

---

Copyright (c) 2015-2017 Castalia Solutions and others.

All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0

which accompanies this distribution, and is available at

<http://www.eclipse.org/legal/epl-v10.html>

**Contributors:**

**Boris Baldassari - Castalia Solutions**

title: "Alambic – R analysis document" author: "Boris Baldassari" output: pdf\_document: toc: true toc\_depth: 3 —

## Summary

This plugin generates a PDF document with information about project **technology.epf**.

This plugin is intended as an example of R Markdown document to help people easily setup their own R analysis on software development data.

# Metrics

Mnemo	Value
DOC_GOV	1
GOV_BOARD_PUBLIC	1
ITS_AUTHORS	222
ITS_AUTHORS_1M	1
ITS_AUTHORS_1W	0
ITS_AUTHORS_1Y	2
ITS_CREATED_1M	1
ITS_CREATED_1W	0
ITS_CREATED_1Y	2
ITS_DIVERSITY_RATIO_1Y	1
ITS_ISSUES_ALL	3220
ITS_OPEN	328
ITS_OPEN_OLD	0
ITS_OPEN_PERCENT	10
ITS_OPEN_UNASSIGNED	0
ITS_UPDATED_1M	1
ITS_UPDATED_1W	0
ITS_UPDATED_1Y	2
MLS_USR_AUTHORS	346
MLS_USR_AUTHORS_1M	0
MLS_USR_AUTHORS_1W	0
MLS_USR_AUTHORS_1Y	0
MLS_USR_DIVERSITY_RATIO_1Y	0
MLS_USR_POSTS	1186
MLS_USR_POSTS_1M	0
MLS_USR_POSTS_1W	0
MLS_USR_POSTS_1Y	0
MLS_USR_THREADS	1186
MLS_USR_THREADS_1M	0
MLS_USR_THREADS_1W	0
MLS_USR_THREADS_1Y	0
OSS_DEP_CHECK	1
OSS_ESCALATE	1
OSS_INCLUSION	1
PROJECT_ACCESS_INFO	1
PROJECT_CI_ACCESS	0
PROJECT_CI_INFO	0
PROJECT_DL_ACCESS	1
PROJECT_DL_INFO	1
PROJECT_DOC_ACCESS	1
PROJECT_DOC_INFO	4
PROJECT_GETTINGSTARTED_INFO	1
PROJECT_ITS_ACCESS	0
PROJECT_ITS_INFO	5
PROJECT_MLS_ACCESS	1
PROJECT_MLS_INFO	1
PROJECT_REL_VOL	5
PROJECT_SCM_ACCESS	1
PROJECT_SCM_INFO	1
SCM_AUTHORS	5
SCM_AUTHORS_1M	0
SCM_AUTHORS_1W	0
SCM_AUTHORS_1Y	0
SCM_BRANCHES	13
SCM_COMMITS	29
SCM_COMMITS_1M	0
SCM_COMMITS_1W	0
SCM_COMMITS_1Y	0
SCM_COMMITTERS	5
SCM_COMMITTERS_1M	0
SCM_COMMITTERS_1W	0
SCM_COMMITTERS_1Y	0
SCM_DIVERSITY_RATIO_1Y	0
SCM_MOD_LINES	9627
SCM_MOD_LINES_1M	0
SCM_MOD_LINES_1W	0
SCM_MOD_LINES_1Y	0

## SCM access

ID: PROJECT\_SCM\_ACCESS

Value: 1 ( 5 / 5 )

Description: Is the source\_repo info correctly filled in the PMI records? The project management infrastructure file holds information about one or more source repositories. This test checks that at least one source repository is defined, and accessible.

## **ITS access**

ID: PROJECT\_ITS\_ACCESS

Value: 0 ( 4 / 5 )

Description: Is the bugzilla info correctly filled in the PMI records? The project management infrastructure file holds information about one or more bugzilla instances. This test checks that at least one bugzilla instance is defined, with a product identifier, a create\_url to enter a new issue, and a query\_url to fetch all the issues for the project.

## **User ML Authors**

ID: MLS\_USR\_AUTHORS

Value: 346

Description: The total number of different identities found in the User mailing list. Having many different authors is a sign of diversity and activity. It makes the support more reliable (i.e. increased presence of people) and more complete (i.e. more eyes to solve a problem).

## **Number of red jobs**

ID: CI\_JOBS\_RED

Value:

Description: The number of red (failed) jobs on the CI engine. Red jobs in CI define failed builds.

## **Total complexity**

ID: SQ\_CPX

Value:

Description: It is the complexity calculated based on the number of paths through the code. Whenever the control flow of a function splits, the complexity counter gets incremented by one. Each function has a minimum complexity of 1. This calculation varies slightly by language because keywords and functionalities do. For more information on complexity for each language, see <https://docs.sonarqube.org/display/SONAR/Metrics+-+Complexity>.

## **Total number of failed builds**

ID: CI\_BUILDS\_FAILED

Value:

Description: The total number of builds with state failed for the project.

## **Stack Overflow Answers (5Y)**

ID: SO\_ANSWERS\_VOL\_5Y

Value:

Description: The number of answers to questions related to the project's tag posted on Stack Overflow during the last 5 years. Having many answers posted about the project indicates a strong interest from the community, and a good support. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## **Duplicated lines (%)**

ID: SQ\_DUPLICATED\_LINES\_DENSITY

Value:

Description: Density of duplication = Duplicated lines / Lines \* 100.

## **SCM Closed Pull requests**

ID: SCM\_PRS\_CLOSED

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) in the closed state in source code management repositories.

## **SCM Commits**

ID: SCM\_COMMITS

Value: 29

Description: Total number of commits in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date).

## **SCM committers**

ID: SCM\_COMMITTERS

Value: 5

Description: Total number of identities found as committers of commits in source code management repository. Source code management repositories are those considered as such in the project documentation. Commits in all branches are considered. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). An identity is considered as committer if it appears as such in the commit record.

## Number of forks

ID: PROJECT\_FORKS

Value:

Description: The number of forks for this project. More forks usually mean a greater activity.

## ITS authors last year

ID: ITS\_AUTHORS\_1Y

Value: 2

Description: Number of authors who created issues during last year. If today is 2017-02-01 then the range is from 2016-02-01 to 2017-02-01.

## Project committers one month

ID: PROJECT\_COMMITTERS\_1M

Value:

Description: Total number of identities found as committers of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## Has Readme

ID: SC\_HAS\_README

Value:

Description: The number of files considered as a readme, as detected by Scancode.

## Dependency check

ID: OSS\_DEP\_CHECK

Value: 1 ( 5 / 5 )

Description: Analysis of dependencies to identify old, vulnerable dependencies, as provided by tools like dependency-check or JFrog/Nexus. As an example, projects hosted at the Eclipse Foundation have a rigorous IP cleanliness procedure that makes sure that all dependencies meet defined requirements.

## Readme

ID: DOC\_README

Value:

Description: Is the documentation info correctly filled in the PMI records? The project management infrastructure file holds information about various documentation and manuals. This test checks the

number of doc-related entries defined in the PMI: build\_doc, documentation, documentation\_url, forums, gettingstarted\_url, mailing\_lists, website\_url, wiki\_url.

## **ITS issues updated last month**

ID: ITS\_UPDATED\_1M

Value: 1 ( 1 / 5 )

Description: Number of issues updated during last month. If today is 2017-02-01 then the range is from 2017-01-01 to 2017-02-01.

## **SCM Changed Lines**

ID: SCM\_MOD\_LINES

Value: 9627

Description: Total number of changed lines (added, removed, changed) in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date).

## **Project committers**

ID: PROJECT\_COMMITTERS

Value:

Description: Total number of identities found as committers of commits in source code management repository. Source code management repositories are those considered as such in the project documentation. Commits in all branches are considered. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). An identity is considered as committer if it appears as such in the commit record.

## **SCM Staled Open Pull requests one month**

ID: SCM\_PRS\_OPENED\_STALED\_1M

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that are in the opened state and have not been updated since one month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## **Number of failed jobs one week**

ID: CI\_JOBS\_FAILED\_1W

Value:

Description: The number of jobs that failed during last week on the CI engine.

## Special files

ID: SC\_SPECIAL\_FILES

Value:

Description: Number of legal, license, readmes, manifests, copyright and other files for key, top-level files. Key files are top- level codebase files such as COPYING, README and package manifests as reported by the `-classify` option 'is\_legal', 'is\_readme', 'is\_manifest' and 'is\_top\_level' flags.

## ITS Open issues

ID: ITS\_OPEN

Value: 328

Description: Number of issues with a state 'open' at the time of analysis.

## ITS authors last month

ID: ITS\_AUTHORS\_1M

Value: 1 ( 1 / 5 )

Description: Number of authors who created issues during last month. If today is 2017-02-01 then the range is from 2017-01-01 to 2017-02-01.

## Maintainability rating

ID: SQ\_SQALE\_RATING

Value:

Description: Rating given to your project related to the value of your Technical Debt Ratio. The default Maintainability Rating grid is: A=0-0.05, B=0.06-0.1, C=0.11-0.20, D=0.21-0.5, E=0.51-1. The Maintainability Rating scale can be alternately stated by saying that if the outstanding remediation cost depends on the time that has already gone into the application: A  $\leq 5\%$ , B between 6 to 10%, C between 11 to 20%, D between 21 to 50%, and anything over 50% is an E. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Maintainability>.

## Total number of Actors

ID: ITS\_PEOPLE

Value:

Description: The total number of people involved in issues tracking, includes authors (submitters), closers and updaters XXX.

## CI access

ID: PROJECT\_CI\_ACCESS

Value: 0 ( 4 / 5 )

Description: Is the continuous integration info correctly filled in the PMI records? The project management infrastructure file holds information about the location of CI services. This test checks the number of ci-related entries defined in the PMI.

## SCM Open Pull requests one week

ID: SCM\_PRS\_OPENED\_1W

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened within the last week in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## SCM committers one month

ID: SCM\_COMMITTERS\_1M

Value: 0

Description: Total number of identities found as committers of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## SCM Still Open Pull requests one month

ID: SCM\_PRS\_OPENED\_STILL\_1M

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened more than one month ago and are still in the opened state in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## Total number of builds

ID: CI\_BUILDS\_VOL

Value:

Description: The total number of builds listed in the GitLab Builds section. Builds are individual runs of jobs. See the GitLab documentation for more details.



## **Project committers one year**

ID: PROJECT\_COMMITTERS\_1Y

Value:

Description: Total number of identities found as committers of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## **User ML Authors one week**

ID: MLS\_USR\_AUTHORS\_1W

Value: 0

Description: The total number of different identities found in the User mailing list during last week. Having many different authors is a sign of diversity and activity. It makes the support more reliable (i.e. increased presence of people) and more complete (i.e. more eyes to solve a problem).

## **Number of minor issues**

ID: SQ\_VIOLATIONS\_MINOR

Value:

Description: The total number of issues (violations) found by SonarQube with a severity equal to MINOR. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Issues>.

## **SCM committers one week**

ID: SCM\_COMMITTERS\_1W

Value: 0

Description: Total number of identities found as committers of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **Number of lines of code**

ID: SQ\_NCLOC

Value:

Description: Number of physical lines that contain at least one character which is neither a whitespace or a tabulation or part of a comment. For Cobol, generated lines of code and pre-processing instructions (SKIP1, SKIP2, SKIP3, COPY, EJECT, REPLACE) are not counted as lines of code.

## **Project authors one month**

ID: PROJECT\_AUTHORS\_1M

Value:

Description: Total number of identities found as authors of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## **Project Commits**

ID: PROJECT\_COMMITS

Value:

Description: Total number of commits in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date).

## **ITS Authors**

ID: ITS\_AUTHORS

Value: 222

Description: Number of different authors who created issues during the lifetime of the project. A high number of authors shows diversity and improves the bus factor of the project.

## **Authors**

ID: SC\_AUTHORS\_VOL

Value:

Description: Number of authors detected in the code.

## **Stack Overflow Votes (5Y)**

ID: SO\_VOTES\_VOL\_5Y

Value:

Description: The total number of votes on questions related to the project's tag on Stack Overflow during the last 5 years. Having many votes on questions about the project indicates a strong interest from the community. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## **Project Commits one year**

ID: PROJECT\_COMMITS\_1Y

Value:

Description: Total number of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

### **Failed builds 1 month**

ID: CI\_BUILDS\_FAILED\_1M

Value:

Description: The number of builds with state failed for the project during the last month.

### **Total number of files analysed**

ID: SC\_FILES\_VOL

Value:

Description: Total number of files analysed (and documented) by Scancode. Metric is the number of files provided in the list of analysed files returned by Scancode.

### **Technical debt**

ID: SQ\_SQALE\_INDEX

Value:

Description: Effort to fix all maintainability issues. The measure is stored in minutes in the DB. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metrics+Definition>. Maintainability.

### **ITS issues created last year**

ID: ITS\_CREATED\_1Y

Value: 2

Description: Number of issues created during last year. If today is 2017-02-01 then the range is from 2016-02-01 to 2017-02-01.

### **Public documented API (%)**

ID: SQ\_PUBLIC\_API\_DOC\_DENSITY

Value:

Description: Density of public documented API = (Public API - Public undocumented API) / Public API \* 100

## Number of comment lines

ID: SQ\_COMMENT\_LINES

Value:

Description: Number of lines containing either comment or commented-out code. Non-significant comment lines (empty comment lines, comment lines containing only special characters, etc.) do not increase the number of comment lines. For Java, file headers are not counted as comment lines (as they usually define the license). Lines containing the following instructions are counted both as comments and lines of code: AUTHOR, INSTALLATION, DATE-COMPILED, DATE-WRITTEN, SECURITY. For more information on comments for each language, see <https://docs.sonarqube.org/display/SONAR/Metrics+-+Comment+lines>.

## User ML Threads one month

ID: MLS\_USR\_THREADS\_1M

Value: 0 ( 1 / 5 )

Description: The total number of threads (one question followed by zero or more answers) found in the User mailing list during last month. Having many threads shows the mailing list is active. It encourages people to participate, ask and answer questions.

## Ratio of green jobs

ID: CI\_JOBS\_GREEN\_RATIO

Value:

Description: The number of green (successful) jobs on the CI engine, divided by the total number of jobs. Green (or blue) jobs in CI define successful builds.

## MLS access

ID: PROJECT\_MLS\_ACCESS

Value: 1 ( 5 / 5 )

Description: Is the communication channel for the team info correctly filled in the PMI records? The project management infrastructure file holds information about the various communication means used by the project. This test checks the number of MLS-related entries defined in the PMI: mailing lists, forums, etc.

## SCM authors

ID: SCM\_AUTHORS

Value: 5

Description: Total number of identities found as authors of commits in source code management repository. Source code management repositories are those considered as such in the project documentation. Commits in all branches are considered. Date used for each commit is 'author date' (when there is a difference between author date and committer date). An identity is considered as author if it appears as such in the commit record (for systems logging several identities related to the commit, authoring identity will be considered).

## Number of functions

ID: SQ\_FUNCS

Value:

Description: Number of functions. Depending on the language, a function is either a function or a method or a paragraph. For Java, constructors are considered as methods and accessors are considered as methods if the sonar.squid.analyse.property.accessors property is set to false. For Cobol, it is the number of paragraphs.

## SCM authors one month

ID: SCM\_AUTHORS\_1M

Value: 0 ( 1 / 5 )

Description: Total number of identities found as authors of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## Number of files

ID: SQ\_FILES

Value:

Description: The total number of files analysed.

## Total number of pipelines

ID: CI\_PIPELINES\_VOL

Value:

Description: The total number of pipelines listed in the GitLab Pipelines section. A pipeline is a group of builds that get executed in stages. See the GitLab documentation for more details.

## Failed builds 1 week

ID: CI\_BUILDS\_FAILED\_1W

Value:

Description: The number of builds with state failed for the project during the last 7 days.

## Comment lines density

ID: SQ\_COMR

Value:

Description: Density of comment lines =  $\text{Comment lines} / (\text{Lines of code} + \text{Comment lines}) * 100$ . With such a formula, 50% means that the number of lines of code equals the number of comment lines and 100%

means that the file only contains comment lines For more information on comments for each language, see <https://docs.sonarqube.org/display/SONAR/Metrics+-+Comment+lines>.

## Number of stars

ID: PROJECT\_STARS

Value:

Description: The number of times people have starred this project. Users use stars to show their interest for a project, and more stars usually mean a greater visibility and interest.

## User ML Threads one week

ID: MLS\_USR\_THREADS\_1W

Value: 0

Description: The total number of threads (one question followed by zero or more answers) found in the User mailing list during last week. Having many threads shows the mailing list is active. It encourages people to participate, ask and answer questions.

## Inclusion initiative

ID: OSS\_INCLUSION

Value: 1 ( 5 / 5 )

Description: Is there a formal procedure to help people join and benefit from the community, whoever they are? As an example

## Package Tangle index

ID: SQ\_PACKAGES\_TANGLE\_IDX

Value:

Description: The Package tangle index, as defined in SonarQube.

## SCM diversity ratio

ID: SCM\_DIVERSITY\_RATIO\_1Y

Value: 0 ( 1 / 5 )

Description: The ration of commits divided by the number of authors, over a period of one year.

## SCM Commits one year

ID: SCM\_COMMITS\_1Y

Value: 0

Description: Total number of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used

for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## **CI information**

ID: PROJECT\_CI\_INFO

Value: 0 ( 4 / 5 )

Description: Is the continuous integration info correctly filled in the PMI records? The project management infrastructure file holds information about the location of CI services. This test checks the number of ci-related entries defined in the PMI.

## **MLS information**

ID: PROJECT\_MLS\_INFO

Value: 1 ( 4 / 5 )

Description: Is the communication channel for the team info correctly filled in the PMI records? The project management infrastructure file holds information about the various communication means used by the project. This test checks the number of MLS-related entries defined in the PMI: mailing lists, forums, etc.

## **Stack Overflow Questions (5Y)**

ID: SO\_QUESTIONS\_VOL\_5Y

Value:

Description: The number of questions related to the project's tag posted on Stack Overflow during the last 5 years. Having many questions posted about the project indicates a strong interest from the community. The list of questions associated to the tag can be browsed on the Stack Overflow web site.

## **DL access**

ID: PROJECT\_DL\_ACCESS

Value: 1 ( 5 / 5 )

Description: Is the access info (downloads, update sites..) correctly filled in the PMI records? The project management infrastructure file holds information about how to access binaries of the project. This test checks the number of access-related entries defined in the PMI: download\_url, downloads, update\_sites.

## **ITS Total issues**

ID: ITS\_ISSUES\_ALL

Value: 3220

Description: Number of issues registered in the database, whatever their state is.

## **ITS diversity ratio**

ID: ITS\_DIVERSITY\_RATIO\_1Y

Value: 1 ( 1 / 5 )

Description: The ration of bugs divided by the number of submitters, over a period of one year.

## **Test coverage**

ID: SQ\_COVERAGE

Value:

Description: Overall test coverage.

## **SCM information**

ID: PROJECT\_SCM\_INFO

Value: 1 ( 4 / 5 )

Description: Is the source\_repo info correctly filled in the PMI records? The project management infrastructure file holds information about one or more source repositories. This test checks that at least one source repository is defined, and accessible.

## **Escalation procedure**

ID: OSS\_ESCALATE

Value: 1 ( 5 / 5 )

Description: Is there an easy and visible way for anybody to report toxic behaviour or unfair practices? As an example, people willing to report unfair practices or bad behaviour within an Eclipse project can send an email at [codeofconduct@eclipse.org](mailto:codeofconduct@eclipse.org).

## **User ML Threads**

ID: MLS\_USR\_THREADS

Value: 1186

Description: The total number of threads (one question followed by zero or more answers) found in the User mailing list. Having many threads shows the mailing list is active. It encourages people to participate, ask and answer questions.

## **SCM Pull requests**

ID: SCM\_PRS

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) in source code management repositories. Source code management repositories are those considered as such in the project documentation.



## DL information

ID: PROJECT\_DL\_INFO

Value: 1 ( 5 / 5 )

Description: Is the access info (downloads, update sites..) correctly filled in the PMI records? The project management infrastructure file holds information about how to access binaries of the project. This test checks the number of access-related entries defined in the PMI: download\_url, downloads, update\_sites.

## Number of yellow jobs

ID: CI\_JOBS\_YELLOW

Value:

Description: The number of yellow (unstable) jobs on the CI engine. Yellow jobs in CI define unstable builds. According to CI's documentation, a build is unstable if it was built successfully and one or more publishers report it unstable. For example if the JUnit publisher is configured and a test fails then the build will be marked unstable.

## File complexity

ID: SQ\_CPX\_FILE\_IDX

Value:

Description: Average complexity by file. File complexity is computed using the Total complexity (SQ\_CPX, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Complexity>) divided by the number of files (SQ\_FILES).

## Code of Conduct

ID: SC\_HAS\_CODEOFCONDUCT

Value:

Description: The number of files considered as a code of conduct, as detected by Scancode. These can be any initiative and written document to regulate the behaviour of individuals As an example, the Eclipse Foundation enforces the following code of conduct: [https://www.eclipse.org/org/documents/Community\\_Code\\_of\\_Conduct.php](https://www.eclipse.org/org/documents/Community_Code_of_Conduct.php) .

## SCM Commits one month

ID: SCM\_COMMITS\_1M

Value: 0 ( 1 / 5 )

Description: Total number of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## **Governance documentation**

ID: DOC\_GOV

Value: 1 ( 5 / 5 )

Description: Is there a documentation explaining how decisions are made?

## **Number of copyrights**

ID: SC\_COPYRIGHTS\_VOL

Value:

Description: Number of copyrights detected by Scancode in the code base.

## **ITS Open issues (%)**

ID: ITS\_OPEN\_PERCENT

Value: 10

Description: Percentage of open issues compared to the overall number of issues registered in the system.

## **SCM Changed Lines one year**

ID: SCM\_MOD\_LINES\_1Y

Value: 0

Description: Total number of changed lines (added, removed, changed) in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## **Sqale Debt ratio**

ID: SQ\_SQALE\_DEBT\_RATIO

Value:

Description: The Technical Debt Ratio, as defined in Sqale. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Maintainability>.

## **Programming languages**

ID: SC\_PROGS\_VOL

Value:

Description: Number of programming languages detected in the code.

## Successful builds 1 week

ID: CI\_BUILDS\_SUCCESS\_1W

Value:

Description: The number of builds with state success for the project during the last 7 days.

## Number of blocker issues

ID: SQ\_VIOLATIONS\_BLOCKER

Value:

Description: The total number of issues (violations) found by SonarQube with a severity equal to BLOCKER. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Issues>.

## Has contributing

ID: SC\_HAS\_CONTRIBUTING

Value:

Description: The number of files considered as a contributing or development guide, as detected by Scancode.

## Docs information

ID: PROJECT\_DOC\_INFO

Value: 4 ( 5 / 5 )

Description: Is the documentation info correctly filled in the PMI records? The project management infrastructure file holds information about various documentation and manuals. This test checks the number of doc-related entries defined in the PMI: build\_doc, documentation, documentation\_url, forums, gettingstarted\_url, mailing\_lists, website\_url, wiki\_url.

## User ML diversity ratio

ID: MLS\_USR\_DIVERSITY\_RATIO\_1Y

Value: 0 ( 1 / 5 )

Description: The ration of posts divided by the number of authors, over a period of one year.

## SCM Branches

ID: SCM\_BRANCHES

Value: 13

Description: Number of branches as returned by `git branch --all`.

## **ITS Late issues**

ID: ITS\_LATE

Value:

Description: Number of issues with a past due date. It is considered good practice to keep this number low. Either fix it or maintain its due date.

## **User ML Posts one year**

ID: MLS\_USR\_POSTS\_1Y

Value: 0

Description: The total number of posts found in the User mailing list during last year. Having many posts shows the mailing list is active. It encourages people to participate, ask and answer questions.

## **SCM Merged Pull requests**

ID: SCM\_PRS\_MERGED

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) in the 'merged' state in source code management repositories.

## **Number of critical issues**

ID: SQ\_VIOLATIONS\_CRITICAL

Value:

Description: The total number of issues (violations) found by SonarQube with a severity equal to CRITICAL. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Issues>.

## **Project authors one year**

ID: PROJECT\_AUTHORS\_1Y

Value:

Description: Total number of identities found as authors of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## **Public API**

ID: SQ\_PUBLIC\_API

Value:

Description: Number of public Classes + number of public Functions + number of public Properties

## **SCM Commits one week**

ID: SCM\_COMMITS\_1W

Value: 0

Description: Total number of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **ITS Pending issues**

ID: ITS\_OPEN\_UNASSIGNED

Value: 0

Description: Number of issues in state open with no assignee (i.e. pending). It is considered to be good practice to keep this number low. In an active project, people would either work on the bug (i.e. assign it) or triage it (pass it to some other state or assigning it).

## **Number of copyright holders**

ID: SC\_HOLDERS\_VOL

Value:

Description: Number of copyright holders detected in the code.

## **Stack Overflow Askers (5Y)**

ID: SO\_ASKERS\_5Y

Value:

Description: The number of distinct people asking questions related to the project's tag posted on Stack Overflow during the last 5 years. Having many people ask questions about the project indicates a strong interest from the community, and a good support. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## **SCM Still Open Pull requests one year**

ID: SCM\_PRS\_OPENED\_STILL\_1Y

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened more than one year ago and are still in the opened state in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## SCM Changed Lines one month

ID: SCM\_MOD\_LINES\_1M

Value: 0 ( 1 / 5 )

Description: Total number of changed lines (added, removed, changed) in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## Contributing

ID: DOC\_CONTRIBUTING

Value:

Description: Is the documentation info correctly filled in the PMI records? The project management infrastructure file holds information about various documentation and manuals. This test checks the number of doc-related entries defined in the PMI: build\_doc, documentation, documentation\_url, forums, gettingstarted\_url, mailing\_lists, website\_url, wiki\_url.

## Line coverage

ID: SQ\_COVERAGE\_LINE

Value:

Description: Line test coverage.

## User ML Authors one month

ID: MLS\_USR\_AUTHORS\_1M

Value: 0 ( 1 / 5 )

Description: The total number of different identities found in the User mailing list during last month. Having many different authors is a sign of diversity and activity. It makes the support more reliable (i.e. increased presence of people) and more complete (i.e. more eyes to solve a problem).

## SCM committers one year

ID: SCM\_COMMITTERS\_1Y

Value: 0

Description: Total number of identities found as committers of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## Number of green jobs

ID: CI\_JOBS\_GREEN

Value:

Description: The number of green (successful) jobs on the CI engine. Green (or blue) jobs in CI define successful builds.

## Has licence

ID: SC\_HAS\_LICENCE

Value:

Description: The number of files considered as legal (i.e. licences), as detected by Scancode.

## ITS information

ID: PROJECT\_ITS\_INFO

Value: 5 ( 5 / 5 )

Description: Is the bugzilla info correctly filled in the PMI records? The project management infrastructure file holds information about one or more bugzilla instances. This test checks that at least one bugzilla instance is defined, with a product identifier, a create\_url to enter a new issue, and a query\_url to fetch all the issues for the project.

## User ML Posts one month

ID: MLS\_USR\_POSTS\_1M

Value: 0 ( 1 / 5 )

Description: The total number of posts found in the User mailing list during last month. Having many posts shows the mailing list is active. It encourages people to participate, ask and answer questions.

## User ML Posts one week

ID: MLS\_USR\_POSTS\_1W

Value: 0

Description: The total number of posts found in the User mailing list during last week. Having many posts shows the mailing list is active. It encourages people to participate, ask and answer questions.

## SCM authors one year

ID: SCM\_AUTHORS\_1Y

Value: 0

Description: Total number of identities found as authors of commits in source code management repositories dated during the last year. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date

and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## Number of major issues

ID: SQ\_VIOLATIONS\_MAJOR

Value:

Description: The total number of issues (violations) found by SonarQube with a severity equal to MAJOR. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Issues>.

## ITS issues updated last week

ID: ITS\_UPDATED\_1W

Value: 0

Description: Number of issues updated during last week. If today is Wed. 2017-02-01 then the range is from Wed. 2017-01-25 to Wed. 2017-02-01.

## Stack Overflow Views (5Y)

ID: SO\_VIEWS\_VOL\_5Y

Value:

Description: The total number of views for questions related to the project's tag on Stack Overflow during the last 5 years. Having many views on questions about the project indicates a strong interest from the community. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## Project Commits one week

ID: PROJECT\_COMMITS\_1W

Value:

Description: Total number of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## Access information

ID: PROJECT\_ACCESS\_INFO

Value: 1 ( 3 / 5 )

Description: Is the access info (downloads, update sites..) correctly filled in the PMI records? The project management infrastructure file holds information about how to access binaries of the project. This test checks the number of access-related entries defined in the PMI: download\_url, downloads, update\_sites.



## **ITS authors last week**

ID: ITS\_AUTHORS\_1W

Value: 0

Description: Number of authors who created issues during last week. If today is Wed. 2017-02-01 then the range is from Wed. 2017-01-25 to Wed. 2017-02-01.

## **Getting started**

ID: PROJECT\_GETTINGSTARTED\_INFO

Value: 1 ( 5 / 5 )

Description: Is the documentation info correctly filled in the PMI records? The project management infrastructure file holds information about various documentation and manuals. This test checks the number of doc-related entries defined in the PMI: build\_doc, documentation, documentation\_url, forums, gettingstarted\_url, mailing\_lists, website\_url, wiki\_url.

## **ITS Old open issues**

ID: ITS\_OPEN\_OLD

Value: 0

Description: Number of dead issues, i.e. issues that are in state open and have not been updated for a long time (one year). It is considered to be good practice to keep this number low because it impacts the confidence in the Issue Tracking System if there are too many open dead issues. Some projects close issues after some time of inactivity and re-open it if needed.

## **Board transparency**

ID: GOV\_BOARD\_PUBLIC

Value: 1 ( 5 / 5 )

Description: Are the minutes of the board publicly available?

## **Project committers one week**

ID: PROJECT\_COMMITTERS\_1W

Value:

Description: Total number of identities found as committers of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'committer date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **Stack Overflow Answer rate (5Y)**

ID: SO\_ANSWER\_RATE\_5Y

Value:

Description: The average number of answers per questions related to the project's tag on Stack Overflow during the last 5 years. Having many answers posted about the project indicates a strong interest from the community, and a good support. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## **Number of jobs**

ID: CI\_JOBS

Value:

Description: The total number of jobs defined on the CI engine.

## **SCM authors one week**

ID: SCM\_AUTHORS\_1W

Value: 0

Description: Total number of identities found as authors of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **SCM Changed Lines one week**

ID: SCM\_MOD\_LINES\_1W

Value: 0

Description: Total number of changed lines (added, removed, changed) in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **ITS issues created last week**

ID: ITS\_CREATED\_1W

Value: 0

Description: Number of issues created during last week. If today is Wed. 2017-02-01 then the range is from Wed. 2017-01-25 to Wed. 2017-02-01.

## Open issues

ID: PROJECT\_ISSUES\_OPEN

Value:

Description: The number of issues opened at the time of analysis on the GitLab project. This information is retrieved from GitLab itself, and may differ from numbers gathered from the actual issue tracking system used.

## Last activity

ID: PROJECT\_LAST\_ACTIVITY\_AT

Value:

Description: The date of last activity for this project. This includes any type of action: changes on issues, git commits or pushes, merge requests, comments... A project with a old last activity timestamp shows the project is dead (or in agony at least).

## ITS issues updated last year

ID: ITS\_UPDATED\_1Y

Value: 2

Description: Number of issues updated during last year. If today is 2017-02-01 then the range is from 2016-02-01 to 2017-02-01.

## User ML Authors one year

ID: MLS\_USR\_AUTHORS\_1Y

Value: 0

Description: The total number of different identities found in the User mailing list during last year. Having many different authors is a sign of diversity and activity. It makes the support more reliable (i.e. increased presence of people) and more complete (i.e. more eyes to solve a problem).

## ITS issues created last month

ID: ITS\_CREATED\_1M

Value: 1

Description: Number of issues created during last month. If today is 2017-02-01 then the range is from 2017-01-01 to 2017-02-01.

## Number of releases

ID: PMI\_REL\_VOL

Value:

Description: The number of releases recorded in the PMI. Milestones are retrieved from the PMI file and are counted whatever their target release is. Milestones are useful to assess the maturity of the release and improves predictability of the project's output, in terms of quality and time.

## **Stack Overflow Answer rate (1Y)**

ID: SO\_ANSWER\_RATE\_1Y

Value:

Description: The average number of answers per questions related to the project's tag on Stack Overflow during the last year. Having many answers posted about the project indicates a strong interest from the community, and a good support. The list of questions and their answers associated to the tag can be browsed on the Stack Overflow web site.

## **Commented code**

ID: SQ\_COM\_CODE

Value:

Description: Commented lines of code See more information about commented code on SonarQube doc web site. There is a well-documented debate on Stack Overflow as well. For more information on comments for each language, see <https://docs.sonarqube.org/display/SONAR/Metrics+-+Comment+lines>.

## **Number of info issues**

ID: SQ\_VIOLATIONS\_INFO

Value:

Description: The total number of issues (violations) found by SonarQube with a severity equal to INFO. For more information on maintainability metrics as computed by SonarQube, see <https://docs.sonarqube.org/display/SONAR/Metric+Definitions#MetricDefinitions-Issues>.

## **Project authors one week**

ID: PROJECT\_AUTHORS\_1W

Value:

Description: Total number of identities found as authors of commits in source code management repositories dated during the last week. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **Project Commits one month**

ID: PROJECT\_COMMITS\_1M

Value:

Description: Total number of commits in source code management repositories dated during the last month. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time

range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## **Doc access**

ID: PROJECT\_DOC\_ACCESS

Value: 1 ( 5 / 5 )

Description: Is the documentation info correctly filled in the PMI records? The project management infrastructure file holds information about various documentation and manuals. This test checks the number of doc-related entries defined in the PMI: build\_doc, documentation, documentation\_url, forums, gettingstarted\_url, mailing\_lists, website\_url, wiki\_url.

## **Branch coverage**

ID: SQ\_COVERAGE\_BRANCH

Value:

Description: Branch test coverage.

## **User ML Posts**

ID: MLS\_USR\_POSTS

Value: 1186

Description: The total number of posts found in the User mailing list. Having many posts shows the mailing list is active. It encourages people to participate, ask and answer questions.

## **Licences check**

ID: SC\_LIC\_CHECK

Value:

Description: Unwanted licences in the code, as provided by tools like ScanCode (or Black Duck or..). All licences found in the codebase that do not conform to a custom regexp are considered wrong. One should have as few as possible of them.

## **Total number of successful builds**

ID: CI\_BUILDS\_SUCCESS

Value:

Description: The total number of builds with state success for the project.

## Number of statements

ID: SQ\_STATEMENTS

Value:

Description: Number of statements. For Java, it is the number of statements as defined in the Java Language Specification but without block definitions. Statements counter gets incremented by one each time a following keyword is encountered: if, else, while, do, for, switch, break, continue, return, throw, synchronized, catch, finally.. Statements counter is not incremented by a class, method, field, annotation definition, package declaration and import declaration. For Cobol, a statement is one of move, if, accept, add, alter, call, cancel, close, compute, continue, delete, display, divide, entry, evaluate, exitProgram, goback, goto, initialize, inspect, merge, multiply, open, perform, read, release, return, rewrite, search, set, sort, start, stop, string, subtract, unstring, write, exec, ibmXmlParse, ibmXmlGenerate, readyReset, mfCommit, mfRollback.

## Project authors

ID: PROJECT\_AUTHORS

Value:

Description: Total number of identities found as authors of commits in source code management repository. Source code management repositories are those considered as such in the project documentation. Commits in all branches are considered. Date used for each commit is 'author date' (when there is a difference between author date and committer date). An identity is considered as author if it appears as such in the commit record (for systems logging several identities related to the commit, authoring identity will be considered).

## Number of releases

ID: PROJECT\_REL\_VOL

Value: 5 ( 4 / 5 )

Description: The number of releases recorded in the PMI. Milestones are retrieved from the PMI file and are counted whatever their target release is. Milestones are useful to assess the maturity of the release and improves predictability of the project's output, in terms of quality and time.

## Successful builds 1 month

ID: CI\_BUILDS\_SUCCESS\_1M

Value:

Description: The number of builds with state success for the project during the last month.

## User ML Threads one year

ID: MLS\_USR\_THREADS\_1Y

Value: 0

Description: The total number of threads (one question followed by zero or more answers) found in the User mailing list during last year. Having many threads shows the mailing list is active. It encourages people to participate, ask and answer questions.

## **SCM Open Pull requests one month**

ID: SCM\_PRS\_OPENED\_1M

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened within the last month in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one month period starting the day before the data retrieval (example: if retrieval is on Feb 3rd, period is from Jan 3rd to Feb 2nd, both included).

## **SCM Open Pull requests one year**

ID: SCM\_PRS\_OPENED\_1Y

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened within the last year in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one year period starting the day before the data retrieval (example: if retrieval is on Feb 3rd 2016, period is from Feb 3rd 2015 to Feb 3rd 2016, both included).

## **SCM Open Pull requests**

ID: SCM\_PRS\_OPENED

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) in the opened state in source code management repositories.

## **SCM Still Open Pull requests one week**

ID: SCM\_PRS\_OPENED\_STILL\_1W

Value:

Description: Total number of Pull Requests (PRs) or Merge Requests (MRs) that have been opened more than one week ago and are still in the opened state in source code management repositories. Source code management repositories are those considered as such in the project documentation. Date used for each commit is 'author date' (when there is a difference between author date and committer date). Time range is measured as a one week period starting the day before the data retrieval.

## **Automatically generated files**

ID: SC\_GENERATED\_VOL

Value:

Description: Number of files tagged as automatically generated, as detected by Scancode.

## Attributes

Mnemo	Value
QM_ACTIVITY	1.00
QM_AGG_COLLAB	5.00
QM_AGG_DECISIONS	5.00
QM_AGG_DOC	4.50
QM_AGG_ENGAGEMENT	1.00
QM_AGG_ETHICS	5.00
QM_AGG_IP	5.00
QM_AGG_OPENNESS	4.70
QM_AGG_OSS_CORE	4.80
QM_DIVERSITY	1.00
QM_DOC	4.20
QM_ECOSYSTEM	1.80
QM_PROCESS	3.80
QM_QUALITY	3.50
QM_REL_ENG	4.00
QM_SCM	2.00
QM_SUPPORT	1.00

### Activity

ID: QM\_ACTIVITY

Value: 1 / 5

Description: The activity of the project's ecosystem, as measured on the mailing lists and configuration management system. An active project will provide a lot of information on the mailing lists, so when an user encounters an issue she will quickly find the information she needs, and has more chances to get answers if she asks. Fixes and improvements are added regularly.

### Collaboration

ID: QM\_AGG\_COLLAB

Value: 5 / 5

Description: Are the required pieces of basic documentation available?

### Decisions

ID: QM\_AGG\_DECISIONS

Value: 5 / 5

Description: How public and open is the project's decision-making process?

### Information

ID: QM\_AGG\_DOC

Value: 4.5 / 5

Description: Is the project well documented?



## **Engagement**

ID: QM\_AGG\_ENGAGEMENT

Value: 1 / 5

Description: How the community is invited to participate, and how much is it heard when talking?

## **Ethics**

ID: QM\_AGG\_ETHICS

Value: 5 / 5

Description: Is the project compliant regarding OSS' ethics?

## **Intellectual Property Management**

ID: QM\_AGG\_IP

Value: 5 / 5

Description: How is intellectual property handled in the project? Is there an IP log? Are the project and the Eclipse Foundation safe regarding IP?

## **Openness**

ID: QM\_AGG\_OPENNESS

Value: 4.7 / 5

Description: Is the project open and transparent?

## **OSS Core**

ID: QM\_AGG\_OSS\_CORE

Value: 4.8 / 5

Description: The conformance of the project to the core Open-Source Software principles and good practices.

## **Diversity**

ID: QM\_DIVERSITY

Value: 1 / 5

Description: The diversity of the project's ecosystem, as measured on the mailing lists and configuration management system. If many different actors from different companies are involved in the project, then it improves its sustainability (by eliminating a single point of failure) and adaptability to different situations. Having developers and users with different contexts and perspectives on the project helps widening its scope and provide a more generic support.

## Documentation

ID: QM\_DOC

Value: 4.2 / 5

Description: The maturity of code. Good code is vital for maintenance and evolution. It will encourage people to contribute, lower the number of bugs, and make a better product for the end-user as well as for the maintainers.

## Ecosystem

ID: QM\_ECOSYSTEM

Value: 1.8 / 5

Description: The sustainability of the ecosystem evolving around the project. Sustainability is a key point for long term support. If there is a lot of activity, if people can get fast and complete answers, if many people from different companies contribute to the project, then it will have more chance to still be there in a few years, and to continue providing fixes and improvements. Ecosystem requirements have been discussed on the mailing list and during meetings, and have been further described on the Polarsys wiki.

## Process

ID: QM\_PROCESS

Value: 3.8 / 5

Description: The maturity of the process used to run the project. A sound process helps people to do things right and ease collaborative work. If the process is documented, has predictable output, helps enforcing good development practices, etc. then new comers will easily find the information to collaborate, test or change code, or participate in the community. A good process also helps producing a good product [[Ing2003](/documentation/references.html#Ing2003)] – although it is agreed that the process is not enough by itself. Process requirements have been discussed on the mailing list and during meetings, and have been further described on the Polarsys wiki. Some may also recognise CMMi Key Process Areas among the attributes.

## Eclipse Maturity

ID: QM\_QUALITY

Value: 3.5 / 5

Description: The overall Maturity of the project. In the context of embedded software, Maturity is usually associated with some kind of reliability (most bugs have been already found) and functionality of code, sustainability of the project (will it still deliver fixes and improvements in a few years), and process predictability. Maturity in the PolarSys context has been further described on the wiki, and is actually precisely defined by the decomposition of this quality model.

## Build and Release Management

ID: QM\_REL\_ENG

Value: 4 / 5

Description: Does the project apply best practices regarding Build and Release management?

## Configuration Management

ID: QM\_SCM

Value: 2 / 5

Description: The maturity of the project regarding access and usage of the configuration management system. Configuration management is an essential part of the collaboration in the project. Access to the source should be documented and facilitated for new comers to easily come in.

## Support

ID: QM\_SUPPORT

Value: 1 / 5

Description: The amount of knowledge provided when someone asks for support. Having many answers on a single question helps better understand how the product works in different conditions, and also provides help for people looking for a similar information later on, since mailing lists are archived and public.

## Git analysis

```
## Error in as.POSIXct.default(evolve$date): do not know how to convert 'evolve$date' to class "POSIXct"
```

The repository contains a total of 0 commits made by 5 authors. The first commit was made on the c074a946de3b785ddad88eea98612489ce9a9e52 and the last analysed commits was made on ae7add0aea9aba20a57ab96026907750f32e05a7.

During the last month, there has been 0 commits made by 0 authors.

## Weekly commits

```
## Error in FUN(X[[i]], ...): object 'evol_xts' not found
```

```
## Error in is.xts(x): object 'evol_xts' not found
```

```
## Error in ggplot(evol_xts_commits, aes(x = index(evol_xts_commits), y = commits)): object 'evol_xts_c
```

## Weekly authors

```
## Error in is.xts(x): object 'evol_xts' not found
```

```
## Error in ggplot(evol_xts_authors, aes(x = index(evol_xts_authors), y = authors)): object 'evol_xts_a
```