# Guilty: MetricsGrimore Ingregration

Sergio Arroutbi Braojos

December 8, 2012

## 1 Introduction

This document is an approachment to the different actions that must be carried out for Guilty tool to be included in MetricsGrimoire forge.

Guilty is a tool written by Carlos Garcia Campos and hosted in LibreSoft. This tool allows dumping, in different output formats, the result of a typical "blame" control version subcommand to dump for a particular file the author, date and commit of the different lines contained in itself.

On the other hand, the existance of MetricsGrimoire Free Software tools forge allows performing different metrics through an Open Source project, in order to cathegorize the status of the project in terms of development status, people involved, mailing activity, and so forth and so on.

In particular, MetricsGrimore analyzes three main aspects for each particular project, focusing on:

1. **Software Configuration Management**
2. **Bug Tracking Tools**
3. **Mail Lists**

Different tools exist on MetricsGrimoire to achieve a deep analysis on the previous environments. This document focuses on Software Configuration Management (SCM), and how Guilty application can help on completion of the metrics being analyzed nowadays by MetricsGrimoire.

## 2 SCM on MetricsGrimoire

Metrics Grimoire provides powerful tools to inspect, export, diagnose and analyze the status of a particular Free Software project software configuration manager. Available tools in MetricsGrimore regarding SCM include:

### 2.1 CVSAnalY

This tool allows, starting from a path containing the source code from a particular Free Software project repository, to dump to a MySQL database a wide set of data associated to the history of that particular project.

Despite its name, CVSAnalY tool supports dumping information from different repositories apart from CVS. In particular, data from Subversion and Git repositories are also supported. Moreover this, Bazaar and Mercurial are also planned to be supported. Among the information data dumped by CVSAnalY, a variety of information is available, such as:

- repositories: identifying the repositories associated to the project, as more than one can exist.
- files: identifying the different files and directories hosted on a particular repository.
- actions: typical actions performed on a file, such as modification, deletion or addition.
- people: different people who ever interacted with the repostory.
- scmlog: identifying commit number and commit messages.
- branches: with the existing branches on the repository.
- tags: showing the different tags created on the repository.

Each of the previous elements is stored in the database, in general with a particular table for each of the data linked, existing relationships among them. For example, a particular file belongs to a repository, an action is related to a file, a branch and a commit, etc.

Moreover this, CVSAnalY tool provide extensions. This extensions allow the completion of the database, with additional information. Examples of extensions provide extra information such as Weeks, Months, File Types, Metrics, etc.

# 3 Guilty

From MetricsGrimoire perspective, although some extensions analyse the lines of code (LOC) being changed on a particular commit, there is not a strict control regarding the LOC hosted on a particular repository.

Taking this lack into account, there is an oportunity to take profit from existing tools that allow a deep inspection of the LOC changed on a particular file, the commit associated to it, the user who performed it and the date where it was completed.

As stated previously, Guilty tool, hosted on Libresoft Git repository, allows performing this type of analysis, line of code oriented, and providing information for each line of code having to do with SCM.

In this aspect, the main purpose will be, **in a first stage, to introduce Guilty application in MetricsGrimoire forge**, allowing to perform a completion of the MySQL database used by CVSAnalY for metrics storage, including a new table to this database, that will keep different entries, one per line of code, containing the author, the date, the file, the line number, the last revision.

**In a second stage**, the desired situation will be refactoring Guilty code, in order **to achieve Guilty to be an extension of CVSAnalY**, in order to provide a unique tool to perform all the diagnostics having to do with SCM.

Regarding metrics to be extracted from Guilty, next list shows some examples of the different possibilities:

- LOC changes on last week/month/year
- LOC changes on a certain preconfigured period
- LOC changes by file and/or commit in a certain period
- LOC changes by file and/or user in a certain period
- LOC changes multiplied per user in a certain period (in order to prioritize activity from different developers)
- LOC changes multiplied per user in a certain period (in order to prioritize activity from different developers)

# 4 Conclusion

Guilty tool provides some enhancements that can help on contributing to Metrics Grimoire, in order to achieve a wider collection of metrics data to be dumped and later analyzed.

A deeper study of this tool, together with an enforcement of its funcionalities and performance, in addition to further development of the metric processing tools hosted in Metrics Grimoire to take profit of this new data and help on the accomplishment of a powerer toolset.