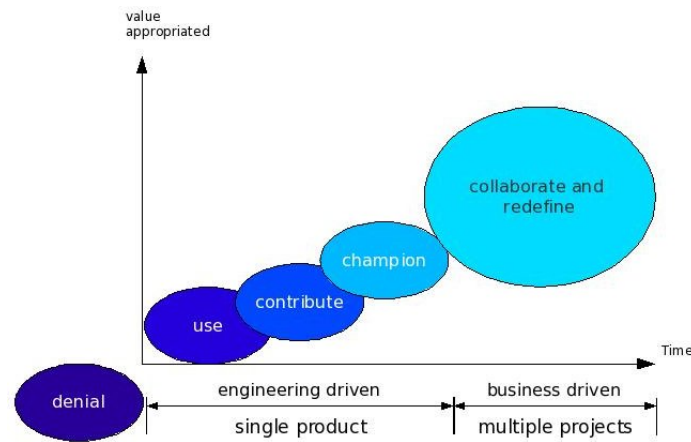


Figure 13: Ladder Model of OSS Adoption

For this reason, we focused the content across broad “chapters”, connected together. The guide content is composed of:

- A SME Guide Introduction
- What’s Free/Libre/Open Source Software?
- Ten myths about free/libre open source software
- Basic FLOSS adoption models
- Finding and selecting software
- Best practices for FLOSS adoption
- FLOSS-based business models
- R&D sharing
- Bibliography
- Appendix 1: estimating the number of active FLOSS projects
- Appendix 2: USB-based SME toolkits

With chapters 1, 2, 4, 5, and the guide catalogue designed for companies interested in start using an OSS tool, and chapters 3, 4, 6, 7 designed for companies interested in starting a business based on FLOSS. The guide catalogue is designed to provide an initial list of packages that are stable and relevant for several different industry sectors; the design of categories and the initial selection was performed in the EU OpenTTT project, where a large scale survey of industry needs was performed, along with an initial match with selected open source applications. The methodology has been used in FLOSSMetrics to help in finding software that was deemed interesting by the targeted end users.

The software catalogue covers software categories and utilities such as: security, data protection and recovery, virtualisation and remote access, network and server management, database management, software development, ERP, CRM, Groupware, VoIP, conferencing and messaging, document management, vertical business applications, content management systems, e-learning applications, graphics Video and CAD, desktop applications, engineering and manufacturing, health care.

6 Pilots and studies

FLOSSMetrics project designed and performed several pilots or studies in order to show the usefulness and applicability of the provided data. This set of 14 studies covers different software engineering areas such as maintainability, quality or efficiency, but also some others from economics or management.

Please notice that these studies should be considered just as examples of cases that can be performed with the FLOSSMetrics database, since many others are possible. But using these studies as starting point, or even as templates for specific studies, researchers or research groups will find it easier to start working with the information which is provided to them by FLOSSMetrics.

The next list contains the description of each FLOSSMetrics studies. For more information the deliverable "D5.1 - High Level Studies"¹⁷ contains the description and results of these studies.

- *On the Validity of The Laws of Software Evolution.* This study tries to determine the speed in which the activity within the lifetime of the project has taken place.
- *Characterization of the Evolution Dynamics of Software.* Software evolution still lacks a theoretical model that explains why and how software evolves. Some studies have proposed that evolution dynamics is a self-organized criticality (SOC) dynamics. This study tests whether or not that dynamics model verify for the projects stored in FLOSSMetrics database.
- *Effort.* This study is based on the comparison between data from FLOSS projects and one version of the COCOMO model for proprietary software suggests that FLOSS productivity and effort estimation models are needed. The model presented here gives a first step to approach this research question and better models may be obtained refining and improving the measurement methodology. In this case, the model developed may be used to measure the possible impact of new or improved processes, methods and tools.
- *Correlation Size and Complexity Metrics.* One of the goals of software engineering is to measure different aspects of software projects, with the aim of finding a small set of attributes that may characterize them. In the case of this study, the first motivation is to find out which independent metrics may be used to characterize size and complexity.
- *Evolution of Core Team Members.* This study shows a methodology intended to characterize the evolution of the core teams of libre software projects. The results shown for the case of Evince show that in that project underwent a generational relay, and those top contributors at the beginning of the project are not contributing any more. This relay was very lively in the past history of the project, and it seems to have stopped in the recent history of the project, where the top contributors seem to have been present since some time ago.

¹⁷<http://flossmetrics.org/sections/deliverables/docs/deliverables/WP5/D5.1-HighLevelStudies.pdf>

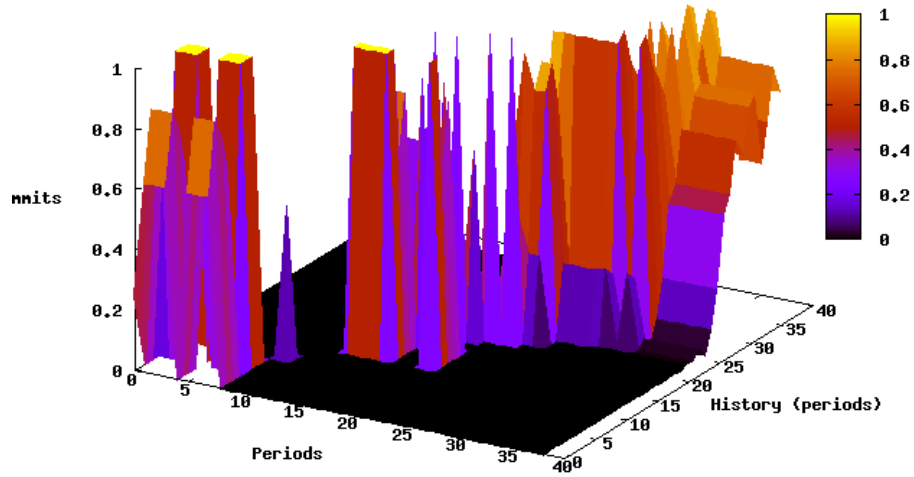


Figure 14: Evolution of core team members in Evince

- *Evolution and Dynamics of Bugs.* The main goal of this study is to understand the bug fixing and reporting process in open source projects (to know how many bugs are reported and the average time it takes to fix a bug in a project's lifetime). It will help to find whether there is any significant correlation between the size of the codebase and the number of bugs being reported.

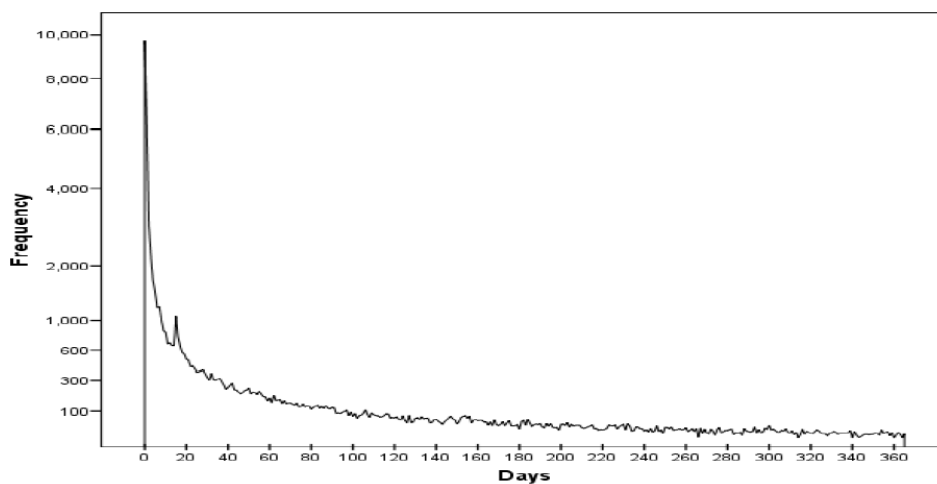


Figure 15: Closed bugs frequency

- *Quality in Open Source Software.* When acquiring software, enterprises are not only interested to know about the product and its quality but also interested in who produced that product and its reputation. For traditional enterprises, reputation can be check based on financial strength of the software provider however, for the FLOSS world, we must find other ways to determine if FLOSS endeavour (or FLOSS project) is serious. This can be done by studying the behaviour of a FLOSS community. In particular, a FLOSS community should behaves in a manner to convince potential FLOSS integrators from Industry that it is dependable. The objective of this study is to measure how feasible is a community.
- *Efficiency.* In FLOSS projects, normally the effort invested is unknown, and therefore might need to be estimated, and is also more diverse than in commercial projects, as it includes core team member, committers, bug reporters and several other groups with varying intensity of participation. The output of a project can be measured using several software metrics like most easily the number of LOC, files, checkins to the source code control system, postings, bug reports, characteristics of development speed (e.g. coefficients of a software evolution equation estimated) or even metrics for product attributes like McCabe's cyclomatic complexity. This range of metrics both for inputs and outputs, and their different scales necessitates application of an appropriate method. This study applies the Data Envelopment Analysis (DEA) method that returns an efficiency score for each project.
- *Project survival.* Since the beginning of free/open source software development thousands of new projects have been initiated. Out of these projects, a small fraction are actually alive and continuously developed, while the majority of them are abandoned. The potential outcome of this kind of study is to determine the probability of the time point at which someone can claim that a free/open source project is inactive and thus considered as abandoned.

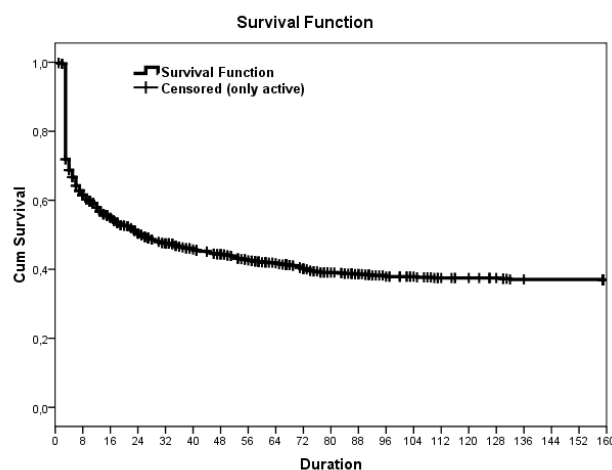


Figure 16: Estimation of the survival function

- *Maintainability Assessment.* The purpose of this study is to define whether there exists a measurable critical point that determines the route of a project as a possibility. The study attempts to determine entrance thresholds for a FLOSS project, i.e. activity related rates (bugs/time, code/time etc) that signal the attraction of many users/developers and therefore the creation of a critical mass of project participants.
- *Development vs. Communication.* The aim of this study is to decide the degree to which developers participate in mailing lists too. Findings of this study would provide further evidence about the way software knowledge circulates within FLOSS projects.
- *Contributor activity.* The objectives of this study are: to discover the quantitative dependence of individual contributor productivity on time, both in total but also per contribution type (functional change in code, bug fix, bug report); to determine the average time dependence for all contributors; through an appropriate statistical analysis, determine whether there are clusters (groups) of contributors with different productivity time dependence and provide time curves for each such group
- *Substitution Cost Estimation.* Substitution cost is the monetary value of the effort necessary for implementing a FLOSS application from scratch in a software company. This monetary amount has many potential uses, e.g. it may be used to estimate the gains from reusing a FLOSS component instead of building it from scratch. This pilot has helped identifying various problems, limitations and issues related to the data and the model precision. The ultimate target is the application of those models on the entire FLOSSMetrics code base.
- *Productivity.* This study develops a spot productivity estimator function based on the survey data combined with data from the source code management repositories and applies this to the projects covered by the survey.

7 Recommendations and guidelines on how to use FLOSSMetrics' results

Note: this section is based on the contents of the "Exploiting FLOSS Research Results for Industry" booklet¹⁸.

Users that are interested in using the FLOSSMetrics results should keep to the following steps in order to maximise the efficiency of the information collected from the FLOSSMetrics repository and to increase its quality.

1. Define what you are looking for and make a list of the information you need

- If you are interested in a FLOSS product or a FLOSS component, FLOSSMetrics can provide you with code as provided by source code management repositories of thousands of FLOSS projects and helps you to decide which package would be the one serving best your particular needs. Searching the FLOSSMetrics website can thus considerably shorten your searching time.
- If you are interested in information about the quality of a certain FLOSS product or component, the FLOSSMetrics database provides you with a number of metrics that help you to make a quality assessment (e.g. the number of bugs and bug fixes, the speed in which bugs are usually fixed, the evolvability and robustness of the code, and the like)
- If you are interested in information about the sustainability and reliability of a FLOSS project, the FLOSSMetrics database provides you with measures that help you to assess the community's vitality, the number of committers, the release schedule, and the like.

2. Make yourself familiar with the FLOSSMetrics website and database

- Check the FLOSSMetrics information sources (via Melquiades site) and get an overview of what is available.
- Collect the information you need.
- Be aware that besides the graphs and data showing trends of general and common interest, there are independent, detailed and open information (in raw data format, not necessarily in graphical form) of the inner life and characteristics of the projects. This is a rich source of information which could give you even deeper insight into projects you are interested in and help you in decision processes.
- Make a list of what information is not provided and inform the maintainer of the both FLOSSMetrics and Melquiades websites and database about these gaps - this will help to continuously increasing the scope and quality of the information that is provided to companies and SMEs. You can do that using

¹⁸<http://flossmetrics.org/docs/ExploitationBooklet.pdf>

the issue tracking system.

3. Evaluate the information you collected from the FLOSSMetrics website and database and give feedback

- Check whether the information that you collected meets your demands.
- If needed, decide what you can do in order to enhance the information quality (e.g. by running tests on the code that have not been done before) or if you want to employ third parties (e.g. a software services company) for this purpose.
- Decide whether or not to use / implement the code / information you have retrieved from FLOSSMetrics.
- Consider to feedback your results to the community and the FLOSSMetrics website and database. Please consider to use, as a good place for sharing them, the Melquiades wiki, where other users can view, comment and discuss your results.

4. Consult the the Small/Medium Enterprise (SME) guide to Open Source Software

- The SME guide is a rich resource that has already been used in a wide variety of contexts and by various actors, including companies, associations, open source vendors, universities and non-profit associations.
- The guide generally addresses two strands information: companies interested in start using an FLOSS tool should consult chapters 1, 2, 4, 5; companies interested in starting a business based on FLOSS consult chapters 3, 4, 6, 7.
- The following list of common questions will help you further navigate the guide and get an idea of what issues and information the guide can assist you with:
 - *What is open source?:* Section 1.1 "What is open source software?"
 - *Do I have to pay to use FLOSS? Is it really free?:* section 1.1 "FLOSS as a licensing model"
 - *How is it possible for FLOSS to be free?:* section 2.8 "Myth #8: There is no money to be made on FLOSS"
 - *Someone told me that FLOSS is not reliable, and is not good enough quality for use by companies:* section 2.2 "Myth #2: FLOSS is not reliable or supported", and section 2.3 "Myth #3: Big companies don't use FLOSS"
 - *I just want to know what kind of software is available:* see Software Catalogue
 - *There is too much software! How can I choose?:* chapter 4, "Finding and selecting software"
 - *What are the licensing constraints? If I use OSS, what happens?:* chapter 1 "What's Free/Libre/Open Source Software?"

- *I decided to use OSS inside of our company/institutions. What is the best way to proceed?:* chapter 5, "Best practices for FLOSS adoption"
- *How can I decide if it is convenient to use OSS inside of a product?:* chapter 7, "R&D sharing"
- *I want to sell services or products based on an open source component. What should I do?:* chapter 3, "Basic FLOSS adoption models", chapter 6, "FLOSS-based business models" and chapter 7, "R&D sharing"

5. Check the Research Community for Resources

- FLOSSMetrics is an enabler of research, and the research community is using FLOSSMetrics data and delivering results and benchmarking that could be of interest to industry.
- FLOSSMetrics provides a metrics dictionary which translates the available metrics from a software engineering point of view to a socio-economic one.
- FLOSSMetrics has performed two studies as good examples of how these data could be helpful to industry: the first one investigates the productivity of FLOSS developers in socio-economics terms; the second one estimates the cost/estimation study based on substitution costs for FLOSS.

6. Tools and Infrastructure

- FLOSSMetrics infrastructure is based on a tool (the retrieval system) that automatizes the analysis of public developing repositories using third party tools as plug-ins. The results are stored in a database and later are accessible via a web site interface named Melquiades.
- Both retrieval system and Melquiades interface are included into the libressoft-tools package¹⁹.
- Other tools used in FLOSSMetrics to analyse repositories such as source code management systems (CVSAnaLY tool), mailing lists (MLStats tool) or issue trackers (Bicho tool) are also available in the libressoft-tools package.
- All the tools used and developed during the FLOSSMetrics project - our own and third party ones -, are FLOSS.
- Consider the cost and benefits of installing the FLOSSMetrics tools and/or infrastructure stand-alone to do your own analysis.
- Get inspired by looking at how the tools and data is used in development forges, such as the Open Source Observatory and Repository (OSOR)²⁰

7. Consider to submit your project for evaluation

¹⁹<http://tools.libressoft.es>

²⁰<http://forge.osor.eu/plugins/metrics/index.php?id=13&type=g>

- If you do not wish to install the tools and/or infrastructure, an independent, vendor-neutral FLOSS Competence Center can perform an evaluation of your project. Decide whether the cost is larger than the benefit of performing the analysis. Such an analysis could be useful for the project to evaluate in terms of quality, maturity, security, amongst others.

8 Availability of Results

The next list provides a summary of the availability of each result that the FLOSSMetrics project has generated.

*Note:*FLOSSMetrics documentation is distributed under "Creative Commons Attribution-Share licenses" and source code under different FLOSS licenses.

- **Deliverables**²¹: all the deliverables are public but those related to the management workpackage (WP12) and the annexes of deliverable D5.1 "Software Engineering Studies" due to copyright issues. The documents
- **Database**: the FLOSSMetrics database is public and available in MySQL dump files via the Melquiades web site.
- **Tools**: the tools implemented (retrieval system, repository finder, bicho) and used (CVSAnalY2, MLStats, etc) are public²² and their source code is distributed under different FLOSS licenses.
- **Melquiades website**²³: all the its contents are public. The source code of the website is also public and available under GPL v3.0 license.
- **Melquiades wiki**²⁴: all the information included in this wiki is public available under CC Attribution-Share Alike 3.0 license.
- **Guide for SMEs**: the guide²⁵ and the contents of the wiki²⁶ are public.
- **FLOSSMetrics website**²⁷: all the contents of the website are public.
- **Description of Work**²⁸: the description of work is public but its economy section. This document is available in the FLOSSMetrics web site.

²¹<http://flossmetrics.org/deliverables>

²²FLOSSMetrics tools are included into the libresoft-tools software (<http://tools.libresoft.es>)

²³<http://melquiades.flossmetrics.org>

²⁴<http://melquiades.flossmetrics.org/wiki>

²⁵<http://guide.flossmetrics.org/smeguide.pdf>

²⁶<http://guide.flossmetrics.org>

²⁷<http://flossmetrics.org>

²⁸<http://flossmetrics.org/docs/DoW-3.0-public.pdf>

9 Partners



Jesus M. Gonzalez-Barahona

jgb@libresoft.es

<http://libresoft.es>

Libre Software offers Software Engineering scientists the possibility not only of having a closer look at the product that is being created, but also of studying in detail the whole development process and its technical, social and economic consequences.

The main research topic at the Universidad Rey Juan Carlos is the quantitative measurement of libre software development patterns and characteristics in order to gain knowledge on the process, mainly by studying the different agents that participate in it, the use of the different development and development-supporting tools as well as the methods that have been followed. The main focus is technically oriented having principally an engineering perspective of the research area in contrast to other research groups which are primarily centered on social and economic aspects.



Frank van der Linden

<http://www.philips.com/>

Philips Medical Systems Nederland B.V. offers a robust portfolio of medical systems. The goal of each product is clear – faster and more accurate diagnosis and treatment. Our product line includes best-in-class technologies in X-ray, ultrasound, magnetic resonance, computed tomography, nuclear medicine, PET, radiation oncology systems, patient monitoring, information management and resuscitation products. We also offer a wide range of services including, but not limited to, training and education, business consultancy, financial services and e-care business services.



Rishab Ghosh

<http://www.merit.unimaas.nl/>

UNU-MERIT is a joint research and training centre of United Nations University (UNU) and Maastricht University, The Netherlands. The joint Institute was created on 1 January 2006 following the integration of the former UNU-Institute for New Technologies (INTECH) in Maastricht, and the Maastricht Economic Research Institute on Innovation and Technology, MERIT, at Maastricht University. UNU-MERIT provides insights into the social, political and economic factors that drive technological change and innovation. The Centre's research and training programmes address a broad range of policy questions relating to the national and international governance of science, technology and innovation, with a particular focus on the creation, diffusion and access to knowledge.



Stefan Koch

<http://www.ai.wu-wien.ac.at>

Wirtschaftsuniversitaet Wien is one of the largest business schools in Europe, with nearly 400 faculty and about 22,000 students in 6 undergraduate and graduate programmes and 12 post-graduate studies. The Department of Information Systems and Operations with its institutes is a leader in business informatics. The Institute for Information Business, partner in FLOSSMetrics, focuses on the influence of information on the economy and on conducting business, with research efforts on novel results in context of timeless economic concepts concerning all stages of the information life cycle. Current topics include information systems development, introduction and marketing, IT-Controlling and infrastructure management, and knowledge processing and management.



Ioannis Stamelos

<http://www.auth.gr/home/>

The Aristotle University of Thessaloniki is the largest Educational Institute of Greece with over 2000 permanent staff and 60000 students. The Programming Languages and Software Engineering Laboratory (PLASE) belongs to the Computer Science Department. The Laboratory was officially constituted 7 years ago and its target is research and teaching of various fields of Informatics such as Information Systems, Software Engineering, Artificial Intelligence and Electronic Commerce. During the last years the laboratory is organizing important panhellenic and international scientific meetings (Conferences, Seminars, Workshops). In addition, recently, AUTH was ranked 9th worldwide and 2nd in Europe, among the top institutions (both from academy and industry) in the field of Systems and Software Engineering by the Journal of Systems and Software (Elsevier). This classification was released in 2006 and takes into account the quantity of publications institutions make in six international journals related to this field.



Conecta was born in 1995 with a specific focus on libre/free open source software. During these 11 years, we focused on collecting knowledge on the many less known open source packages, and thanks to the participation in many European research project we had the opportunity to build one of the largest database of open source software in many different thematic areas.

Carlo Daffara

<http://www.conecta.it/>



Zea Partners (Zea) is a non-profit business partner network for companies building Zope systems. Zea includes founders of Zope, Silva, and Plone, along with leading SMEs (Small/Medium-sized Enterprises) around the world. The partnership of Zea is focused on growing open source and open source business by pooling resources for collective action.

Xavier Heymans

<http://www.zeapartners.org>