# Data

Concerning the data that WP3 is going to analyze and that concerns your User Case:

## GitLab Reader

As you indicate in your User Case, the information will come mostly from GitLab. Currently, we do not have a reader for GitLab, as we expect to use a client generator developed by YORK. However, in the case the client from YORK is not possible to modify to read GitLab, we will provide a reader based on the technologies that currently are used for other readers like JIRA or BitBucket. Therefore, we assure that CROSSMINER will be able to retrieve information from GitLab.

## JIRA Reader

This reader is implemented and working. It can retrieve issues and their corresponding comments.

## SYMPA mailing list

We have a first version of the reader ready to be integrated in CROSSMINER. The reader can read from email archives.

## IRC reader

As discussed recently in our project meeting in L’ Aquila and later on 31/10 via email exchange with Martin, there is no API for IRC. Our only way of processing IRC discussions is by parsing log files.

Martin noted that logging conversations is restricted by GDPR, raises considerations and should be handled carefully.

Martin suggested that we can develop a parser for Ubuntu's IRC logs and if we get good results OW2 would be very happy because it would means this could be a new data source for your MRL model and you may decide to implement bots in the future.

# Scenarios

In this part, we will discuss the elements that we will provide concerning the two scenarios that are described in OW2’s User Case.

## Project Maturity Level

The goal of this scenario is to evaluate how well CROSSMINER is able to determine the maturity of a project, but also, to improve the information and knowledge regarding other projects for either current developers or newcomers.

In this aspect, there are different elements that WP3 will be able to provide for the analysis of maturity. These are explained as follows:

### Historic Metrics

Using different NLP tools and the data retrieved with our different readers, we can provide metrics. These metrics viewed from a larger perspective can be used to determine how mature a project is. The historic metrics show how a project evolves with respect to a variety of aspects.

#### Issue tracking systems

For the issue tracking systems, like JIRA or the one included in GitLab, we support metrics related to the following aspects:

* Number of bugs: It indicates how many bugs, through the time, are in total all the issue tracking systems related to a project.
* Number of comments: This metric indicates, through the time, the evolution in the quantity of comments for all the bugs related to a project.
* Emotions: This metric expresses, grosso modo, the emotions found in the issue tracker for a given project.
* Number new bugs: It indicates how many bugs are created every delta through a lapse of time.
* Number of new users: How many users are new in the bug tracking system.
* Open time: This metric shows the average open time of an issue.
* Number of patches: It indicates the number of patches located in the issue tracking system.
* Number of request and replies: From all the comments in the issue tracking system, how many of them are replies and how many are requests.
* Response time: It presents the average time to reply a comment in an issue tracker.
* Sentiments: Which sentiments are found in a issue tracking system for a project.
* Severity: The severity level of the bugs found in the issue tracker. This metric classifies a bug report into 1 of 7 severity levels such as Blocker, Critical, Major, Normal, Minor, Trivial or Enhancement.
* Status: It indicates the issues status, like open or closed.
* Topics: Using clustering methods, this metric indicates the topics that are discussed through the time.
* Unasnwered bugs: It indicates how many issues do not have any reply.
* Users: How many users are found in a issue tracker system.

As well, there are some multi-aspects metrics like “Severity – Bug status” or “Severity – Response time”. These metrics can make the understanding of a project easier and to know which are they advantages or disadvantages.

#### Newsgroups

The metrics regarding newsgroups are fully implemented and can provide the following information:

* Number of articles; It determines the number of articles found in a newsgroup.
* Emotions: Which emotions are found in the newgroup.
* Number of threads: How many threads a newsgroup contains. In other words, how many different branches have been created since the first email was sent to the newsgroup.
* Number of new users: How many users make use of the newsgroup.
* Number of request and replies: From the total number of emails belonging to the newsgroup, how many of them are request and how many are replies.
* Response time: The average time to reply a message.
* Sentiments: The sentiments found in the newsgroup.
* Severity: The severity level of the messages exchanges in the newsgroup. The severity is measured in terms of how severe an issue is.
* Number of new threads: How many threads are created per delta.
* Topics: Using the clustering algorithm, we determine which are the most frequent topics discussed in the newsgroup.
* Number of unanswered topics: How many messages haven’t been answered.
* Number of users: Number of users that make use of the newsgroup.

Similarly to the metrics for issue tracking systems, there are some multi-aspects metrics.

#### Forums

These metrics are currently being implemented, most of them are going to be similar to those provide for newsgroups, however, the final metrics depend on the evolution of Eclipse forums API. At the current state, we have the following metrics in progress of being implemented:

* Number of posts: For every topic (i.e. forum thread), how many posts in total exist.
* Emotions: Which are the emotions located in the forum.
* Number of topics: How many forum threads contain the forum.
* Number of request and replies: From all the posts in the forum, how many are request and how many are reply.
* Sentiments: Which are the sentiments that can be found in a forum.
* Severity: In the case the forums are used to express issues, we can determine their severity.
* Number of new topics: Number of new forum threads created.
* Topics: Number of content topics, i.e. subjects, are discussed in the forum.
* Number of unanswered topics: Number of forums threads which do not have a reply.

### Documentation analysis

This work is still in progress as it is part of our last deliverable on Month 30. We have detected that you need tools to determine whether a project contains documents like: Start Guide, User Guide, Developer Guide, as well as, examples for developers or whether the documentation contains a license.

If you can provide us with a large number of examples of these documents, it can help use to research tools that would be able to find these documents or elements.

## User/Developer Scenario

In this scenario, you expect to search, using natural language, different types of information, from code snippets to configuration files, passing through mailing list. As well, you expect to determine if a text makes reference to one or several bugs. Furthermore, you want to determine the sentiment of messages and group messages to suggest possible answers to users or developers.

Thus, at the current state, we are able to provide indexes of different sources such as:

* issue tracking systems
* forums
* newsgroups
* Stack Overflow
* Hacker News

These indexes will allow doing queries in natural language. Moreover, they will provide elements to filter like data source, sentiment, emotions, whether they contain code or not, among others. Most of them will match with the historic metrics explained previously.

Regarding the similarity of messages, the indexing system will be able to provide access to information, from a variety of sources, that are similar to a specific query.