

**QualiPSO**  
*Quality Platform for Open Source Software*  
**IST- FP6-IP-034763**



**Working document wd 5.6.2**  
**Trustworthiness models for Open Source Software, version 3.0**

Luigi Lavazza  
Sandro Morasca  
Davide Taibi  
Davide Tosi

Due date of deliverable: 31/10/2010

Actual submission date: 31/01/2011

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is partially funded by EU under the grant of IST-FP6-034763.

## Change History

Version	Date	Status	Author (Partner)	Description
1.1	13/5/2009	Draft	INS	First draft, analysis of reliability only
1.2	30/6/2009	Draft	INS	Second draft, analysis of trustworthiness included
2.0	31/7/2009	Final	INS	Takes into account comments. Includes the analysis of relationships between objective software qualities.
3.0	31/01/2011	Final	INS	Includes the analysis of all the subjective qualities evaluated via questionnaires and all the objective measures.

## What is new

This document differs from the previous versions of this document, in that it reports the final and complete results of the analysis of data concerning OSS trustworthiness and properties that have been collected throughout the project.

The final quantitative models of the trustworthiness of OSS products as a function of objectively measurable characteristics are illustrated.

## EXECUTIVE SUMMARY

### Task 5.6.2 - Model building: objectives

In the Description of Work the task is defined as follows:

*The information collected in Task 5.6.1 will be analyzed to find out whether the factors identified were actually influential on the trustworthiness of the OSS products and artefacts. A variety of different statistical techniques will be used for data analysis, based on the specific independent and dependent variables involved and the objectives of the data analysis. Models will be built and lessons learned will be made explicit.*

The goal of the task is to build a quantitative model of software trustworthiness.

Such model can explain how the trustworthiness perceived by users depends on the actual (mostly objectively measurable) qualities of the OS software products.

The produced models also allow users to estimate how likely it is that a given OSS product will be considered trustworthy, given its measurable characteristics.

The results of the task are packaged as models and associated suggested interpretations, which can be easily understood and used by the practitioners in the European software industry.

### Method

The analysis presented here was carried out on the data collected by Task 5.6.1 and stored in the measurement repository. These data were analyzed to find relations that link the trustworthiness of the OSS products and artefacts with the objectively measurable characteristics and qualities of the SW.

To this end, we mainly used binary logistic regression. Binary logistic regression is a statistical technique that is used when the dependent is a dichotomy and the independent variables are of any type.

In our case the dependent variable represents the subjective evaluations of users concerning various qualities. Every quality evaluation is a dichotomy in the sense that users' evaluations are divided into two sets: the one containing positive evaluations and the one containing negative evaluations<sup>1</sup>.

The collected data were of two types: subjective evaluations by users, and objective measures. We tested all the possible correlations, i.e., all the combinations  $y=f(X)$ , where  $y$  is a subjective evaluation and  $X$  is an array of measures. Of these potential correlations, only the statistically significant ones were selected as valid models and are thus reported in this document.

---

<sup>1</sup> Evaluations are classified as positive or negative with respect to a threshold that was initially chosen in an arbitrary way and was then refined in order to get the most significant results.

## Results

We were able to find several statistically significant correlational models for the prediction of users' perceptions of a number of user-relevant qualities such as reliability, usability, portability, functionality, interoperability, security, performance, usefulness of the developer community, documentation quality, and overall trustworthiness. We call the ensemble of these correlational models MOSST (Model of Open Source Software Trustworthiness). Thus, we were able to find at least one quantitative model for predicting every subjective quality for which we collected data from users by means of questionnaires. The models in MOSST are built by using data on a number of objective measures on OSS products and projects, like modularity, defect density, size, number of downloads, to predict the user-relevant qualities. .

As we focused on Java and C++ products, we derived three classes of prediction models:

- models for Java programs' perceived qualities with the objective measures produced by a collection of QualiPSO tools, including MacXim (for both object-oriented measures and Elementary Code Assessment rule violations), StatSVN (for measures concerning software configurations and versioning activities) and OSLC (for measures concerning licensing);
- models for C++ programs, which were obtained considering the code measures produced by Kalibro;
- models for Java and C++ programs' perceived qualities with the objective measures produced by MacXim and Kalibro. Since java and C++ are subject to different measures, only the measures that are common to both languages were considered.

## Document Information

<b>IST Project Number</b>	FP6 – 034763			<b>Acronym</b>	QualiPSO
<b>Full title</b>	Quality Platform for Open Source Software				
<b>Project URL</b>	<a href="http://www.qualipso.org">http://www.qualipso.org</a>				
<b>Document URL</b>					
<b>EU Project officer</b>	Michel Lacroix				

<b>Deliverable</b>	<b>Number</b>	wd 5.6.2 QualiPSO	<b>Title</b>	Trustworthiness models for Open Source Software, version 3.0
<b>Work package</b>	<b>Number</b>	5.6	<b>Title</b>	Experimentation and model building
<b>Activity</b>	<b>Number</b>	A5	<b>Title</b>	Trustworthy Results

<b>Date of delivery</b>	<b>Contractual</b>	31/10/2010	<b>Actual</b>	31/01/2011
<b>Status</b>				final <input checked="" type="checkbox"/>
<b>Nature</b>	Report <input checked="" type="checkbox"/> Demonstrator <input type="checkbox"/> Other <input type="checkbox"/>			
<b>Dissemination Level</b>	Public <input checked="" type="checkbox"/> Consortium <input checked="" type="checkbox"/>			
<b>Abstract (for dissemination)</b>	<p>The quality of Open Source Software (OSS) is much debated, since OSS is used by a continuously growing number of people and organizations. However, the discussions on the quality of OSS are usually based on opinions, feelings, personal preferences, and sometimes even political ideas. This document reports on an analysis of the perceived quality of OSS and the objectively measurable factors that may influence it. Specifically, the users' and developers' evaluations of the trustworthiness of OSS products (and of related qualities, like reliability and functionality) were collected and correlated to objective code measures. The result is a set of quantitative models that account for the dependence of the perceivable qualities of OSS on objectively observable qualities of the code. The ensemble of these models is called MOSST (Model of Open Source Software Trustworthiness). MOSST can be used by: 1) end-users and developers that would like to reuse existing OSS products and components, to evaluate the level of trustworthiness, reliability, usability and several other qualities of these OSS products that can be expected based on objectively observable characteristics of OSS product and projects; 2) the developers of OSS products, who can set code quality targets based on the level of trustworthiness, reliability, usability and several other qualities they want to achieve.</p>			
<b>Keywords</b>	Trustworthiness models, data analysis, empirical study, logistic regression.			
<b>Authors (Partner)</b>	Luigi Lavazza (University of Insubria)			
<b>Responsible Author</b>	Luigi Lavazza	<b>Email</b>	luigi.lavazza@uninsubria.it	
	Partner	<b>Phone</b>	+39-0332-219830	

## TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY .....</b>	<b>3</b>
TASK 5.6.2 - MODEL BUILDING: OBJECTIVES .....	3
METHOD.....	3
RESULTS.....	4
<b>TABLE OF CONTENTS .....</b>	<b>6</b>
<b>1   INTRODUCTION .....</b>	<b>8</b>
1.1 ANALYSIS GOALS .....	8
1.2 TOOLS.....	9
1.3 ANALYSIS PROCEDURES.....	10
1.4 THE DATASET.....	12
1.5 HOW TO READ THE RESULTS .....	12
<b>2   ANALYSIS OF JAVA PRODUCTS .....</b>	<b>14</b>
2.1 RELIABILITY .....	14
2.2 USABILITY .....	17
2.3 PORTABILITY .....	28
2.4 HOW WELL ARE FUNCTIONAL REQUIREMENTS SATISFIED.....	36
2.5 INTEROPERABILITY .....	44
2.6 SECURITY.....	47
2.7 PERFORMANCE (IN TERMS OF SPEED).....	47
2.8 USEFULNESS OF THE PRODUCT DEVELOPER COMMUNITY.....	50
2.9 DOCUMENTATION QUALITY.....	52
2.10 TRUSTWORTHINESS WITH RESPECT TO NON OPEN SOURCE (CLOSED SOURCE) PRODUCTS.....	56
2.11 TRUSTWORTHINESS WITH RESPECT TO OPEN SOURCE PRODUCTS .....	58
2.12 TRUSTWORTHINESS .....	59
<b>3   ANALYSIS OF JAVA AND C++ PROGRAM QUALITIES VS. OO CHARACTERISTICS</b>	<b>67</b>
3.1 RELIABILITY .....	67
3.2 USABILITY .....	71
3.3 PORTABILITY .....	75
3.4 HOW WELL ARE FUNCTIONAL REQUIREMENTS SATISFIED.....	77
3.5 INTEROPERABILITY .....	82
3.6 SECURITY.....	82
3.7 PERFORMANCE (IN TERMS OF SPEED) .....	83
3.8 USEFULNESS OF THE PRODUCT DEVELOPER COMMUNITY.....	84
3.9 DOCUMENTATION QUALITY.....	85
3.10 TRUSTWORTHINESS WITH RESPECT TO NON OPEN SOURCE (CLOSED SOURCE) PRODUCTS.....	90
3.11 TRUSTWORTHINESS WITH RESPECT TO OPEN SOURCE PRODUCTS .....	95
3.12 TRUSTWORTHINESS.....	96
<b>4   ANALYSIS OF C++ PRODUCTS VS. SPECIFIC CODE MEASURES .....</b>	<b>100</b>
4.1 RELIABILITY .....	100
4.2 USABILITY .....	101
4.3 PORTABILITY .....	102
4.4 HOW WELL ARE FUNCTIONAL REQUIREMENTS SATISFIED.....	103
4.5 INTEROPERABILITY .....	105
4.6 SECURITY.....	105
4.7 PERFORMANCE (IN TERMS OF SPEED) .....	106
4.8 USEFULNESS OF THE PRODUCT DEVELOPER COMMUNITY .....	106
4.9 DOCUMENTATION QUALITY.....	107
4.10 TRUSTWORTHINESS WITH RESPECT TO NON OPEN SOURCE (CLOSED SOURCE) PRODUCTS ...	108

---

4.11	TRUSTWORTHINESS WITH RESPECT TO OPEN SOURCE PRODUCTS .....	110
4.12	TRUSTWORTHINESS.....	113
<b>5</b>	<b>CONCLUSIONS .....</b>	<b>114</b>
5.1	USAGE OF THE RESULTS .....	114
5.2	IMPROVEMENTS OVER THE STATE OF THE ART .....	116
<b>6</b>	<b>REFERENCES.....</b>	<b>118</b>
<b>7</b>	<b>APPENDIXES.....</b>	<b>121</b>
7.1	THE LOGISTIC FUNCTION.....	121
7.2	ELEMENTARY CODE ASSESSMENT RULES .....	122
7.3	MEASURE NAMES AND DEFINITIONS .....	124

## 1 INTRODUCTION

### 1.1 Analysis goals

The data collection activities performed in WP5.6.1 provided a data concerning 44 OSS products, 22 written in Java and 22 written in C++.

The Java projects that were considered in the experimentation are the following:

- Ant
- Checkstyle
- Eclipse
- Findbugs
- Hibernate
- HttpUnit
- Jack.Commons IO
- Jasper Reports
- JBoss
- JFreeChart
- JMeter
- Log4J
- PMD
- Saxon
- Servicemix
- Spring Framework
- Struts
- Tapestry
- Weka
- Xalan
- Xerces

The C++ projects that were considered in the experimentation are the following:

- Axis
- BusyBox
- CVS
- CygWin
- DDD
- GDB
- Gnu C Library
- Gnu GCC
- Lib XML
- Linux Kernel
- Mono
- MySQL
- OpenLDAP
- Open Pegasus
- Open SSL
- Perl

- PostgreSQL
- SpiderMonkey
- SQLite
- Subversion
- TCL/Tk
- TPTP
- Velocity

For each of these products mentioned above we have:

- Evaluations concerning the qualities of the products that are perceived subjectively by the users. These evaluations are expressed in an ordinal scale (form 0 = totally unsatisfactory to 6 = fantastic).
- Measures that capture in an objective way the characteristics of the software.

The subjective qualities that were investigated are the following:

- Trustworthiness
- Reliability
- Usability
- Portability
- Functionality
- Interoperability
- Security
- Speed
- Community
- Documentation Quality
- Overall trustworthiness with respect to OSS Competitors
- Overall trustworthiness with respect to CSS Competitors

The goal of the analysis reported in the rest of the document is to evaluate – through statistical methods – whether there are relations that link the subjective perception of OSS product qualities with objective measures of the software. For instance, the analysis includes verifying the existence of relations of the trustworthiness with characteristics like software size, complexity and modularity.

## 1.2 Tools

The analysis of the data provided by Task 5.6.1 was carried out using appropriate tools. In particular, statistical tools were needed in order to perform the necessary computations and verify whether the factors identified in WP5.3 are actually influential on the trustworthiness of the OSS products and artefacts.

The analysis performed in WP5.6 is characterized by quite classical statistical techniques. Therefore we did not look for a particularly sophisticated tool; rather we sought a tool that

- a) Can be integrated at the data level with the measurement repository. In fact, we need to extract the required data from the repository and feed them to the analysis tools in a simple and efficient way.

- b) Provides all the required statistical tools, in particular logistic regression.
- c) Is programmable, in order to let us define the statistic procedures to be applied repetitively.

On the basis of these requirements, it was decided to use R [12], the tool that was already successfully used in WP5.1. R is a GPL-licensed language and environment for statistical computing and graphics that is reasonably easy to use and comes with a huge repository packages for analysis, database integration, etc. (see the Comprehensive R Archive Network at <http://cran.r-project.org/>).

### 1.3 Analysis procedures

As summarized in section 1.1 above, and described in the working documents and deliverables of WP5.6, the main goal of this task is to correlate subjective user evaluations with subjective software measures.

All subjective evaluations are expressed by each user in an ordinal scale with grades from zero to six.

Since we have interviewed several users about a given quality of a given OSS product, we need to reduce this amount of data to a single number that can be effectively treated. To this end, we establish a threshold that represents an acceptable quality level and then partition the population of the respondents into two datasets: one containing the users that rated the product below the threshold, and one containing the users that rated the product above the threshold.

More formally, given an OSS product P and a quality Q, we start from the multiset<sup>2</sup> of evaluations  $E = \{e_i\}$ , where  $i \in [1..N]$  indicates the i-th user, N is the number of interviewed users, and  $e_i$  is the rating of the quality Q of product P according to the i-th user.

By establishing a threshold T, we can partition E into  $E_s$  and  $E_u$ , the multisets of satisfied and unsatisfied users, respectively:

$$E_s = \{x \mid x \in E \wedge x > T\}$$

$$E_u = \{x \mid x \in E \wedge x \leq T\}$$

Now, we are not interested in distinguishing user identities; rather, we are interested in how many users are satisfied and how many are unsatisfied. To this end, we consider the pairs  $\langle |E_s|, |E_u| \rangle$  of the cardinalities of  $E_s$  and  $E_u$ .

Of course, we have a pair  $\langle |E_s|, |E_u| \rangle$  for every subjective quality defined in the GQM plan [8][9][10] and actually collected [15][16]. For every quality we have thus a pair, which can be interpreted as a percentage of satisfaction ( $|E_s|/(|E_u|+|E_s|) = |E_s|/N$ ).

---

<sup>2</sup> A multiset or bag is a set with repetitions. This clearly accounts for the fact that multiple users can assign a given quality of a given product the same grade.

Since we performed the evaluation of several OSS products, we actually have a vector of pairs and percentages:

$V_e = \langle P_j \rangle$ , where  $P_j$  is the pair  $\langle |E_s|, |E_u| \rangle$  concerning the  $j$ -th OSS product.

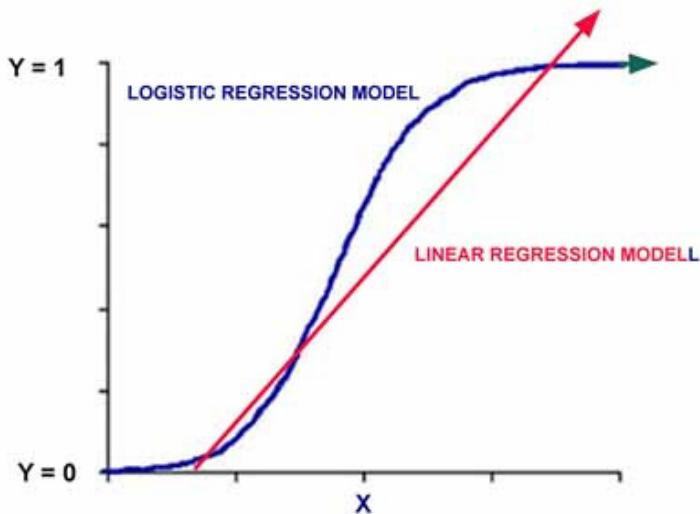
Actually we have not just one vector, but several: one for each investigated quality. Similarly, we have a vector for each objective quality that has been measured.

The analysis consists in correlating a vector of subjective evaluations with one or more vectors of objective measures, in order to evaluate to what extent the qualities perceived by the users depend on the internal, objectively measurable qualities. For instance, we correlated Trustworthiness to the measures of size and complexity, as well as reliability to the measures of modularity.

The analysis was based on binary logistic regression. Binary (or binomial) logistic regression is a form of regression which is used when the dependent is a dichotomy and the independents are of any type.

Logistic regression has many analogies to linear regression. Unlike the latter, however, logistic regression does not assume linearity of relationship between the independent variables and the dependent, does not require normally distributed variables, does not assume homoscedasticity, and in general has less stringent requirements. It does, however, require that observations be independent and that the independent variables be linearly related to the logit of the dependent.

The logistic curve, illustrated in Figure 1, is better for modeling binary dependent variables coded 0 or 1 because it comes closer to hugging the  $y=0$  and  $y=1$  points on the  $y$  axis. Even more, the logistic function is bounded by 0 and 1, whereas the linear regression function may predict values above 1 and below 0.



**Figure 1. Logistic vs. linear regression curves.**

Additional information on the linear regression is reported in the appendix.

We tested all the possible correlations, i.e., all the combinations  $y=f(X)$ , where  $y$  is a subjective evaluation and  $X$  is an array of measures. Of these potential correlations, only the statistically significant ones were selected as valid models and are thus reported in this document.

It must be noted that using multiple independent variables leads to the danger of ‘overfitting’. Since the number of OSS products we analyzed is not that big, we limited the number of independent variables to a maximum of 4 in every correlation. However, the user is invited to use models that involve as few independent variables as possible.

## 1.4 The dataset

The analysis reported here is based on the subjective evaluations about OSS collected throughout the QualiPSO project.

For every subjective evaluation we used the numbers of satisfied and not satisfied users. The threshold is 4, i.e., users who ranked a product  $> 4$  were counted as satisfied, while those who ranked it  $\leq 4$  were counted as not satisfied.

For each product we have a variable number of users’ evaluations, since most popular products like Eclipse or MySQL tend to be evaluated by more users than products –like Weka or Tapestry– that are of interest to a smaller, often specialized, set of users.

Moreover, some users reported a low familiarity with the products in the questionnaires.

Accordingly, we had to select the data to be used for the analysis: only products for which no less than six subjective evaluations expressed by users having a good familiarity with product were considered in the analysis. As a consequence, every analysis involved 16 to 18 products, depending on the specific quality being considered (users were free to express opinions only on a subset of the products’ qualities).

## 1.5 How to read the results

Every correlation found is illustrated by means of a set of results from the statistical analysis as illustrated in Figure 2.

```

Reliability vs. LCOM , Num. interfaces
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.3718709280 0.3043314130 4.507819 6.549741e-06
x1          -0.0007125015 0.0002546989 -2.797426 5.151154e-03
x2          -0.0014663310 0.0005899464 -2.485533 1.293577e-02
R2log = 0.9215926
Excluded as outliers: Eclipse HttpUnit Ant Struts (4/16)
MMRE = 19.73220
Pred(25) = 75
Error range = [-15.67529 .. 131.7843]

```

**Figure 2. Data about a correlation.**

The first line indicates the correlation being reported: the correlation reported in Figure 2 concerns reliability vs. LCOM (the lack of cohesion between methods) and the number of interfaces.

The following lines reports in the first column the values of the coefficients of the correlation (where x1 and x2 indicate the independent variables as reported in the title, thus x1 = LCOM and x2 = Num. interfaces). Therefore,

$$z = 1.371870928 - 0.0007125015 x_1 - 0.001466331 x_2 \text{ and } \text{Reliability}(z) = \frac{1}{1+e^{-z}}.$$

The column ‘Pr(>|z|)’ indicates the significance of the coefficients: all the values, except the one concerning the intercept, should be < 0.05. We adopt 0.05 as a threshold, as usually done in empirical software engineering.

R2log is the value of  $R^2_{\log}$ , a measure of goodness of fit defined in [19] that ranges between 0 and 1: the higher  $R^2_{\log}$ , the higher the effect of the model’s explanatory variables, the more accurate the model.

The next line reports the products that were excluded from the analysis, having been considered outliers. In our example, 4 products out of 17 (namely, Eclipse, HttpUnit, Ant and Struts) were excluded as outliers.

The last three lines give some indication on the precision of the fitting. MMRE (Mean Magnitude Relative Error) indicates what is the average absolute percent error: values below 25% are generally considered good. Pred(25) indicates how many products are within  $\pm 25\%$  error with respect to the regression line. Finally the error range indicates the minimum and maximum distance between observed values and estimated ones (always in percentage terms).

## 2 ANALYSIS OF JAVA PRODUCTS

The subjectively evaluated qualities were correlated with:

- The object-oriented code measures provided by MacXim (CBO, LCOM, Comment lines, Comment lines per class, eLOC per class, eLOC, McCabe, NOC, Num. abstract classes, Num. attributes per class, Num. classes, Num. interfaces, Num. interfaces per class, Num. methods, Num. methods per class, Num. methods per interface, Num. packages, Num. parameters per method, Num. public methods, RFC).
- The measures concerning software configurations and versioning activities provided by StatSVN (number of developers, number of commits, average LOC added per year, average LOC deleted per year, average LOC changed per year, average major releases per year, average minor releases per year, average file size, average files added per year, average files removed per year, average number of revisions per file).
- The measures provided by OSLC (Copyrighted Files, Copyright Holders, Distinct Licenses, Global Conflicts, Licensed Files, Reference Conflicts, Uncertain Licensed Files, Unlicensed Files).
- The measures provided by ECA (Elementary Code Analysis) evaluation tools incorporated in MacXim. These measures were not used individually, in order to keep reasonably small the amount of measures to be analyzed. We summed the rule violations according to the expected severity of the violation, thus obtaining three aggregated measures: the total number of ECA rule violations, the number of critical ECA rule violations, the number of very critical ECA rule violations.

Here follow the results of the analysis, grouped per quality.

### 2.1 Reliability

The significant models found for OSS products Reliability are reported below.

```
=====
Reliability vs. avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0393186568 0.1471872020 0.2671337 0.78936625
x1          0.0008904967 0.0004195495 2.1225070 0.03379519
R2log = 0.9119883
Excluded as outliers: Eclipse Log4J (2/13)
MMRE = 13.76082
Pred(25) = 84.61538
Error range = [-17.97649 .. 53.10874]
=====
Reliability vs. ECACritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.795064318 0.186064270 4.273063 1.928061e-05
x1          -0.003509867 0.001722697 -2.037425 4.160747e-02
R2log = 0.9082172
Excluded as outliers: Hibernate Eclipse HttpUnit (3/14)
MMRE = 19.63718
Pred(25) = 71.42857
Error range = [-25.75427 .. 103.2572]
```

```
=====
Reliability vs. LCOM , Num. interfaces
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.3718709280 0.3043314130 4.507819 6.549741e-06
x1          -0.0007125015 0.0002546989 -2.797426 5.151154e-03
x2          -0.0014663310 0.0005899464 -2.485533 1.293577e-02
R2log = 0.9215926
Excluded as outliers: Eclipse HttpUnit Ant Struts (4/16)
MMRE = 19.73220
Pred(25) = 75
Error range = [-15.67529 .. 131.7843]
=====

Reliability vs. LCOM , UncertainLicensedFiles
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.2245487433 0.2881379256 4.249870 2.138946e-05
x1          -0.0007735773 0.0002804339 -2.758501 5.806704e-03
x2          -0.0002911288 0.0001478451 -1.969147 4.893623e-02
R2log = 0.9177043
Excluded as outliers: Eclipse HttpUnit Struts (3/15)
MMRE = 21.12243
Pred(25) = 86.66667
Error range = [-20.37880 .. 126.8356]
=====

Reliability vs. LCOM , ECACritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.3058266753 0.3118370372 4.187529 2.820078e-05
x1          -0.0006565385 0.0002513258 -2.612300 8.993523e-03
x2          -0.0013650272 0.0006307700 -2.164065 3.045936e-02
R2log = 0.9229959
Excluded as outliers: Eclipse HttpUnit Struts (3/14)
MMRE = 20.45398
Pred(25) = 85.71429
Error range = [-18.19397 .. 129.8031]
=====

Reliability vs. LCOM , ECAveryCritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.2480367237 0.2959900934 4.216481 2.481437e-05
x1          -0.0006015196 0.0002423174 -2.482363 1.305143e-02
x2          -0.0023937469 0.0011465181 -2.087841 3.681222e-02
R2log = 0.9223916
Excluded as outliers: Eclipse HttpUnit Struts (3/14)
MMRE = 20.23879
Pred(25) = 85.71429
Error range = [-20.74960 .. 127.2924]
=====

Reliability vs. eLOC per class , Num. packages
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.92757228 0.493480064 3.906079 9.380573e-05
x1          -0.01267220 0.005318638 -2.382603 1.719070e-02
x2          -0.00937155 0.004132174 -2.267946 2.333247e-02
R2log = 0.928269
Excluded as outliers: Hibernate Httpunit Eclipse Struts weka (5/16)
MMRE = 26.84691
Pred(25) = 68.75
Error range = [-95.14194 .. 132.7176]
=====
```

```

Reliability vs. number_of_commits , avg_loc_del_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept)  5.743092e-02 1.928602e-01  0.2977852 0.76586712
x1          2.162240e-04 9.234622e-05  2.3414496 0.01920902
x2         -1.042659e-05 5.286799e-06 -1.9721933 0.04858755
R2log = 0.9090994
Excluded as outliers: Eclipse Hibernate Findbugs Xerces (4/13)
MMRE = 25.07036
Pred(25) = 61.53846
Error range = [-65.08979 .. 64.27492]
=====
Reliability vs. number_of_commits, avg_number_of_revisions_per_file
              Estimate Std. Error   z value Pr(>|z|)
(Intercept)  0.8920003252 2.535470e-01  3.518087 0.000434670
x1          0.0001288062 5.530034e-05  2.329212 0.019847837
x2         -0.1339535226 4.821182e-02 -2.778437 0.005462102
R2log = 0.9204644
Excluded as outliers: Eclipse (1/13)
MMRE = 10.67684
Pred(25) = 92.3077
Error range = [-14.19207 .. 33.96322]
=====
```

The first model seems to indicate that reliability is higher for actively maintained products (many files removed per year).

The second model was quite expected: it says that reliability of products is inversely proportional to the number of most critical ECA rules violated. In other words, the higher the quality of the code, the more reliable the product.

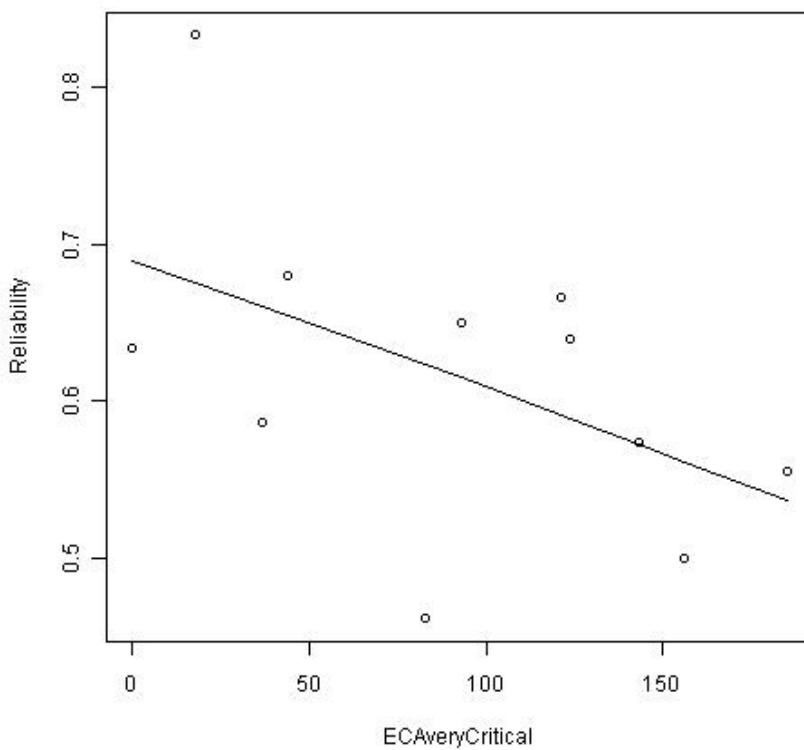
The third model seems to indicate that reliability is somehow proportional to modularity (good cohesion and a small number of interfaces).

The fourth model indicates that beside cohesion, the lack of files whose license is uncertain is also beneficial for reliability.

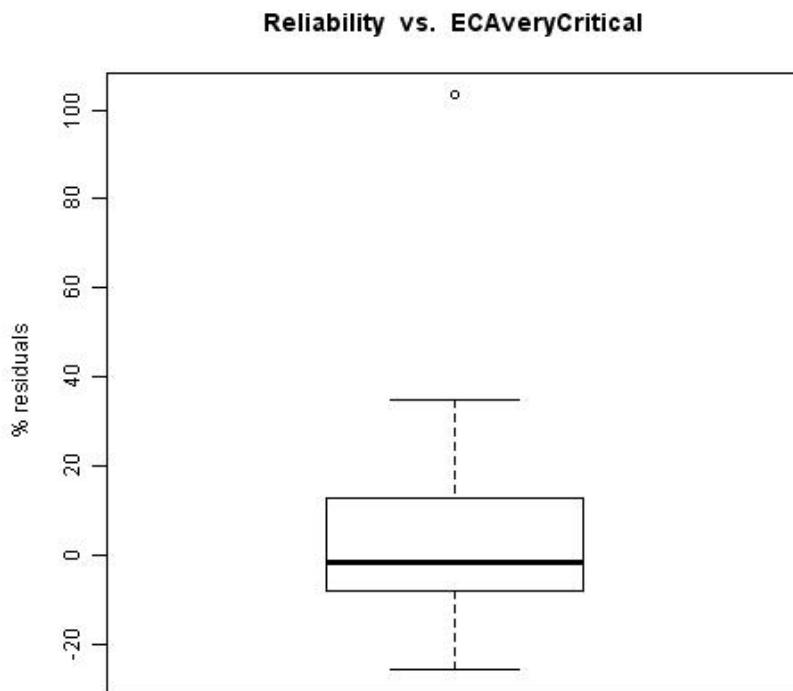
The fifth and sixth models stress the importance of cohesion and code quality (already found in previous models).

The seventh model indicates that small classes and small packages increase the reliability. This is also a not surprising result, since small classes and packages are more easily tested, thus contribute positively to reliability.

The last two models seem to indicate that reliability is higher for actively maintained products (high number of commits) that are fairly stable (few LOCs are deleted and most files do not need revisions).



**Figure 3. Reliability vs. Total ECA rules violated: plot of the regression line.**



**Figure 4. Reliability vs. Total ECA rules violated: boxplot of relative residuals.**

## 2.2 Usability

The significant models found for OSS products Usability are reported below.

```
=====
Usability vs. Num. classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.4732851621 0.2854750720 -1.657886 0.09734045
x1          0.0004609994 0.0001951329  2.362490 0.01815265
R2log = 0.9392631
Excluded as outliers: Log4J Hibernate Checkstyle Struts (4 / 16)
MMRE = 23.29477
Pred(25) = 68.75
Error range = [ -41.37845 .. 77.36136 ]
=====
Usability vs. Num. methods per interface
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2797823 0.2010649 -1.391503 0.16407303
x1          0.2686397 0.1238888  2.168394 0.03012869
R2log = 0.9025593
Excluded as outliers: Log4J Eclipse JBoss (3 / 14)
MMRE = 17.37485
Pred(25) = 78.57143
Error range = [ -37.29551 .. 54.66447 ]
=====
Usability vs. Num. parameters per method
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.046185 1.018763 -2.008499 0.04459027
x1          2.480265 1.128161  2.198503 0.02791327
R2log = 0.908859
Excluded as outliers: JFreeChart Eclipse Log4J Xerces (4 / 16)
MMRE = 21.25926
Pred(25) = 75
Error range = [ -35.22808 .. 91.39669 ]
=====
Usability vs. loc_total
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.061352e-01 1.940042e-01 4.670699 3.001768e-06
x1          -3.446864e-06 1.052909e-06 -3.273659 1.061648e-03
R2log = 0.9198953
Excluded as outliers: Hibernate Httpunit Eclipse Xalan (4 / 13)
MMRE = 21.82275
Pred(25) = 69.23077
Error range = [ -41.46637 .. 98.83877 ]
=====
Usability vs. number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.46917258 0.27110062 -1.730622 0.08351928
x1          0.02933536 0.01256109  2.335416 0.01952172
R2log = 0.9096772
Excluded as outliers: Log4J Saxon JBoss Eclipse (4 / 13)
MMRE = 15.42022
Pred(25) = 84.61538
Error range = [ -52.42674 .. 19.70674 ]
=====
Usability vs. avg_file_size
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.634744779 0.222396295 2.854116 0.00431568
x1          -0.003339909 0.001691749 -1.974235 0.04835502
```

```

R2log = 0.9105853
Excluded as outliers: Eclipse Log4J HttpUnit ( 3 / 13 )
MMRE = 17.23427
Pred(25) = 84.61538
Error range = [ -21.67194 .. 80.25315 ]
=====
Usability vs. avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.448839333 0.2268746116 -1.978359 0.0478882698
x1          0.002066045 0.0006118604  3.376661 0.0007337129
R2log = 0.9067334
Excluded as outliers: Log4J Eclipse Hibernate Struts ( 4 / 13 )
MMRE = 24.86088
Pred(25) = 69.23077
Error range = [ -23.27171 .. 73.4178 ]
=====
Usability vs. avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.220728746 0.1731528182 -1.274763 0.202393264
x1          0.001288239 0.0004524008  2.847562 0.004405548
R2log = 0.9005475
Excluded as outliers: Eclipse Log4J JBoss ( 3 / 13 )
MMRE = 15.10629
Pred(25) = 100
Error range = [ -24.30732 .. 24.82526 ]
=====
Usability vs. eLOC per class , Num. interfaces per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.90809830 0.53248871 -1.705385 0.0881226291
x1          0.04728134 0.01342736  3.521268 0.0004294887
x2          -4.61696038 1.27936320 -3.608796 0.0003076218
R2log = 0.950309
Excluded as outliers: Ant JFreeChart Struts Xerces JMeter ( 5 / 16 )
MMRE = 27.78419
Pred(25) = 62.5
Error range = [ -90.48443 .. 61.84469 ]
=====
Usability vs. eLOC per class , Num. packages
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.430484906 0.4700161464  3.043480 0.002338587
x1          -0.012321271 0.0051921097 -2.373076 0.017640630
x2          -0.001238334 0.0006129948 -2.020138 0.043369071
R2log = 0.9430268
Excluded as outliers: Log4J HttpUnit Struts Ant ( 4 / 16 )
MMRE = 18.70431
Pred(25) = 75
Error range = [ -33.35146 .. 95.6641 ]
=====
Usability vs. eLOC per class , avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.22716e-01 4.449350e-01 -1.624318 0.104307921
x1          1.94868e-02 7.053179e-03  2.762840 0.005730091
x2          -2.27246e-06 1.130310e-06 -2.010474 0.044381015
R2log = 0.9467294
Excluded as outliers: JFreeChart Xerces JBoss ( 3 / 13 )
MMRE = 24.17765

```

```

Pred(25) = 69.23077
Error range = [ -18.53226 .. 107.2472 ]
=====
Usability vs. eLOC per class , avg_major_release_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.84478383 0.442275103 -1.910087 0.05612204
x1          0.01615446 0.006697202 2.412121 0.01586003
x2          0.60178525 0.245006191 2.456204 0.01404133
R2log = 0.9489831
Excluded as outliers: JFreeChart Xerces JBoss ( 3 / 13 )
MMRE = 22.81104
Pred(25) = 69.23077
Error range = [ -18.49922 .. 103.7485 ]
=====
Usability vs. eLOC per class , avg_files_rem_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.200114944 0.5074247052 -2.365109 0.01802475
x1          0.017699532 0.0071513728 2.474984 0.01332422
x2          0.001206152 0.0004872897 2.475227 0.01331516
R2log = 0.9205234
Excluded as outliers: Eclipse JFreeChart Xerces Log4J ( 4 / 13 )
MMRE = 18.39188
Pred(25) = 76.92308
Error range = [ -17.91773 .. 81.11372 ]
=====
Usability vs. eLOC per class , ECACritical
    Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.666516648 0.527227089 -1.264193 0.20616086
x1          0.025085968 0.010042191 2.498057 0.01248760
x2          -0.006714106 0.002612079 -2.570407 0.01015790
R2log = 0.9487568
Excluded as outliers: JFreeChart Hibernate Xerces HttpUnit ( 4 / 14 )
)
MMRE = 25.90199
Pred(25) = 64.28571
Error range = [ -57.96616 .. 114.2122 ]
=====
Usability vs. eLOC , avg_files_rem_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.107361e-01 2.918627e-01 -2.435173 0.014884665
x1          8.674047e-06 3.854499e-06 2.250370 0.024425483
x2          1.692661e-03 5.222595e-04 3.241034 0.001190968
R2log = 0.8975387
Excluded as outliers: Eclipse Hibernate Log4J JBoss ( 4 / 13 )
MMRE = 16.98671
Pred(25) = 76.92308
Error range = [ -36.00127 .. 50.9857 ]
=====
Usability vs. NOC , number_of_developers
    Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.56418872 0.51173381 3.056645 0.002238290
x1          -0.68556564 0.30123153 -2.275876 0.022853426
x2          -0.02362911 0.01116890 -2.115617 0.034377434
R2log = 0.9513555
Excluded as outliers: HttpUnit JFreeChart Struts ( 3 / 13 )
MMRE = 25.68527

```

```

Pred(25) = 76.92308
Error range = [ -20.09223 .. 108.9933 ]
=====
Usability vs. NOC , avg_number_of_revisions_per_file
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.0689382 0.36806575 2.904204 0.003681878
x1          0.6053643 0.28733275 2.106841 0.035131397
x2         -0.1818691 0.05541145 -3.282158 0.001030159
R2log = 0.9112076
Excluded as outliers: Findbugs Eclipse Hibernate ( 3 / 13 )
MMRE = 18.69598
Pred(25) = 61.53846
Error range = [ -58.42445 .. 46.70719 ]
=====
Usability vs. Num. attributes per class , Num. interfaces per class
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.53127865 0.28165818 1.886253 0.059260838
x1          0.06648172 0.02527876 2.629944 0.008539902
x2         -1.76845581 0.62755283 -2.818019 0.004832095
R2log = 0.9132966
Excluded as outliers: xalan JFreeChart Eclipse Ant ( 4 / 16 )
MMRE = 18.83749
Pred(25) = 68.75
Error range = [ -68.17059 .. 54.06654 ]
=====
Usability vs. Num. attributes per class , avg_loc_changed_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.972951e-01 2.420576e-01 -0.815075 0.41502940
x1          6.631019e-02 2.595649e-02 2.554667 0.01062895
x2         -2.666759e-05 1.150135e-05 -2.318648 0.02041413
R2log = 0.8997073
Excluded as outliers: Eclipse Hibernate ( 2 / 13 )
MMRE = 25.25004
Pred(25) = 61.53846
Error range = [ -89.89975 .. 37.30748 ]
=====
Usability vs. Num. attributes per class , avg_files_rem_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.079189821 0.3489044792 -3.093081 0.0019809005
x1          0.106122518 0.0290350152 3.654984 0.0002571979
x2          0.001670088 0.0004991416 3.345920 0.0008201009
R2log = 0.9318668
Excluded as outliers: Eclipse Xalan JMeter ( 3 / 13 )
MMRE = 15.01141
Pred(25) = 76.92308
Error range = [ -9.942106 .. 58.59815 ]
=====
Usability vs. Num. classes , Num. methods per class
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.3654822472 0.2776598126 1.316295 0.18807502
x1          0.0005182086 0.0002254357 2.298698 0.02152208
x2         -0.0444067358 0.0223474887 -1.987102 0.04691111
R2log = 0.8823279
Excluded as outliers: Log4J Hibernate Eclipse Ant ( 4 / 16 )
MMRE = 23.87645
Pred(25) = 56.25

```

```
Error range = [ -37.53895 .. 79.86704 ]
=====
Usability vs. Num. classes , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.6790602671 0.2766909451 -2.454219 0.014119083
x1          0.0005142939 0.0001912329  2.689358 0.007158957
x2          0.0014231908 0.0004942367  2.879573 0.003982138
R2log = 0.9120165
Excluded as outliers: Eclipse Hibernate Log4J ( 3 / 13 )
MMRE = 15.32
Pred(25) = 76.92308
Error range = [ -32.38148 .. 68.51867 ]
=====
Usability vs. Num. interfaces , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.754376269 0.322566924 -2.338666 0.019352731
x1          0.003882209 0.001836247  2.114208 0.034497476
x2          0.001728266 0.000538344  3.210337 0.001325794
R2log = 0.8958849
Excluded as outliers: Eclipse Hibernate Log4J JBoss ( 4 / 13 )
MMRE = 19.48534
Pred(25) = 69.23077
Error range = [ -35.4947 .. 62.24117 ]
=====
Usability vs. Num. interfaces per class , Num. methods per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.10872969 0.27592003 -0.3940624 0.6935350059
x1          -2.43659458 0.75854285 -3.2122043 0.0013172068
x2          0.09402438 0.02628879  3.5765959 0.0003480977
R2log = 0.9213135
Excluded as outliers: Eclipse JFreeChart Struts ( 3 / 16 )
MMRE = 19.54766
Pred(25) = 75
Error range = [ -65.03677 .. 44.84540 ]
=====
Usability vs. Num. interfaces per class , Num. packages
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.1009208832 0.283908870 3.877726 0.0001054372
x1          -1.3218246639 0.546487801 -2.418763 0.0155733681
x2          -0.0009345925 0.000467712 -1.998222 0.0456925924
R2log = 0.9454472
Excluded as outliers: Ant JFreeChart Httpunit ( 3 / 16 )
MMRE = 16.38311
Pred(25) = 75
Error range = [ -47.18181 .. 83.2688 ]
=====
Usability vs. Num. interfaces per class , number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.86880567 0.48786997 3.830540 0.0001278621
x1          -1.22314240 0.35128275 -3.481931 0.0004978114
x2          -0.03807267 0.01620513 -2.349421 0.0188026234
R2log = 0.9121575
Excluded as outliers: HttpUnit JBoss Eclipse Hibernate ( 4 / 13 )
MMRE = 19.81592
Pred(25) = 76.92308
Error range = [ -30.38837 .. 121.7043 ]
```

```
=====
Usability vs. Num. methods , Num. methods per class
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.268879e-01 2.931784e-01 2.138247 0.032496673
x1          9.974715e-05 3.269197e-05 3.051121 0.002279890
x2         -9.731235e-02 3.368719e-02 -2.888705 0.003868320
R2log = 0.8957724
Excluded as outliers: Log4J Hibernate Eclipse Ant Weka ( 5 / 16 )
MMRE = 30.65533
Pred(25) = 62.5
Error range = [ -58.40045 .. 132.0877 ]
=====

Usability vs. Num. methods , avg_files_rem_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.854953e-01 2.855710e-01 -2.050262 0.040338869
x1          3.983765e-05 1.818001e-05 2.191288 0.028430931
x2          1.395258e-03 4.957861e-04 2.814234 0.004889359
R2log = 0.906584
Excluded as outliers: Eclipse Hibernate Log4J ( 3 / 13 )
MMRE = 16.69642
Pred(25) = 69.23077
Error range = [ -31.44183 .. 56.53412 ]
=====

Usability vs. Num. methods per class , avg_files_added_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.1396787530 0.5052277324 -2.255772 0.024084891
x1          0.0818983999 0.0256710061 3.190307 0.001421216
x2          0.0004772905 0.0002211901 2.157829 0.030941097
R2log = 0.9499781
Excluded as outliers: JFreeChart JMeter ( 2 / 13 )
MMRE = 19.90111
Pred(25) = 84.61538
Error range = [ -19.52010 .. 92.81466 ]
=====

Usability vs. Num. methods per class , avg_files_rem_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.186171726 0.4239170061 -2.798123 0.0051400609
x1          0.079130273 0.0230647778 3.430784 0.0006018394
x2          0.001322637 0.0005088943 2.599042 0.0093484430
R2log = 0.9308186
Excluded as outliers: Eclipse JFreeChart JMeter ( 3 / 13 )
MMRE = 16.46971
Pred(25) = 84.61538
Error range = [ -10.12352 .. 82.66432 ]
=====

Usability vs. Num. packages , avg_files_rem_per_year
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.632150721 0.2806154924 -2.252729 0.024276227
x1          0.012457331 0.0048841889 2.550542 0.010755545
x2          0.001429013 0.0004869044 2.934895 0.003336606
R2log = 0.9111153
Excluded as outliers: Eclipse Hibernate Log4J JMeter ( 4 / 13 )
MMRE = 23.29986
Pred(25) = 69.23077
Error range = [ -29.14035 .. 84.75073 ]
=====
```

```

Usability vs. Num. parameters per method , avg_file_size
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.662568449 0.500019409 -1.325085 0.185142808
x1          1.635772542 0.686020387  2.384437 0.017105267
x2         -0.006787682 0.002015904 -3.367066 0.000759724
R2log = 0.9492697
Excluded as outliers: JBoss HttpUnit ( 2 / 13 )
MMRE = 16.54066
Pred(25) = 84.61538
Error range = [ -25.28347 .. 87.58913 ]
=====
Usability vs. avg_number_of_revisions_per_file Num. parameters per method ,
                                              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2175400 0.4997104 -0.4353321 0.6633213918
x1          1.9033040 0.7208828  2.6402407 0.0082847164
x2         -0.1797835 0.0546203 -3.2915140 0.0009964967
R2log = 0.9497509
Excluded as outliers: JFreeChart Findbugs Hibernate ( 3 / 13 )
MMRE = 17.82268
Pred(25) = 76.92308
Error range = [ -41.31139 .. 89.54016 ]
=====
Usability vs. number_of_commits , avg_files_rem_per_year
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.560598e-01 0.2373186595 -2.343094 0.019124586
x1          9.164413e-05 0.0000433666  2.113242 0.034580024
x2          1.538045e-03 0.0004758770  3.232023 0.001229172
R2log = 0.9103466
Excluded as outliers: Eclipse Log4J JBoss ( 3 / 13 )
MMRE = 11.57351
Pred(25) = 84.61538
Error range = [ -27.50233 .. 22.54850 ]
=====
Usability vs. number_of_commits , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.6557596526 5.109438e-01  3.240591 0.001192823
x1          0.0002307736 9.814174e-05  2.351431 0.018701341
x2          -0.3054341222 1.068833e-01 -2.857640 0.004268041
R2log = 0.9146407
Excluded as outliers: Eclipse HttpUnit Hibernate Findbugs ( 4 / 13 )
MMRE = 18.22890
Pred(25) = 76.92308
Error range = [ -58.51632 .. 27.41050 ]
=====
Usability vs. avg_loc_del_per_year , avg_file_size
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.906431e-01 1.754877e-01  4.505405 6.624651e-06
x1          -2.447899e-06 1.054856e-06 -2.320601 2.030840e-02
x2          -3.749960e-03 1.306794e-03 -2.869589 4.110060e-03
R2log = 0.9482245
Excluded as outliers: HttpUnit ( 1 / 13 )
MMRE = 16.21929
Pred(25) = 92.3077
Error range = [ -17.56321 .. 89.51372 ]
=====
```

```

Usability vs. avg_loc_del_per_year , avg_files_added_per_year
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -4.381405e-01 2.462994e-01 -1.778894 0.075257177
x1          -1.964038e-05 6.463761e-06 -3.038537 0.002377302
x2           2.562246e-03 8.235506e-04  3.111219 0.001863166
R2log = 0.952149
Excluded as outliers: struts JMeter Log4J ( 3 / 13 )
MMRE = 18.46832
Pred(25) = 84.61538
Error range = [ -63.83171 .. 79.02812 ]
=====
Usability vs. avg_loc_del_per_year , avg_files_rem_per_year
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 5.847125e-02 1.416577e-01  0.4127643 0.6797793110
x1          -5.142351e-06 1.973764e-06 -2.6053528 0.0091779701
x2           2.067575e-03 6.038529e-04  3.4239722 0.0006171294
R2log = 0.9219159
Excluded as outliers: Eclipse Struts ( 2 / 13 )
MMRE = 14.23453
Pred(25) = 84.61538
Error range = [ -10.72666 .. 46.82964 ]
=====
Usability vs. avg_major_release_per_year , avg_file_size
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.340175511 0.11194727  3.038712 0.002375918
x1          0.416710678 0.20641292  2.018821 0.043505860
x2          -0.002460059 0.00119341 -2.061370 0.039267745
R2log = 0.939896
Excluded as outliers: ( 0 / 13 )
MMRE = 15.50968
Pred(25) = 84.61538
Error range = [ -29.87119 .. 64.26385 ]
=====
Usability vs. LCOM , Num. methods per interface ,
avg_major_release_per_year
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -2.225976522 0.6913944961 -3.219546 0.0012839367
x1          0.001841313 0.0005583906  3.297536 0.0009753732
x2          0.887781407 0.2535378546  3.501573 0.0004625196
x3          -2.852646934 0.9046018817 -3.153483 0.0016133445
R2log = 0.9194257
Excluded as outliers: Log4J Eclipse ( 2 / 12 )
MMRE = 16.99972
Pred(25) = 83.33333
Error range = [ -91.57478 .. 48.61518 ]
=====
Usability vs. LCOM , RFC , avg_major_release_per_year
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.0374325055 3.772317e-01 -2.750120 5.957344e-03
x1          -0.0029982675 8.837536e-04 -3.392651 6.921979e-04
x2          0.0001167500 3.323474e-05  3.512890 4.432613e-04
x3           3.7688278352 9.537662e-01  3.951522 7.765582e-05
R2log = 0.9111722
Excluded as outliers: Hibernate JBoss Eclipse ( 3 / 13 )
MMRE = 22.19637
Pred(25) = 69.23077

```

```
Error range = [ -16.60725 .. 84.97433 ]
=====
Usability      vs.          NOC ,           Num.       methods ,
avg_number_of_revisions_per_file
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)  1.477409e+00 4.165150e-01  3.547071 0.0003895389
x1          9.295042e-01 3.299413e-01  2.817180 0.0048447320
x2         -3.405378e-05 1.038496e-05 -3.279144 0.0010412231
x3         -2.280161e-01 6.050863e-02 -3.768323 0.0001643476
R2log = 0.925187
Excluded as outliers: Eclipse Findbugs ( 2 / 13 )
MMRE = 16.9192
Pred(25) = 76.92308
Error range = [ -76.39676 .. 33.73418 ]
=====
Usability vs. NOC , RFC , avg_number_of_revisions_per_file
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)  1.452023e+00 4.107902e-01  3.534707 0.0004082269
x1          9.797879e-01 3.387624e-01  2.892257 0.0038248503
x2         -1.889615e-05 5.710899e-06 -3.308787 0.0009370094
x3         -2.288090e-01 6.045424e-02 -3.784829 0.0001538140
R2log = 0.9254908
Excluded as outliers: Eclipse Findbugs ( 2 / 13 )
MMRE = 16.98263
Pred(25) = 76.92308
Error range = [ -77.42106 .. 34.18234 ]
=====
Usability vs. Num. attributes per class , avg_loc_del_per_year , avg_files_added_per_year
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept) -1.095939e+00 4.723017e-01 -2.320421 0.0203181258
x1          9.250108e-02 3.409821e-02  2.712784 0.0066720606
x2         -1.132612e-05 3.704026e-06 -3.057786 0.0022297888
x3          1.801703e-03 5.166984e-04  3.486953 0.0004885581
R2log = 0.9513335
Excluded as outliers: xalan Struts ( 2 / 13 )
MMRE = 15.47669
Pred(25) = 76.92308
Error range = [ -21.37831 .. 57.05885 ]
=====
Usability vs. Num. methods per interface , number_of_developers , avg_files_added_per_year
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)  2.148413575 0.5505372832  3.902394 9.524582e-05
x1          0.600612126 0.2157155547  2.784278 5.364699e-03
x2         -0.089144031 0.0281935728 -3.161856 1.567668e-03
x3         -0.001220703 0.0004121622 -2.961706 3.059400e-03
R2log = 0.9356262
Excluded as outliers: HttpUnit JFreeChart Eclipse ( 3 / 12 )
MMRE = 28.56627
Pred(25) = 75
Error range = [ -2.608493 .. 166.0575 ]
=====
Usability vs. Num. parameters per method , loc_total , avg_number_of_revisions_per_file
              Estimate   Std. Error   z value   Pr(>|z|)
```

```
(Intercept) -2.928106e-01 5.039019e-01 -0.5810864 0.561182226
x1          2.192435e+00 7.503722e-01 2.9217968 0.003480185
x2         -1.727975e-06 6.001366e-07 -2.8793026 0.003985558
x3         -1.673897e-01 5.001101e-02 -3.3470567 0.000816745
R2log = 0.9534111
Excluded as outliers: JFreeChart Findbugs ( 2 / 13 )
MMRE = 15.80513
Pred(25) = 84.61538
Error range = [ -36.13173 .. 92.21334 ]
=====
Usability vs. RFC , avg_loc_added_per_year ,
avg_minor_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.241931e+00 4.321956e-01 -2.873540 0.004059000
x1          4.416968e-05 1.344214e-05 3.285911 0.001016533
x2         -1.329754e-05 4.248424e-06 -3.129993 0.001748107
x3          2.164158e-01 6.715334e-02 3.222711 0.001269834
R2log = 0.9179676
Excluded as outliers: Eclipse Log4J ( 2 / 13 )
MMRE = 18.31503
Pred(25) = 84.61538
Error range = [ -95.76142 .. 20.94983 ]
=====
Usability vs. number_of_commits , avg_loc_del_per_year ,
avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.124434e+00 6.089170e-01 3.488873 0.0004850622
x1          4.331064e-04 1.461201e-04 2.964043 0.0030362537
x2         -4.645171e-06 1.542704e-06 -3.011057 0.0026033973
x3          -4.413094e-01 1.452192e-01 -3.038918 0.0023742919
R2log = 0.95314
Excluded as outliers: HttpUnit PMD ( 2 / 13 )
MMRE = 14.72492
Pred(25) = 92.3077
Error range = [ -82.91184 .. 23.47023 ]
=====
Usability vs. number_of_commits , avg_minor_release_per_year ,
avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2750728727 2.926978e-01 -0.9397845 0.3473280997
x1          0.0002052348 7.785322e-05 2.6361763 0.0083846160
x2         -0.2371973319 7.185502e-02 -3.3010542 0.0009632227
x3          0.0025156723 6.821387e-04 3.6879189 0.0002260957
R2log = 0.9296755
Excluded as outliers: Eclipse PMD JBoss Findbugs ( 4 / 13 )
MMRE = 15.27782
Pred(25) = 69.23077
Error range = [ -34.63390 .. 56.81614 ]
=====
Usability vs. avg_major_release_per_year , avg_file_size ,
avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.062105133 0.270747737 3.922859 8.750445e-05
x1          0.664056360 0.223363072 2.972991 2.949133e-03
x2         -0.003598993 0.001300378 -2.767651 5.646192e-03
x3          -0.133133846 0.045095850 -2.952242 3.154759e-03
```

```
R2log = 0.953425
Excluded as outliers: Findbugs xalan ( 2 / 13 )
MMRE = 13.65839
Pred(25) = 84.61538
Error range = [ -53.9137 .. 10.38267 ]
=====
```

It must be observed that it is not straightforward to establish a relation between a quality that is based almost exclusively on external elements (e.g., the user interface) and the measures of the internal qualities. Anyway, it many statistically valid correlations were found.

Of these, a few ones are particularly interesting:

- Several qualities of the code appear correlated with usability. Quite noticeably, these are –to some extent– related with code features that are perceptible by the user. For instance, usability appears better for programs featuring large classes, with many methods per interface and many parameters per method.
- Other characteristics that seem to favour usability are: small size of the application, a big community of developers, small files.
- The model involving the number of effective LOC per class and the number of interfaces per class indicates that big classes with few interfaces are more usable.
- The model involving the number of children (per class) and the number of developers suggests that many developers and big generalization hierarchies tend to lower usability.
- Similarly, many developers and many interfaces tend to lower usability.

## 2.3 Portability

The significant models found for OSS product Portability are reported below.

```
=====
Portability vs. NOC
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.006118287 0.1886794 0.03242690 0.97413161
x1          0.440620371 0.1784647 2.46894945 0.01355104
R2log = 0.9353108
Excluded as outliers: ( 0 / 16 )
MMRE = 10.71177
Pred(25) = 93.75
Error range = [ -24.75138 .. 58.92724 ]
=====

Portability vs. Num. packages
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.004194415 0.238703004 -0.01757169 0.98598054
x1          0.008580771 0.004146088 2.06960670 0.03848919
R2log = 0.9016106
Excluded as outliers: Hibernate Eclipse Log4J JBoss HttpUnit ( 5 /
16 )
MMRE = 18.38079
Pred(25) = 81.25
```

```
Error range = [ -20.48394 .. 70.48043 ]
=====
Portability vs. number_of_commits
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0120278013 1.609803e-01 0.07471598 0.940440690
x1          0.0001965578 6.729124e-05 2.92100173 0.003489079
R2log = 0.9422092
Excluded as outliers: Findbugs Xerces ( 2 / 13 )
MMRE = 15.81645
Pred(25) = 76.92308
Error range = [ -21.73155 .. 50.15273 ]
=====
Portability vs. avg_loc_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.848468e-01 1.239126e-01 1.491751 0.13576436
x1          1.332811e-06 6.510682e-07 2.047115 0.04064683
R2log = 0.9343379
Excluded as outliers: Log4J ( 1 / 13 )
MMRE = 14.23090
Pred(25) = 84.61538
Error range = [ -25.49119 .. 53.39045 ]
=====
Portability vs. avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.760508e-01 1.262019e-01 1.394994 0.16301761
x1          2.217064e-06 1.068060e-06 2.075786 0.03791379
R2log = 0.9344607
Excluded as outliers: Log4J ( 1 / 13 )
MMRE = 14.26379
Pred(25) = 84.61538
Error range = [ -25.27657 .. 52.4856 ]
=====
Portability vs. avg_minor_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.5721014 0.10693332 5.350077 8.791698e-08
x1          -0.1004144 0.04541679 -2.210954 2.703904e-02
R2log = 0.939608
Excluded as outliers: PMD ( 1 / 13 )
MMRE = 15.41213
Pred(25) = 84.61538
Error range = [ -52.87962 .. 62.96803 ]
=====
Portability vs. avg_file_size
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.659864885 0.11899503 5.545315 2.934258e-08
x1          -0.002665836 0.00124469 -2.141767 3.221220e-02
R2log = 0.943187
Excluded as outliers: HttpUnit ( 1 / 13 )
MMRE = 13.16084
Pred(25) = 84.61538
Error range = [ -15.15406 .. 82.41299 ]
=====
Portability vs. avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1108000471 0.1437219040 0.7709336 0.44074628
x1          0.0006008881 0.0002678429 2.2434350 0.02486878
```

```
R2log = 0.9352592
Excluded as outliers: Log4J ( 1 / 13 )
MMRE = 12.84646
Pred(25) = 84.61538
Error range = [ -18.42208 .. 47.84871 ]
=====
Portability vs. eLOC per class , Num. interfaces per class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.01542158 0.28934495  0.05329827 0.95749426
x1          0.01168316 0.00522462  2.23617506 0.02534031
x2         -0.83340573 0.35321617 -2.35947786 0.01830067
R2log = 0.935528
Excluded as outliers: ( 0 / 16 )
MMRE = 10.39283
Pred(25) = 93.75
Error range = [ -17.66421 .. 59.24904 ]
=====
Portability vs. eLOC per class , number_of_developers
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.57724529 0.571693613 -2.758900 0.005799635
x1          0.02267898 0.006953652  3.261449 0.001108444
x2          0.03269050 0.015799349  2.069104 0.038536305
R2log = 0.9480355
Excluded as outliers: JBoss Xerces JFreeChart ( 3 / 13 )
MMRE = 16.68267
Pred(25) = 76.92308
Error range = [ -11.22802 .. 52.24266 ]
=====
Portability vs. eLOC per class , number_of_commits
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.18305335 0.4995846665 -2.368074 0.017880969
x1          0.01630795 0.0067285300  2.423702 0.015363199
x2          0.00023042 0.0000800607  2.878066 0.004001213
R2log = 0.950917
Excluded as outliers: Findbugs Xerces JFreeChart Hibernate ( 4 / 13
)
MMRE = 17.05902
Pred(25) = 76.92308
Error range = [ -18.53022 .. 49.13461 ]
=====
Portability vs. eLOC per class , avg_files_added_per_year
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.659366993 0.418045644 -1.577261 0.11473554
x1          0.011194816 0.005191845  2.156231 0.03106565
x2          0.000291356 0.000146580  1.987693 0.04684569
R2log = 0.9398228
Excluded as outliers: Log4J JFreeChart ( 2 / 13 )
MMRE = 13.62420
Pred(25) = 76.92308
Error range = [ -26.68583 .. 33.47497 ]
=====
Portability vs. eLOC , number_of_commits
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -2.518569e-01 2.173471e-01 -1.158777 0.24654699
x1          3.117141e-06 1.478170e-06  2.108784 0.03496320
x2          1.462096e-04 5.868028e-05  2.491630 0.01271583
```

```
R2log = 0.9427572
Excluded as outliers: Log4J Findbugs ( 2 / 13 )
MMRE = 13.02975
Pred(25) = 84.61538
Error range = [ -20.71025 .. 33.50521 ]
=====
Portability vs. eLOC , avg_loc_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.329257e-01 2.400702e-01 -1.386785 0.16550734
x1           7.566515e-06 3.720623e-06  2.033669 0.04198499
x2           9.848155e-06 4.202827e-06  2.343222 0.01911802
R2log = 0.902121
Excluded as outliers: Log4J Hibernate Eclipse ( 3 / 13 )
MMRE = 18.04398
Pred(25) = 69.23077
Error range = [ -33.59961 .. 64.02265 ]
=====
Portability vs. NOC , avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.3960619 0.2586403 -1.531323 0.125689464
x1           0.7118446 0.2200544  3.234858 0.001217032
x2           0.5762714 0.2515248  2.291112 0.021956957
R2log = 0.9412296
Excluded as outliers: ( 0 / 13 )
MMRE = 10.64644
Pred(25) = 92.3077
Error range = [ -21.11247 .. 46.37953 ]
=====
Portability vs. Num. attributes per class , Num. interfaces per
class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.32471919 0.25489890  1.273914 0.202694026
x1           0.07036238 0.02574657  2.732884 0.006278249
x2          -1.23692159 0.60769420 -2.035434 0.041807198
R2log = 0.9195973
Excluded as outliers: xalan Eclipse JFreeChart ( 3 / 16 )
MMRE = 16.44728
Pred(25) = 81.25
Error range = [ -52.68945 .. 52.80556 ]
=====
Portability vs. Num. attributes per class , avg_loc_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.245831e-01 3.029183e-01 -1.731764 0.083315563
x1           7.867423e-02 2.670383e-02  2.946177 0.003217278
x2           2.485275e-06 8.292496e-07  2.997017 0.002726354
R2log = 0.9428937
Excluded as outliers: xalan ( 1 / 13 )
MMRE = 11.46008
Pred(25) = 76.92308
Error range = [ -27.69325 .. 44.04283 ]
=====
Portability vs. Num. attributes per class , avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.985127258 0.4267070915 -2.308673 0.020961719
x1           0.103678016 0.0322403014  3.215789 0.001300863
```

```

x2          0.000649628 0.0002089110  3.109592 0.001873462
R2log =  0.9437716
Excluded as outliers: xalan ( 1 / 13 )
MMRE = 10.74839
Pred(25) = 84.61538
Error range = [ -27.70668 .. 50.44839 ]
=====
Portability vs. Num. attributes per class , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.747862486 0.3316380303 -2.255056 0.0241297844
x1          0.083732794 0.0262664842  3.187819 0.0014335031
x2          0.001205168 0.0003517711  3.425999 0.0006125431
R2log = 0.9458639
Excluded as outliers: xalan ( 1 / 13 )
MMRE = 10.40512
Pred(25) = 92.3077
Error range = [ -14.36646 .. 45.60686 ]
=====
Portability vs. Num. classes , number_of_commits
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.5430990054 2.719832e-01 -1.996811 0.045845701
x1          0.0004568289 1.694052e-04  2.696664 0.007003791
x2          0.0001392666 6.424995e-05  2.167576 0.030190957
R2log = 0.9456483
Excluded as outliers: Hibernate Log4J Findbugs ( 3 / 13 )
MMRE = 14.21276
Pred(25) = 76.92308
Error range = [ -25.76938 .. 45.86888 ]
=====
Portability vs. Num. classes , avg_minor_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.9088609952 0.3716682083 -2.445356 0.014470932
x1          0.0006517561 0.0002123150  3.069759 0.002142312
x2          0.1269413418 0.0520342523  2.439573 0.014704643
R2log = 0.908363
Excluded as outliers: Hibernate Log4J Eclipse ( 3 / 13 )
MMRE = 13.62914
Pred(25) = 84.61538
Error range = [ -45.61771 .. 59.36778 ]
=====
Portability vs. Num. interfaces , number_of_commits
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.4478618249 2.769859e-01 -1.616912 0.105897247
x1          0.0013682635 6.026205e-04  2.270523 0.023175891
x2          0.0001952674 6.400575e-05  3.050778 0.002282490
R2log = 0.943583
Excluded as outliers: Findbugs Log4J ( 2 / 13 )
MMRE = 12.57970
Pred(25) = 84.61538
Error range = [ -26.78737 .. 41.59215 ]
=====
Portability vs. Num. interfaces per class , avg_loc_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.970830e-01 5.201808e-01 -1.724560 0.084606803
x1          1.929766e+00 9.568846e-01  2.016718 0.043724982
x2          2.135463e-06 7.441444e-07  2.869689 0.004108755

```

```
R2log = 0.9506278
Excluded as outliers: Log4J JFreeChart PMD Xalan ( 4 / 13 )
MMRE = 20.84615
Pred(25) = 69.23077
Error range = [ -44.30088 .. 79.26454 ]
=====
Portability vs. Num. interfaces per class ,
avg_loc_changed_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.809138e-01 5.540023e-01 -1.770595 0.07662802
x1           1.965565e+00 9.749763e-01  2.016013 0.04379862
x2           1.238586e-05 5.820232e-06  2.128069 0.03333135
R2log = 0.9186119
Excluded as outliers: Log4J JFreeChart Eclipse PMD ( 4 / 13 )
MMRE = 20.58737
Pred(25) = 76.92308
Error range = [ -50.07036 .. 81.08735 ]
=====
Portability vs. Num. methods , number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.855473e-01 3.506386e-01 -1.955139 0.05056663
x1           3.950307e-05 1.741586e-05  2.268223 0.02331559
x2           2.514099e-02 1.208546e-02  2.080267 0.03750105
R2log = 0.88616
Excluded as outliers: JBoss Hibernate Log4J Eclipse ( 4 / 13 )
MMRE = 13.27631
Pred(25) = 76.92308
Error range = [ -31.64981 .. 39.47828 ]
=====
Portability vs. Num. methods , number_of_commits
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.118394e-01 2.777290e-01 -1.842946 0.06533696
x1           3.540696e-05 1.796632e-05  1.970741 0.04875352
x2           1.457845e-04 6.401731e-05  2.277267 0.02277027
R2log = 0.9059039
Excluded as outliers: Findbugs Hibernate Log4J Eclipse ( 4 / 13 )
MMRE = 13.31735
Pred(25) = 76.92308
Error range = [ -25.97897 .. 38.64894 ]
=====
Portability vs. Num. methods per class , number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.76214314 0.41110009 -1.853911 0.06375184
x1           0.04083547 0.01711757  2.385587 0.01705187
x2           0.02705574 0.01247101  2.169491 0.03004546
R2log = 0.9069518
Excluded as outliers: Eclipse JBoss ( 2 / 13 )
MMRE = 14.42247
Pred(25) = 92.3077
Error range = [ -19.49528 .. 35.44429 ]
=====
Portability vs. Num. methods per class , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.401151288 0.4784291674 -2.928649 0.0034043820
x1           0.086881803 0.0241250590  3.601309 0.0003166185
x2           0.001779986 0.0004668373  3.812861 0.0001373672
```

```
R2log = 0.9506442
Excluded as outliers: Saxon JFreeChart ( 2 / 13 )
MMRE = 10.61243
Pred(25) = 84.61538
Error range = [ -16.01616 .. 43.45743 ]
=====
Portability vs. Num. packages , files_count_total
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.0301593966 1.916745e-01 -0.1573469 0.874971424
x1          -0.0014385022 7.141690e-04 -2.0142322 0.043985166
x2           0.0002725920 8.894285e-05  3.0647998 0.002178157
R2log = 0.944879
Excluded as outliers: Struts ( 1 / 13 )
MMRE = 12.51286
Pred(25) = 84.61538
Error range = [ -15.83264 .. 67.21671 ]
=====
Portability vs. number_of_commits , avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.135053e-01 1.795502e-01 -0.6321646 0.52727931
x1           1.306292e-04 5.840338e-05  2.2366724 0.02530776
x2           2.168563e-06 1.101396e-06  1.9689230 0.04896193
R2log = 0.9421034
Excluded as outliers: Findbugs Log4J ( 2 / 13 )
MMRE = 12.53054
Pred(25) = 76.92308
Error range = [ -26.61639 .. 38.44538 ]
=====
Portability vs. number_of_commits, avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.9410407612 0.3063268623  3.072015 0.002126189
x1          0.0002265082 0.0000896267  2.527240 0.011496287
x2          -0.1857985624 0.0831592345 -2.234251 0.025466594
R2log = 0.9452566
Excluded as outliers: HttpUnit ( 1 / 13 )
MMRE = 7.85448
Pred(25) = 92.3077
Error range = [ -28.58283 .. 21.08558 ]
=====
Portability vs. avg_loc_del_per_year , avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.153454e-01 3.493394e-01 -1.761454 0.078161555
x1          -4.343670e-05 1.381101e-05 -3.145077 0.001660431
x2           5.419683e-03 1.690392e-03  3.206169 0.001345150
R2log = 0.9509937
Excluded as outliers: Struts JMeter Saxon JBoss ( 4 / 13 )
MMRE = 26.54814
Pred(25) = 69.23077
Error range = [ -93.72741 .. 114.0597 ]
=====
Portability vs. Comment_lines, number_of_developers, avg_file_size
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.449330e-01 3.461394e-01 -1.863218 0.062431648
x1           1.631269e-05 5.194586e-06  3.140326 0.001687601
x2            5.893910e-02 1.942241e-02  3.034592 0.002408609
x3           -1.388103e-02 4.895099e-03 -2.835700 0.004572538
```

```
R2log = 0.9198574
Excluded as outliers: Eclipse JFreeChart Log4J ( 3 / 13 )
MMRE = 13.94420
Pred(25) = 84.61538
Error range = [ -82.56909 .. 39.83818 ]
=====
Portability vs. Comment lines per class , Num. parameters per
method , avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.175496e+00 2.080734e+00 -2.967941 0.0029980210
x1           3.821655e-02 1.122704e-02  3.403975 0.0006641286
x2           5.711432e+00 1.929679e+00  2.959784 0.0030785493
x3           6.624022e-06 2.129262e-06  3.110948 0.0018648771
R2log = 0.9504527
Excluded as outliers: JFreeChart Xerces Xalan ( 3 / 13 )
MMRE = 21.36284
Pred(25) = 76.92308
Error range = [ -8.991063 .. 92.0739 ]
=====
Portability vs. NOC , Num. public methods ,
avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.047801e+00 4.698460e-01 2.230095 0.025741130
x1           1.065387e+00 4.068001e-01 2.618945 0.008820225
x2           -3.320832e-05 1.245291e-05 -2.666712 0.007659722
x3           -1.739705e-01 6.110712e-02 -2.846976 0.004413663
R2log = 0.9239383
Excluded as outliers: Findbugs Eclipse ( 2 / 12 )
MMRE = 12.1079
Pred(25) = 91.66667
Error range = [ -64.46087 .. 19.66994 ]
=====
Portability vs. NOC , RFC , avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.439646e-01 4.196585e-01 2.249364 0.0244893781
x1           1.153994e+00 3.462502e-01 3.332831 0.0008596702
x2           -1.542661e-05 5.963490e-06 -2.586842 0.0096860024
x3           -1.701780e-01 6.089283e-02 -2.794714 0.0051945605
R2log = 0.9236245
Excluded as outliers: Eclipse Findbugs ( 2 / 13 )
MMRE = 12.11932
Pred(25) = 92.3077
Error range = [ -65.87813 .. 21.19443 ]
=====
Portability vs. Num. attributes per class , number_of_developers ,
avg_file_size
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.55255496 0.338490736 -1.632408 0.102593582
x1           0.05533821 0.017885634  3.094003 0.001974754
x2           0.04609953 0.016996677  2.712267 0.006682464
x3           -0.01094665 0.003521662 -3.108375 0.001881193
R2log = 0.9187988
Excluded as outliers: JFreeChart Hibernate Eclipse ( 3 / 13 )
MMRE = 16.86146
Pred(25) = 84.61538
Error range = [ -86.40784 .. 16.88319 ]
```

```
=====
Portability vs. Num. interfaces , avg_loc_del_per_year ,
avg_files_added_per_year
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.388153e-01 3.234037e-01 -1.047654 0.294797884
x1          -2.300478e-03 8.562732e-04 -2.686617 0.007217970
x2          -3.272058e-05 1.138772e-05 -2.873321 0.004061812
x3           4.874270e-03 1.549901e-03  3.144891 0.001661487
R2log = 0.9295195
Excluded as outliers: struts JMeter Saxon Eclipse ( 4 / 13 )
MMRE = 24.08698
Pred(25) = 76.92308
Error range = [ -86.86945 .. 113.7307 ]
=====

Portability vs. RFC , number_of_developers , avg_file_size
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.393973e-01 3.080732e-01 -1.101677 0.2706019396
x1           1.583194e-05 5.672573e-06  2.790962 0.0052551626
x2           5.717448e-02 1.832104e-02  3.120701 0.0018042140
x3          -1.376401e-02 3.891501e-03 -3.536940 0.0004047911
R2log = 0.9481513
Excluded as outliers: JFreeChart xalan ( 2 / 13 )
MMRE = 17.54613
Pred(25) = 84.61538
Error range = [ -92.08473 .. 22.93954 ]
=====
```

The results found for portability seem very reasonable, and generally conformant to the expectations.

The fact that portability grows with the NOC (number of children, i.e., the number of sub-classes) seems to indicate that portability is favoured by rich generalization hierarchies. This seems reasonable: the richer the hierarchy, the easier to encapsulate and share the required adaptations.

The fact that portability grows with the number of packages seems to indicate that in a system with several packages it is easier to find packages that do not depend on the specific platform, and can thus be ported with little effort.

## 2.4 How well are functional requirements satisfied

The significant models found for the quality of functionality of OSS product are reported below.

```
=====
Functionality vs. Num. interfaces per class
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.7450709 0.2549091 2.922888 0.00346801
x1          -1.0983750 0.5303379 -2.071085 0.03835083
R2log = 0.9317681
Excluded as outliers: JFreeChart Ant Log4J ( 3 / 16 )
MMRE = 18.98631
Pred(25) = 87.5
Error range = [ -52.31743 .. 83.93556 ]
```

```
=====
Functionality vs. Num. parameters per method
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.49077  1.185073 -2.945616 0.003223126
x1          4.16690  1.328144  3.137385 0.001704621
R2log = 0.914783
Excluded as outliers: JFreeChart Eclipse Log4J Xerces Xalan ( 5 / 16 )
)
MMRE = 25.05199
Pred(25) = 68.75
Error range = [ -32.37865 .. 85.00896 ]
=====

Functionality vs. eLOC per class , NOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.25111836 0.535759829 4.201730 2.648825e-05
x1         -0.01058974 0.004512445 -2.346784 1.893621e-02
x2         -1.03228470 0.328260474 -3.144712 1.662503e-03
R2log = 0.9466003
Excluded as outliers: struts HttpUnit JBoss ( 3 / 16 )
MMRE = 24.09179
Pred(25) = 81.25
Error range = [ -17.43163 .. 135.2985 ]
=====

Functionality vs. McCabe , Num. interfaces per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1368442  0.3981805  0.3436739 0.731091577
x1          0.4176823  0.1753541  2.3819360 0.017221888
x2         -1.4897286  0.5364611 -2.7769553 0.005487073
R2log = 0.9373074
Excluded as outliers: JFreeChart Ant ( 2 / 16 )
MMRE = 19.12763
Pred(25) = 75
Error range = [ -65.79397 .. 76.1848 ]
=====

Functionality vs. McCabe , Num. packages
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.757822813 0.607124101 -2.895327 0.003787637
x1          0.825670508 0.260955347  3.164030 0.001556008
x2          0.001780181 0.000643463  2.766563 0.005665057
R2log = 0.9403595
Excluded as outliers: Log4J Xerces ( 2 / 16 )
MMRE = 17.37332
Pred(25) = 75
Error range = [ -26.99862 .. 89.81255 ]
=====

Functionality vs. NOC , avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.867490  0.4552082 -1.905699 0.056689205
x1          0.853847  0.3708497  2.302407 0.021312227
x2          1.020235  0.3187589  3.200648 0.001371188
R2log = 0.9454726
Excluded as outliers: Findbugs Struts ( 2 / 13 )
MMRE = 18.66207
Pred(25) = 61.53846
Error range = [ -49.12099 .. 36.20203 ]
=====
```

```

Functionality           vs.           Num.       abstract      classes   ,
avg_major_release_per_year
                         Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.44136623 0.47193257 -3.054178 0.0022567789
x1          0.02767946 0.00830486  3.332924 0.0008593848
x2          0.98212339 0.37244580  2.636957 0.0083653556
R2log = 0.9055075
Excluded as outliers: Hibernate Log4J Eclipse ( 3 / 12 )
MMRE = 17.23076
Pred(25) = 75
Error range = [ -40.51264 .. 66.5348 ]
=====
Functionality vs. Num. attributes per class , avg_loc_del_per_year
                         Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.014968e+00 3.764819e-01 -2.695928 0.007019284
x1          1.175199e-01 3.309772e-02  3.550694 0.000384217
x2          2.321163e-05 8.270373e-06  2.806600 0.005006741
R2log = 0.9202994
Excluded as outliers: xalan JMeter Eclipse Hibernate ( 4 / 13 )
MMRE = 32.83968
Pred(25) = 61.53846
Error range = [ -13.42438 .. 105.1776 ]
=====
Functionality           vs.           Num.       attributes     per      class   ,
avg_files_rem_per_year
                         Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.148410053 0.3905895255 -2.940197 0.0032800390
x1          0.113267363 0.0314302479  3.603769 0.0003136355
x2          0.002018125 0.0006017499  3.353761 0.0007972107
R2log = 0.931045
Excluded as outliers: Eclipse Xalan JMeter Saxon ( 4 / 13 )
MMRE = 21.48039
Pred(25) = 69.23077
Error range = [ -19.09805 .. 92.29205 ]
=====
Functionality vs. Num. classes , avg_loc_del_per_year
                         Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -6.153276e-01 2.659876e-01 -2.313370 0.020702332
x1          5.774833e-04 1.948560e-04  2.963641 0.003040227
x2          2.167706e-05 8.133599e-06  2.665125 0.007695968
R2log = 0.9090974
Excluded as outliers: Log4J Hibernate Eclipse JMeter ( 4 / 13 )
MMRE = 27.8931
Pred(25) = 69.23077
Error range = [ -36.41704 .. 94.76576 ]
=====
Functionality           vs.           Num.       classes , avg_files_rem_per_year
                         Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.895667615 0.3308817598 -2.706911 0.006791237
x1          0.000658236 0.0002233337  2.947320 0.003205413
x2          0.001853535 0.0006346102  2.920746 0.003491948
R2log = 0.9091585
Excluded as outliers: Eclipse Hibernate Log4J Saxon ( 4 / 13 )
MMRE = 20.39907
Pred(25) = 61.53846
Error range = [ -34.98736 .. 60.99512 ]

```

```
=====
Functionality vs. Num. interfaces per class , Num. methods per
class
              Estimate Std. Error   z value Pr(>|z|)
(Intercept)  0.18749004 0.27605507  0.6791762 0.4970262105
x1          -2.50502145 0.74828084 -3.3477022 0.0008148451
x2           0.07833879 0.02596526  3.0170616 0.0025523790
R2log = 0.916687
Excluded as outliers: JFreechart Eclipse Struts ( 3 / 16 )
MMRE = 21.79355
Pred(25) = 68.75
Error range = [ -76.55168 .. 68.99152 ]
=====

Functionality vs. Num. interfaces per class , Num. parameters per
method
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.9015462 0.6673132 -1.351009 0.17669252
x1          -0.6489929 0.2948490 -2.201103 0.02772876
x2           1.5241062 0.7718360  1.974650 0.04830783
R2log = 0.905158
Excluded as outliers: Eclipse Log4J ( 2 / 16 )
MMRE = 16.91334
Pred(25) = 87.5
Error range = [ -24.15567 .. 73.72886 ]
=====

Functionality vs. Num. methods , avg_loc_del_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -5.331824e-01 2.680320e-01 -1.989249 0.046673688
x1           4.647024e-05 1.807923e-05  2.570366 0.010159110
x2           2.103153e-05 7.991604e-06  2.631703 0.008495798
R2log = 0.9034713
Excluded as outliers: Log4J Hibernate Eclipse JMeter ( 4 / 13 )
MMRE = 28.68040
Pred(25) = 53.84615
Error range = [ -35.85081 .. 93.79683 ]
=====

Functionality vs. Num. methods , avg_files_rem_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -9.435916e-01 3.721781e-01 -2.535323 0.011234371
x1           6.253675e-05 2.318895e-05  2.696834 0.007000221
x2           2.004282e-03 6.768899e-04  2.961016 0.003066264
R2log = 0.9057255
Excluded as outliers: Eclipse Log4J Hibernate Saxon ( 4 / 13 )
MMRE = 20.84533
Pred(25) = 61.53846
Error range = [ -36.84204 .. 57.3247 ]
=====

Functionality vs. Num. parameters per method ,
avg_loc_added_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.957916e+00 9.140141e-01 -2.142107 0.03218486
x1           2.224509e+00 9.385256e-01  2.370217 0.01777765
x2           2.151534e-06 1.017503e-06  2.114524 0.03447050
R2log = 0.9407627
Excluded as outliers: Log4J JFreeChart ( 2 / 13 )
MMRE = 19.27807
=====
```

```

Pred(25) = 76.92308
Error range = [ -25.93316 .. 58.3703 ]
=====
Functionality vs. Num. parameters per method , avg_file_size
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.733445101 0.991043422 -1.749111 0.0802718080
x1          2.901564383 1.157421528  2.506921 0.0121787913
x2         -0.008904432 0.002528503 -3.521623 0.0004289144
R2log = 0.9154348
Excluded as outliers: Httpunit Eclipse JBoss Hibernate ( 4 / 13 )
MMRE = 17.35488
Pred(25) = 84.61538
Error range = [ -34.56152 .. 110.4916 ]
=====
Functionality vs. Num. parameters per method , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.3580182 0.97622847 -1.391086 0.164199202
x1          2.6272449 1.11027959  2.366291 0.017967304
x2         -0.1449286 0.04713002 -3.075080 0.002104461
R2log = 0.9212523
Excluded as outliers: Findbugs JFreeChart Eclipse Log4J ( 4 / 13 )
MMRE = 16.37434
Pred(25) = 84.61538
Error range = [ -46.54962 .. 62.30518 ]
=====
Functionality vs. number_of_commits , avg_files_added_per_year
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.6668107696 2.932823e-01 -2.273614 0.022989201
x1          0.0001439326 4.906963e-05  2.933232 0.003354535
x2          0.0006864119 2.524275e-04  2.719244 0.006543131
R2log = 0.900696
Excluded as outliers: Eclipse Log4J JBoss ( 3 / 13 )
MMRE = 14.82855
Pred(25) = 84.61538
Error range = [ -30.28455 .. 21.50222 ]
=====
Functionality vs. avg_major_release_per_year , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8607637 0.25257748  3.407920 0.0006546019
x1          0.7928650 0.24034152  3.298910 0.0009706113
x2         -0.1335213 0.04397685 -3.036171 0.0023960304
R2log = 0.948359
Excluded as outliers: Findbugs JBoss ( 2 / 13 )
MMRE = 14.71929
Pred(25) = 84.61538
Error range = [ -57.35873 .. 33.24373 ]
=====
Functionality vs. CBO , number_of_developers , avg_file_size
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.465616614 0.889233767 -2.772743 0.005558602
x1          0.424667287 0.135511970  3.133799 0.001725590
x2          0.069288488 0.021160921  3.274361 0.001059014
x3         -0.009744562 0.003548663 -2.745981 0.006033019
R2log = 0.9169556

```

```

Excluded as outliers: Eclipse JFreeChart Log4J Struts ( 4 / 13 )
MMRE = 25.39488
Pred(25) = 69.23077
Error range = [ -85.28802 .. 80.22599 ]
=====
Functionality vs. LCOM , eLOC per class , avg_loc_changed_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.2967276150 4.893075e-01 2.650128 0.0080461230
x1          0.0033643798 9.897612e-04 3.399183 0.0006758744
x2         -0.0377942452 1.305957e-02 -2.893990 0.0038038087
x3         -0.0001064282 3.091127e-05 -3.443023 0.0005752505
R2log = 0.9208278
Excluded as outliers: Eclipse Hibernate Xalan PMD ( 4 / 13 )
MMRE = 31.92146
Pred(25) = 69.23077
Error range = [ -99.99955 .. 24.34012 ]
=====
Functionality vs. LCOM , McCabe , Num. methods
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.402197e+00 9.983944e-01 -3.407668 0.0006552057
x1          1.090754e-03 3.170164e-04 3.440688 0.0005802378
x2          1.807018e+00 4.954502e-01 3.647224 0.0002650884
x3         -7.960914e-05 2.117883e-05 -3.758901 0.0001706613
R2log = 0.929928
Excluded as outliers: Hibernate Eclipse Xerces PMD JFreeChart ( 5 /
16 )
MMRE = 23.15751
Pred(25) = 75
Error range = [ -96.11598 .. 67.56698 ]
=====
Functionality vs. LCOM , McCabe , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.311980e+00 9.881129e-01 -3.351823 0.0008028135
x1          1.030515e-03 3.083718e-04 3.341794 0.0008323872
x2          1.760586e+00 4.900222e-01 3.592870 0.0003270560
x3         -4.054816e-05 1.096388e-05 -3.698341 0.0002170129
R2log = 0.9292027
Excluded as outliers: Hibernate Eclipse Xerces PMD JFreeChart ( 5 /
16 )
MMRE = 22.94586
Pred(25) = 75
Error range = [ -95.24089 .. 66.66683 ]
=====
Functionality vs. LCOM , Num. classes , avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.919498470 0.3201555460 -2.872037 4.078356e-03
x1          -0.001279773 0.0004963768 -2.578228 9.930851e-03
x2          0.001331442 0.0003927655 3.389917 6.991389e-04
x3          2.554518397 0.6226761886 4.102483 4.087401e-05
R2log = 0.9125509
Excluded as outliers: Hibernate JBoss Eclipse ( 3 / 13 )
MMRE = 19.34007
Pred(25) = 76.92308
Error range = [ -11.54300 .. 68.1454 ]
=====
Functionality vs. eLOC , number_of_developers , avg_file_size

```

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.992921e-01 3.490623e-01 -2.003345 0.045140323
x1          1.266613e-05 4.502034e-06  2.813424 0.004901702
x2          5.409217e-02 1.751427e-02  3.088463 0.002011946
x3         -1.222264e-02 4.037460e-03 -3.027310 0.002467405
R2log = 0.9064673
Excluded as outliers: Eclipse JFreeChart Log4J Hibernate ( 4 / 13 )
MMRE = 21.18931
Pred(25) = 76.92308
Error range = [ -87.69373 .. 48.97809 ]
=====
Functionality vs. McCabe , Num. attributes per class , RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.872054e+00 7.575738e-01 -3.791121 1.499686e-04
x1          1.362732e+00 3.416458e-01  3.988727 6.642878e-05
x2          7.196692e-02 2.554630e-02  2.817117 4.845680e-03
x3         -2.737974e-05 8.377811e-06 -3.268126 1.082621e-03
R2log = 0.9241361
Excluded as outliers: Hibernate Eclipse xalan xerces ( 4 / 16 )
MMRE = 23.78099
Pred(25) = 68.75
Error range = [ -88.6702 .. 80.99423 ]
=====
Functionality vs. McCabe , Num. public methods , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.328009e-01 5.559435e-01  1.497996 0.1341343503
x1          5.319692e-01 1.991706e-01  2.670923 0.0075642963
x2         -5.559763e-05 2.091174e-05 -2.658680 0.0078447522
x3         -1.825461e-01 5.227363e-02 -3.492126 0.0004791929
R2log = 0.9527055
Excluded as outliers: Hibernate Findbugs ( 2 / 12 )
MMRE = 15.20557
Pred(25) = 83.33333
Error range = [ -63.70004 .. 12.57883 ]
=====
Functionality vs. NOC , number_of_developers , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.29958018 0.92967571  3.549173 0.0003864427
x1         -2.61164122 0.79706360 -3.276578 0.0010507321
x2          0.04861172 0.01489923  3.262699 0.0011035657
x3         -0.21208193 0.05929593 -3.576669 0.0003479998
R2log = 0.9218768
Excluded as outliers: JBoss Struts Eclipse Xalan ( 4 / 13 )
MMRE = 23.63456
Pred(25) = 76.92308
Error range = [ -46.20078 .. 96.78939 ]
=====
Functionality vs. Num. abstract classes , number_of_commits , avg_number_of_revisions_per_file
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.0739778658 0.6223007294  3.332758 0.0008598964
x1        -0.0110511395 0.0039341929 -2.808998 0.0049695969
x2          0.0002598977 0.0000997688  2.605000 0.0091874229
x3        -0.3278165878 0.1120994265 -2.924338 0.0034518995

```

```
R2log = 0.9545767
Excluded as outliers: Hibernate Findbugs HttpUnit ( 3 / 12 )
MMRE = 19.57507
Pred(25) = 66.66667
Error range = [ -61.55242 .. 27.00523 ]
=====
Functionality vs. Num. methods , Num. methods per class ,
avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0694395339 3.359944e-01 0.2066687 0.8362685744
x1          0.0001311857 3.645003e-05 3.5990565 0.0003193738
x2         -0.1163979089 3.781620e-02 -3.0779908 0.0020840132
x3          1.3582689658 4.746267e-01 2.8617625 0.0042129244
R2log = 0.8989034
Excluded as outliers: Hibernate Log4J Eclipse JBoss ( 4 / 13 )
MMRE = 17.50591
Pred(25) = 76.92308
Error range = [ -50.17684 .. 67.15441 ]
=====
Functionality vs. Num. methods , files_count_total ,
avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.333963e+00 4.626388e-01 2.883379 0.003934334
x1         -2.755409e-05 9.604289e-06 -2.868936 0.004118550
x2          1.907948e-04 7.357363e-05 2.593250 0.009507367
x3         -1.673926e-01 5.448983e-02 -3.071997 0.002126317
R2log = 0.9291725
Excluded as outliers: Findbugs Eclipse Struts ( 3 / 13 )
MMRE = 16.16133
Pred(25) = 84.61538
Error range = [ -60.76103 .. 58.34261 ]
=====
Functionality vs. Num. methods per class , RFC ,
avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.169164e-02 3.358231e-01 0.1539252 0.8776687103
x1         -1.085938e-01 3.671259e-02 -2.9579440 0.0030969837
x2          6.573557e-05 1.884409e-05 3.4883917 0.0004859356
x3          1.079603e+00 3.750030e-01 2.8789197 0.0039903994
R2log = 0.9092877
Excluded as outliers: Hibernate Log4J Eclipse ( 3 / 13 )
MMRE = 16.99585
Pred(25) = 76.92308
Error range = [ -53.24238 .. 66.22875 ]
=====
```

Quite interestingly the degree of satisfaction of functionality seems to be correlated to:

- The number of interfaces per class: few interfaces favour the focusing on user relevant functionalities;
- The number of parameters per method: several parameters allow expressing more precisely and flexibly the functionality that methods have to provide.

- Quite interestingly, some other methods seem to indicate that a higher McCabe's complexity –which is generally considered negative– characterizes more functionally satisfactory products.

## 2.5 Interoperability

The significant models found for OSS product Interoperability are reported below.

```
=====
Interoperability      vs.      Comment      lines      per      class,
avg_minor_release_per_year
              Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.94854075 0.39812367 -2.382528 0.01719423
x1          0.02185037 0.00902074  2.422236 0.01542531
x2          0.09648049 0.04114832  2.344701 0.01904236
R2log = 0.94568
Excluded as outliers: JFreeChart Xerces Hibernate ( 3 / 13 )
MMRE = 22.53636
Pred(25) = 69.23077
Error range = [ -27.87267 .. 85.69022 ]
=====

Interoperability      vs.      Comment      lines      per      class,
avg_files_rem_per_year
              Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.952668736 0.3761882495 -2.532426 0.011327642
x1          0.020268193 0.0081942844  2.473455 0.013381369
x2          0.001820175 0.0006566018  2.772113 0.005569364
R2log = 0.9243362
Excluded as outliers: Eclipse Saxon JFreeChart Xerces ( 4 / 13 )
MMRE = 25.49909
Pred(25) = 61.53846
Error range = [ -11.08827 .. 79.64887 ]
=====

Interoperability vs. NOC , Num. methods per interface
              Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.2769657 0.5250736 -2.431975 0.01501675
x1          0.8443261 0.3777582  2.235097 0.02541099
x2          0.2734487 0.1135832  2.407474 0.01606331
R2log = 0.9128017
Excluded as outliers: Eclipse Log4J ( 2 / 14 )
MMRE = 16.93735
Pred(25) = 78.57143
Error range = [ -29.80316 .. 73.30486 ]
=====

Interoperability vs. NOC , avg_files_added_per_year
              Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.233395138 0.4512595445 -2.733228 0.006271694
x1          0.886416103 0.3782517281  2.343456 0.019106040
x2          0.001204708 0.0004386329  2.746505 0.006023391
R2log = 0.883836
Excluded as outliers: Eclipse Hibernate JBoss Log4J ( 4 / 13 )
MMRE = 20.63851
Pred(25) = 61.53846
Error range = [ -30.20372 .. 69.15352 ]
```

```
=====
Interoperability      vs.      Num.      attributes      per      class      ,
avg_files_rem_per_year
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.966482675 0.392668022 -2.461323 0.013842584
x1          0.065639529 0.029437625  2.229783 0.025761822
x2          0.001794850 0.000635083  2.826166 0.004710879
R2log = 0.9242188
Excluded as outliers: Eclipse Saxon Xalan ( 3 / 13 )
MMRE = 16.10620
Pred(25) = 61.53846
Error range = [ -11.97632 .. 50.04999 ]
=====

Interoperability      vs.      Num.      attributes      per      class      ,
avg_number_of_revisions_per_file
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 0.37416975 0.22485231  1.664069 0.096098649
x1          0.03165887 0.01425633  2.220688 0.026372087
x2          -0.09900444 0.03815263 -2.594957 0.009460265
R2log = 0.9441634
Excluded as outliers: ( 0 / 13 )
MMRE = 9.34896
Pred(25) = 92.3077
Error range = [ -23.23866 .. 36.90387 ]
=====

Interoperability vs. Num. packages , avg_files_added_per_year
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.8735346284 0.3563265128 -2.451501 0.01422620
x1          0.0101662849 0.0046845465  2.170175 0.02999360
x2          0.0009092894 0.0003858134  2.356811 0.01843260
R2log = 0.8964629
Excluded as outliers: Eclipse Hibernate Log4J ( 3 / 13 )
MMRE = 24.91969
Pred(25) = 53.84615
Error range = [ -31.39552 .. 91.39903 ]
=====

Interoperability      vs.      number_of_developers      ,
avg_number_of_revisions_per_file
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 0.33871179 0.32333420  1.047559 0.294841634
x1          0.02557603 0.01261630  2.027222 0.042639732
x2          -0.10740966 0.04005098 -2.681823 0.007322211
R2log = 0.9125277
Excluded as outliers: JBoss Eclipse ( 2 / 13 )
MMRE = 11.20099
Pred(25) = 92.3077
Error range = [ -19.32354 .. 34.55965 ]
=====

Interoperability      vs.      avg_loc_changed_per_year      ,
avg_minor_release_per_year
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -7.255751e-01 2.919291e-01 -2.485450 0.01293879
x1          1.837159e-05 8.863449e-06  2.072736 0.03819688
x2          1.196343e-01 4.944316e-02  2.419633 0.01553619
R2log = 0.880416
Excluded as outliers: Eclipse Log4J JBoss Hibernate ( 4 / 13 )
=====
```

```

MMRE = 19.82313
Pred(25) = 76.92308
Error range = [ -37.60926 .. 63.93443 ]
=====
Interoperability vs. LCOM , Num. attributes per class ,
avg_file_size
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.784203149 0.5943773606 -3.001802 0.002683867
x1          -0.002934514 0.0009850208 -2.979139 0.002890595
x2           0.280814462 0.0904945997  3.103107 0.001915001
x3           0.014072216 0.0050155709  2.805706 0.005020652
R2log = 0.9548116
Excluded as outliers: PMD Xalan JBoss Saxon ( 4 / 13 )
MMRE = 23.78304
Pred(25) = 69.23077
Error range = [ -97.91362 .. 54.8396 ]
=====
Interoperability vs. Comment_lines , Num. methods per interface ,
avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.731700e+00 5.746545e-01 -3.013463 0.002582848
x1           1.371726e-05 5.271175e-06  2.602316 0.009259661
x2            3.424600e-01 1.272790e-01  2.690625 0.007131827
x3           1.400984e-05 5.272336e-06  2.657236 0.007878436
R2log = 0.9023645
Excluded as outliers: Eclipse Log4J Hibernate ( 3 / 12 )
MMRE = 23.1591
Pred(25) = 75
Error range = [ -64.09418 .. 94.80483 ]
=====
Interoperability vs. eLOC , McCabe , Num. parameters per method
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.852608e+00 1.103145e+00  1.679387 0.093076714
x1          -1.239671e-05 4.009718e-06 -3.091666 0.001990368
x2            1.750873e+00 5.429769e-01  3.224582 0.001261569
x3           -5.757139e+00 2.033498e+00 -2.831150 0.004638095
R2log = 0.9242631
Excluded as outliers: Hibernate Eclipse JFreeChart Xalan ( 4 / 16 )
MMRE = 23.44430
Pred(25) = 68.75
Error range = [ -89.31794 .. 27.20029 ]
=====
Interoperability vs. eLOC , Num. methods per interface ,
avg_loc_del_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.835106e+00 6.097241e-01 -3.009732 0.002614786
x1           1.585528e-05 6.100226e-06  2.599130 0.009346050
x2            3.123285e-01 1.209456e-01  2.582388 0.009811921
x3           1.348695e-05 5.199549e-06  2.593869 0.009490258
R2log = 0.9027649
Excluded as outliers: Eclipse Log4J Hibernate ( 3 / 12 )
MMRE = 23.67843
Pred(25) = 75
Error range = [ -66.95667 .. 95.03747 ]
=====
```

## 2.6 Security

The significant models found for OSS product security are reported below.

```
=====
Security vs. Num. parameters per method
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.9851316  0.4394675 -2.241648 0.02498410
x1          1.1150867  0.5064968  2.201567 0.02769589
R2log = 0.9332337
Excluded as outliers: ( 0 / 16 )
MMRE = 12.6%
Pred(25) = 93.75
Error range = [ -20.41925 .. 21.6 ]
=====
```

Only one statistically significant correlation involving Security was found. This seems to confirm that security is difficult to evaluate on the basis of internal characteristics; in particular, measures that indicate a good design are not able to support the perception of a good security level.

The only correlation found says that the applications that are considered more secure by users have methods that have been well specified as far as I/O parameters are concerned. This is quite credible.

## 2.7 Performance (in terms of Speed)

The significant models found for OSS product speed are reported below.

```
=====
Speed vs. LCOM
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.009537732 0.14262804  0.06687136 0.94668411
x1         -0.000395338 0.00019667 -2.01015888 0.04441438
R2log = 0.9221913
Excluded as outliers: Eclipse Hibernate PMD ( 3 / 16 )
MMRE = 18.64037
Pred(25) = 81.25
Error range = [ -34.44296 .. 113.2987 ]
=====

Speed vs. eLOC
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -4.644805e-02 1.261658e-01 -0.3681509 0.71276072
x1          -3.292392e-06 1.503309e-06 -2.1900960 0.02851728
R2log = 0.9209137
Excluded as outliers: Eclipse ( 1 / 16 )
MMRE = 15.74848
Pred(25) = 75
Error range = [ -26.16792 .. 62.69641 ]
=====

Speed vs. Num. interfaces
```

```

          Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.100310576 0.159169346 0.6302129 0.52855530
x1         -0.003023446 0.001381866 -2.1879442 0.02867366
R2log = 0.9157114
Excluded as outliers: Eclipse JBoss Hibernate ( 3 / 16 )
MMRE = 15.57327
Pred(25) = 87.5
Error range = [ -43.84856 .. 22.57831 ]
=====
Speed vs. Num. public methods
          Estimate Std. Error   z value Pr(>|z|)
(Intercept) 8.736932e-02 1.722447e-01 0.5072396 0.61198672
x1        -3.151493e-05 1.601829e-05 -1.9674346 0.04913313
R2log = 0.9189843
Excluded as outliers: Hibernate Eclipse ( 2 / 15 )
MMRE = 18.48592
Pred(25) = 86.66667
Error range = [ -31.41413 .. 83.51267 ]
=====
Speed vs. avg_loc_changed_per_year
          Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.967090e-01 1.061988e-01 -1.852272 6.398674e-02
x1        -8.954855e-06 1.594165e-06 -5.617272 1.939963e-08
R2log = 0.9443573
Excluded as outliers: ( 0 / 13 )
MMRE = 9.619438
Pred(25) = 100
Error range = [ -20.70618 .. 24.57343 ]
=====
Speed vs. avg_major_release_per_year
          Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.5562534 0.1412891 -3.936987 8.251118e-05
x1        0.5149317 0.2507610  2.053476 4.002639e-02
R2log = 0.9229947
Excluded as outliers: Eclipse ( 1 / 13 )
MMRE = 13.74866
Pred(25) = 92.3077
Error range = [ -14.82389 .. 56.59987 ]
=====
Speed vs. ECAtot
          Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.1121920267 1.273292e-01 -0.881118 0.37825397
x1        -0.0001123617 5.494527e-05 -2.044975 0.04085728
R2log = 0.915779
Excluded as outliers: Eclipse ( 1 / 14 )
MMRE = 16.68673
Pred(25) = 78.57143
Error range = [ -25.95910 .. 72.2058 ]
=====
Speed vs. ECAcritical
          Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.034002061 0.1412762444 -0.2406778 0.80980482
x1        -0.001257149 0.0005228072 -2.4046133 0.01618959
R2log = 0.918071
Excluded as outliers: Eclipse ( 1 / 14 )
MMRE = 15.62997

```

```

Pred(25) = 78.57143
Error range = [ -23.65276 .. 73.51257 ]
=====
Speed vs. ECAveryCritical
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.057662307 0.143083582 -0.4029974 0.68695016
x1          -0.002091511 0.000973537 -2.1483631 0.03168492
R2log = 0.9163461
Excluded as outliers: Eclipse (1 / 14)
MMRE = 17.2565
Pred(25) = 78.57143
Error range = [ -25.17004 .. 82.84554 ]
=====
Speed vs. LCOM , Num. abstract classes
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.2571669343 0.2020958999 1.272500 0.20319570
x1          -0.0004935335 0.0002169290 -2.275092 0.02290042
x2          -0.0039756765 0.0015553041 -2.556205 0.01058207
R2log = 0.929076
Excluded as outliers: Eclipse PMD (2 / 15)
MMRE = 17.16227
Pred(25) = 86.66667
Error range = [ -38.8406 .. 105.9486 ]
=====
Speed vs. Num. abstract classes , Num. methods per class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.528893345 0.328382974 1.610599 0.107267115
x1          -0.004130625 0.001594311 -2.590853 0.009573826
x2          -0.033852840 0.016929267 -1.999664 0.045536589
R2log = 0.9247591
Excluded as outliers: Eclipse (1 / 15)
MMRE = 17.91647
Pred(25) = 93.33333
Error range = [ -21.58915 .. 108.9888 ]
=====
```

All the statistically significant correlations found (except one) are characterized by negative coefficients, i.e., they indicate the factors that decrease the efficiency of the analyzed applications. Among these factors are:

- Lack of Cohesion
- Application size
- The number of interfaces
- The number of interfaces or public methods per class
- The quality of the code

The correlation characterized by the better precision indicates that efficiency is most hindered by the modification of large amounts of code. This is quite credible: maintenance activities often disrupt the efficiency of programs, because maintainers do not have a precise perception of the rationale and mechanisms that guarantee the efficiency of the program and that were conceived by the original developers.

## 2.8 Usefulness of the product developer community

The significant models found for the usefulness of the OSS product developer community are reported below.

```
=====
Community vs. Comment_lines
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -6.107549e-01 1.592258e-01 -3.835778 0.0001251674
x1          3.555450e-06 1.580453e-06  2.249640 0.0244717891
R2log = 0.9441336
Excluded as outliers: struts JBoss ( 2 / 16 )
MMRE = 13.47508
Pred(25) = 81.25
Error range = [ -39.84053 .. 33.93528 ]
=====
Community vs. eLOC
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -6.064049e-01 1.551502e-01 -3.908503 9.286997e-05
x1          3.124606e-06 1.362489e-06  2.293307 2.183034e-02
R2log = 0.944317
Excluded as outliers: struts JBoss ( 2 / 16 )
MMRE = 13.02088
Pred(25) = 81.25
Error range = [ -39.68846 .. 34.65983 ]
=====
Community vs. NOC
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.0721555 0.3467189 -3.092290 0.001986184
x1          0.6980704 0.3032436  2.302012 0.021334512
R2log = 0.9444442
Excluded as outliers: struts JBoss ( 2 / 16 )
MMRE = 13.64843
Pred(25) = 81.25
Error range = [ -56.43863 .. 27.32624 ]
=====
Community vs. Num. interfaces
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.572017292 0.1449517171 -3.946261 7.938112e-05
x1          0.001291589 0.0005486797  2.353995 1.857287e-02
R2log = 0.9428153
Excluded as outliers: struts Hibernate ( 2 / 16 )
MMRE = 14.08051
Pred(25) = 81.25
Error range = [ -38.31731 .. 38.79392 ]
=====
Community vs. avg_minor_release_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.6651097 0.17917467 -3.712075 0.0002055673
x1          0.1575647 0.06016305  2.618962 0.0088197758
R2log = 0.9201169
Excluded as outliers: Eclipse PMD ( 2 / 13 )
MMRE = 18.89634
```

```

Pred(25) = 84.61538
Error range = [ -24.08803 .. 75.3877 ]
=====
Community vs. avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.7539948850 0.1863750661 -4.045578 5.219416e-05
x1          0.0009194239 0.0003408322  2.697585 6.984438e-03
R2log = 0.905083
Excluded as outliers: Eclipse Hibernate JBoss ( 3 / 13 )
MMRE = 16.63442
Pred(25) = 84.61538
Error range = [ -22.59236 .. 41.23304 ]
=====
Community vs. avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.620325331 0.1726199537 -3.593590 0.000326153
x1          0.001028721 0.0004764248  2.159251 0.030830694
R2log = 0.9102702
Excluded as outliers: Eclipse JBoss ( 2 / 13 )
MMRE = 15.91004
Pred(25) = 84.61538
Error range = [ -26.31858 .. 30.09996 ]
=====
Community vs. avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.22556328 0.21868229  1.031466 0.30232247
x1          -0.07726632 0.03560442 -2.170133 0.02999678
R2log = 0.9416892
Excluded as outliers: ( 0 / 13 )
MMRE = 11.36680
Pred(25) = 92.3077
Error range = [ -24.70167 .. 25.28183 ]
=====
Community vs. McCabe , number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.14068354 0.37999532 -0.3702244 0.711215296
x1          -0.66074402 0.27967275 -2.3625613 0.018149139
x2          0.06185318 0.02329124  2.6556411 0.007915781
R2log = 0.9497033
Excluded as outliers: JBoss Saxon JFreeChart ( 3 / 13 )
MMRE = 20.63438
Pred(25) = 69.23077
Error range = [ -72.15354 .. 47.90988 ]
=====
Community vs. Num. packages , avg_minor_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.769124159 0.2123371311 -3.622184 0.0002921262
x1          0.002149763 0.0008957929  2.399844 0.0164020531
x2          0.158006584 0.0599756092  2.634514 0.0084257822
R2log = 0.9452808
Excluded as outliers: Hibernate PMD ( 2 / 13 )
MMRE = 20.8996
Pred(25) = 76.92308
Error range = [ -18.46425 .. 77.69021 ]
=====
Community vs. number_of_developers , avg_files_rem_per_year

```

```

      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.9083582048 0.2600482183 -3.493038 0.0004775593
x1          0.0181644658 0.0083008096  2.188276 0.0286494751
x2          0.0009104597 0.0004575253  1.989966 0.0465946924
R2log = 0.9188059
Excluded as outliers: Eclipse (1 / 13)
MMRE = 11.48175
Pred(25) = 84.61538
Error range = [ -18.34736 .. 31.71261 ]
=====
```

Some of the models of the quality of developers communities are quite enlightening: the utility of the community appears to be proportional to:

- The number of comment lines: good communities properly comment the code, so that sharing the knowledge about code becomes easier.
- The product size (in effective LOC): larger products call for organized, effective communities.
- The generalization hierarchy: good communities organize well the code in terms of inheritance relations.
- The number of interfaces: products with many interfaces call for organized, effective communities.
- The amount of process activities (represented by the numbers of minor releases per year, files added per year, files removed per year).
- The maturity of the product (represented by few revisions per file, which means also that most file require little or no modifications).

Another interesting model is the one that indicates that many developers that succeed in keeping McCabe's complexity low form an effective community.

## 2.9 Documentation Quality

The significant models found for OSS product documentation are reported below.

```

=====
DocQuality vs. avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.971990895 0.1950676328 -4.982840 6.265761e-07
x1          0.001388026 0.0003463704  4.007346 6.140485e-05
R2log = 0.9039233
Excluded as outliers: Eclipse Hibernate Log4J (3 / 13)
MMRE = 26.67465
Pred(25) = 69.23077
Error range = [ -28.87373 .. 82.9359 ]
=====
```

```

=====
DocQuality vs. McCabe , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.906267144 0.5905146323  1.534707 0.124855731
x1          -0.611268853 0.2616864731 -2.335883 0.019497366
x2          0.001231225 0.0004604331  2.674059 0.007493935
=====
```

```
R2log = 0.9213972
Excluded as outliers: Eclipse Hibernate HttpUnit ( 3 / 13 )
MMRE = 32.21578
Pred(25) = 76.92308
Error range = [ -8.303711 .. 220.5464 ]
=====
DocQuality vs. Num. methods , avg_major_release_per_year
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.827327e+00 5.339879e-01 -3.422039 0.0006215339
x1          6.858379e-05 2.860584e-05  2.397545 0.0165053457
x2          7.661133e-01 3.767803e-01  2.033316 0.0420206198
R2log = 0.9049725
Excluded as outliers: Hibernate Struts Log4J Eclipse ( 4 / 13 )
MMRE = 23.72513
Pred(25) = 61.53846
Error range = [ -75.2069 .. 74.77335 ]
=====
DocQuality vs. Num. methods per interface , avg_files_rem_per_year
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.355179527 0.1614658018 -2.199720 0.027826779
x1          -0.411938635 0.1446969637 -2.846906 0.004414641
x2          0.002390961 0.0008086415  2.956762 0.003108879
R2log = 0.9505327
Excluded as outliers: JBoss Saxon ( 2 / 12 )
MMRE = 21.46283
Pred(25) = 66.66667
Error range = [ -64.70189 .. 62.2226 ]
=====
DocQuality vs. Num. public methods , avg_major_release_per_year
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.737036e+00 4.983675e-01 -3.485452 0.0004913072
x1          7.329941e-05 3.067189e-05  2.389791 0.0168579763
x2          8.063410e-01 3.833795e-01  2.103245 0.0354443690
R2log = 0.9045322
Excluded as outliers: Eclipse Log4J Hibernate ( 3 / 12 )
MMRE = 19.49775
Pred(25) = 75
Error range = [ -40.20908 .. 69.13603 ]
=====
DocQuality vs. RFC , avg_major_release_per_year
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.864070e+00 5.414701e-01 -3.442610 0.0005761305
x1          3.737690e-05 1.534821e-05  2.435262 0.0148810235
x2          8.660546e-01 3.955164e-01  2.189681 0.0285473916
R2log = 0.9055634
Excluded as outliers: Hibernate Struts Eclipse Log4J ( 4 / 13 )
MMRE = 23.37525
Pred(25) = 61.53846
Error range = [ -75.44821 .. 73.81566 ]
=====
DocQuality           vs.           avg_loc_del_per_year      ,
avg_number_of_revisions_per_file
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.977328e-01 3.610768e-01  1.655417 0.09783987
x1         -2.833119e-06 1.260732e-06 -2.247201 0.02462718
x2         -1.006794e-01 4.749546e-02 -2.119769 0.03402555
```

```
R2log = 0.9451484
Excluded as outliers: JBoss HttpUnit ( 2 / 13 )
MMRE = 20.71966
Pred(25) = 69.23077
Error range = [ -25.95749 .. 133.3430 ]
=====
DocQuality vs. avg_files_rem_per_year , DistinctLicenses
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.159241499 0.23813028 -0.6687159 0.5036767378
x1          0.002193904 0.00058773  3.7328443 0.0001893296
x2          -0.292768856 0.10818060 -2.7062971 0.0068038148
R2log = 0.914773
Excluded as outliers: Eclipse PMD JBoss ( 3 / 13 )
MMRE = 25.71771
Pred(25) = 69.23077
Error range = [ -25.88590 .. 110.0218 ]
=====
DocQuality vs. avg_files_rem_per_year , GlobalConflits
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.545872893 0.1464844531 -3.726490 0.0001941645
x1          0.001857884 0.0005258968  3.532792 0.0004111957
x2          -0.172469898 0.0750569425 -2.297854 0.0215700986
R2log = 0.9173308
Excluded as outliers: Eclipse PMD ( 2 / 13 )
MMRE = 25.41794
Pred(25) = 69.23077
Error range = [ -25.9635 .. 139.1082 ]
=====
DocQuality           vs.           avg_number_of_revisions_per_file , 
ReferenceConflicts
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.55637478 0.324682476  1.713597 0.08660284
x1          -0.09315608 0.044485565 -2.094074 0.03625336
x2          -0.01305393 0.006491948 -2.010788 0.04434784
R2log = 0.9162061
Excluded as outliers: Eclipse HttpUnit JBoss ( 3 / 13 )
MMRE = 21.80449
Pred(25) = 76.92308
Error range = [ -24.17132 .. 140.8625 ]
=====
DocQuality           vs.           avg_number_of_revisions_per_file , 
UncertainLicensedFiles
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.701472700 0.5161775062  3.296294 0.0009796951
x1          -0.271129653 0.0718459870 -3.773762 0.0001608042
x2          -0.001117045 0.0004104269 -2.721667 0.0064953621
R2log = 0.916066
Excluded as outliers: Eclipse Hibernate Findbugs ( 3 / 13 )
MMRE = 21.51298
Pred(25) = 76.92308
Error range = [ -61.86057 .. 41.89651 ]
=====
DocQuality           vs.           LCOM ,           number_of_commits , 
avg_number_of_revisions_per_file
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.8904641341 0.5089468024  3.714463 2.036356e-04
```

```

x1      -0.0013645103 0.0004219224 -3.234031 1.220561e-03
x2      0.0003436489 0.0001048866 3.276386 1.051449e-03
x3      -0.3240773825 0.0791537252 -4.094278 4.234853e-05
R2log = 0.9251528
Excluded as outliers: Eclipse JFreeChart Hibernate ( 3 / 13 )
MMRE = 19.55496
Pred(25) = 76.92308
Error range = [ -76.65345 .. 78.69672 ]
=====
DocQuality vs. McCabe , number_of_developers ,
avg_major_release_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.34058074 0.41768084 0.815409 0.4148381954
x1      -1.34911047 0.37827426 -3.566488 0.0003617978
x2      0.09196995 0.02701623 3.404248 0.0006634657
x3      1.14739506 0.32913651 3.486077 0.0004901607
R2log = 0.9498544
Excluded as outliers: JBoss Saxon JFreeChart ( 3 / 13 )
MMRE = 30.86441
Pred(25) = 61.53846
Error range = [ -91.5906 .. 123.7628 ]
=====
DocQuality vs. Num. attributes per class , avg_loc_changed_per_year
, avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.475552372 3.641790e-01 -4.051731 5.084012e-05
x1      0.0681825224 2.277817e-02 2.993327 2.759539e-03
x2      -0.0000245755 7.263675e-06 -3.383342 7.160931e-04
x3      0.0015591900 3.853879e-04 4.045768 5.215182e-05
R2log = 0.9221577
Excluded as outliers: Eclipse ( 1 / 13 )
MMRE = 15.97876
Pred(25) = 76.92308
Error range = [ -55.66203 .. 43.04478 ]
=====
DocQuality vs. number_of_commits , avg_file_size ,
avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.9620604834 0.3931890634 -2.446814 0.014412522
x1      -0.0005881969 0.0001920358 -3.062955 0.002191633
x2      -0.0066717596 0.0022714775 -2.937189 0.003312017
x3      0.4652761030 0.1682858065 2.764797 0.005695822
R2log = 0.9520892
Excluded as outliers: HttpUnit PMD ( 2 / 13 )
MMRE = 52.9687
Pred(25) = 84.61538
Error range = [ -68.04876 .. 538.0575 ]
=====
```

Some of the models reported above indicate a quite understandable influence of process and product attributes on the documentation quality. For instance, the model that involves McCabe's complexity and the number of files removed per year says that documentation is generally better if products are simpler and they are further simplified by removing files (which is very easy to believe!).

## 2.10 Trustworthiness with respect to non Open Source (closed source) products

The significant models found for trustworthiness of OSS products with respect to closed-source competitors are reported below.

```
=====
CssCompetitors vs. Num. methods per interface
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.01937423  0.2880372 -0.06726294 0.94637238
x1          0.32132320  0.1616961  1.98720386 0.04689981
R2log = 0.8845989
Excluded as outliers: Ant Log4J Eclipse JBoss ( 4 / 14 )
MMRE = 13.21328
Pred(25) = 92.85714
Error range = [ -23.82053 .. 27.60377 ]
=====

CssCompetitors vs. Num. public methods
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.455624e+00 3.080639e-01 4.725071 2.300346e-06
x1          -6.631991e-05 2.552896e-05 -2.597830 9.381483e-03
R2log = 0.940286
Excluded as outliers: Hibernate Httpunit Log4J Saxon ( 4 / 15 )
MMRE = 17.98912
Pred(25) = 73.33333
Error range = [ -57.06001 .. 56.36143 ]
=====

CssCompetitors vs. avg_loc_added_per_year
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 7.099362e-01 1.386742e-01 5.119455 3.064204e-07
x1          -1.078314e-05 5.188498e-06 -2.078278 3.768377e-02
R2log = 0.909512
Excluded as outliers: Eclipse Hibernate PMD ( 3 / 13 )
MMRE = 21.55156
Pred(25) = 69.23077
Error range = [ -85.30261 .. 31.53923 ]
=====

CssCompetitors vs. eLOC per class , Num. interfaces per class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.32393462 0.422340469 0.7669988 0.443082250
x1          0.02029101 0.008203071 2.4735875 0.013376404
x2          -2.47936110 0.763618664 -3.2468576 0.001166867
R2log = 0.9431205
Excluded as outliers: JFreeChart Struts ( 2 / 16 )
MMRE = 11.17303
Pred(25) = 87.5
Error range = [ -67.74832 .. 20.91101 ]
=====

CssCompetitors vs. eLOC , Num. attributes per class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.868297e+00 4.352697e-01 4.292275 1.768518e-05
x1          -5.424478e-06 2.162070e-06 -2.508928 1.210982e-02
x2          -8.477650e-02 3.129539e-02 -2.708914 6.750387e-03
R2log = 0.9436018
Excluded as outliers: xalan Struts HttpUnit ( 3 / 16 )
```

```

MMRE = 17.53337
Pred(25) = 81.25
Error range = [ -79.55742 .. 62.16023 ]
=====
CssCompetitors vs. Num. abstract classes , avg_loc_changed_per_year
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.229616e-01 1.506087e-01 3.472319 0.0005159834
x1          1.011691e-02 4.843799e-03 2.088631 0.0367409083
x2          -3.953368e-05 1.747116e-05 -2.262796 0.0236482701
R2log = 0.9162679
Excluded as outliers: Eclipse Xalan ( 2 / 12 )
MMRE = 21.25506
Pred(25) = 75
Error range = [ -93.25162 .. 34.44312 ]
=====
CssCompetitors vs. Num. attributes per class , Num. interfaces per
class
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8268770 0.31288172 2.642778 0.00822289
x1          0.0922271 0.04592669 2.008137 0.04462869
x2          -1.8489955 0.81092630 -2.280103 0.02260158
R2log = 0.9136195
Excluded as outliers: JFreeChart JMeter Xalan Log4J Eclipse ( 5 / 16
)
MMRE = 17.39607
Pred(25) = 75
Error range = [ -65.57258 .. 54.55013 ]
=====
CssCompetitors vs. Num. attributes per class , Num. public methods
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.721397e+00 3.782676e-01 4.550739 5.345785e-06
x1          -7.398270e-02 2.913191e-02 -2.539576 1.109870e-02
x2          -2.274836e-05 9.381677e-06 -2.424765 1.531829e-02
R2log = 0.9203068
Excluded as outliers: xalan Eclipse Httpunit ( 3 / 15 )
MMRE = 15.33009
Pred(25) = 86.66667
Error range = [ -67.42814 .. 59.25337 ]
=====
CssCompetitors vs. avg_loc_del_per_year , avg_files_added_per_year
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.691809e-01 1.596831e-01 2.311960 0.02077991
x1          -8.480383e-06 3.593819e-06 -2.359714 0.01828904
x2          1.092654e-03 4.454859e-04 2.452724 0.01417790
R2log = 0.943119
Excluded as outliers: struts ( 1 / 13 )
MMRE = 9.655532
Pred(25) = 92.3077
Error range = [ -13.12781 .. 37.83713 ]
=====
```

## 2.11 Trustworthiness with respect to Open Source products

The significant models found for trustworthiness of OSS products with respect to other open source competitors are reported below.

```
=====
OssCompetitors vs. NOC
    Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1353703 0.2558067 0.5291897 0.59667386
x1          0.5464770 0.2676106 2.0420602 0.04114556
R2log = 0.9137373
Excluded as outliers: struts Eclipse ( 2 / 16 )
MMRE = 10.63992
Pred(25) = 87.5
Error range = [ -27.67993 .. 39.60142 ]
=====

OssCompetitors vs. avg_files_added_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.2986039291 0.1724483862 1.731555 0.08335276
x1          0.0007219583 0.0003586082 2.013223 0.04409116
R2log = 0.9075203
Excluded as outliers: Eclipse Hibernate ( 2 / 13 )
MMRE = 11.56205
Pred(25) = 92.3077
Error range = [ -16.56029 .. 31.39113 ]
=====

OssCompetitors vs. avg_files_rem_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.3667191884 0.1449333573 2.530261 0.01139778
x1          0.0008974991 0.0004468773 2.008379 0.04460300
R2log = 0.915074
Excluded as outliers: Eclipse ( 1 / 13 )
MMRE = 9.65309
Pred(25) = 92.3077
Error range = [ -15.99937 .. 34.36061 ]
=====

OssCompetitors vs. Comment lines per class ,
avg_files_added_per_year
    Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.3065353449 0.3412067533 -0.898386 0.36897984
x1          0.0134078953 0.0065998498 2.031546 0.04219969
x2          0.0009208207 0.0003732333 2.467145 0.01361951
R2log = 0.9174157
Excluded as outliers: Eclipse JFreeChart Hibernate ( 3 / 13 )
MMRE = 10.50353
Pred(25) = 84.61538
Error range = [ -7.918586 .. 32.89196 ]
=====

OssCompetitors vs. eLOC per class , Num. interfaces per class
    Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.07551386 0.414313541 0.1822626 0.85537665
x1          0.01848722 0.007910356 2.3370903 0.01943449
x2          -1.73133253 0.726189317 -2.3841339 0.01711937
R2log = 0.9420207
Excluded as outliers: struts JFreeChart ( 2 / 16 )
=====
```

```

MMRE = 11.70374
Pred(25) = 81.25
Error range = [ -47.90818 .. 33.19902 ]
=====
OssCompetitors vs. eLOC per class , avg_major_release_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.82150137 0.602505873 -1.363474 0.17273303
x1          0.02143296 0.009085452  2.359041 0.01832221
x2         -1.98986228 1.005809803 -1.978368 0.04788717
R2log = 0.946567
Excluded as outliers: struts Log4J PMD JBoss ( 4 / 13 )
MMRE = 21.87424
Pred(25) = 69.23077
Error range = [ -76.65577 .. 12.6339 ]
=====
OssCompetitors vs. eLOC per class , avg_minor_release_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.83269504 0.586398029 -1.420017 0.155602796
x1          0.02225704 0.008625197  2.580467 0.009866678
x2         -0.16151870 0.077885689 -2.073792 0.038098651
R2log = 0.9517325
Excluded as outliers: struts PMD Hibernate JFreeChart ( 4 / 13 )
MMRE = 18.05857
Pred(25) = 76.92308
Error range = [ -67.5641 .. 35.07406 ]
=====
OssCompetitors vs. avg_loc_del_per_year , avg_files_added_per_year
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 3.742391e-01 1.482118e-01  2.525028 0.01156890
x1          -5.063380e-06 2.493110e-06 -2.030949 0.04226020
x2           7.147071e-04 3.155361e-04  2.265056 0.02350922
R2log = 0.9422954
Excluded as outliers: ( 0 / 13 )
MMRE = 7.04258
Pred(25) = 92.3077
Error range = [ -14.46197 .. 35.19925 ]
=====
OssCompetitors vs. avg_file_size , LicensedFiles
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.662479443 0.1848145237  3.584564 0.0003376421
x1          0.006671308 0.0033874786  1.969402 0.0489069013
x2         -0.001005512 0.0004765122 -2.110150 0.0348454532
R2log = 0.9013024
Excluded as outliers: Eclipse JBoss Hibernate JFreeChart ( 4 / 13 )
MMRE = 18.29131
Pred(25) = 69.23077
Error range = [ -50.37394 .. 43.59771 ]
=====
```

## 2.12 Trustworthiness

The significant models found for OSS product trustworthiness are reported below.

```
=====
Trustworthiness vs. LCOM
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.9449059604 0.1558679970 6.062219 1.342561e-09
x1          -0.0004187303 0.0001849532 -2.263981 2.357531e-02
R2log = 0.9118105
Excluded as outliers: Eclipse HttpUnit Hibernate (3 / 16)
MMRE = 14.69376
Pred(25) = 93.75
Error range = [ -19.97479 .. 67.77752 ]
=====

Trustworthiness vs. Num. interfaces
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8362203859 0.1284887587 6.508121 7.609663e-11
x1          -0.0009178223 0.0004346276 -2.111744 3.470841e-02
R2log = 0.9401851
Excluded as outliers: HttpUnit (1 / 16)
MMRE = 13.95487
Pred(25) = 87.5
Error range = [ -20.14775 .. 62.92066 ]
=====

Trustworthiness vs. Num. methods
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.020510e+00 1.797032e-01 5.678864 1.355922e-08
x1          -3.341328e-05 1.336678e-05 -2.499725 1.242898e-02
R2log = 0.943928
Excluded as outliers: Hibernate Httpunit Saxon (3 / 16)
MMRE = 15.78368
Pred(25) = 81.25
Error range = [ -34.05307 .. 68.48912 ]
=====

Trustworthiness vs. Num. public methods
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.050003e+00 2.030121e-01 5.172122 2.314510e-07
x1          -4.291938e-05 1.789357e-05 -2.398592 1.645823e-02
R2log = 0.9450356
Excluded as outliers: Hibernate Httpunit Saxon (3 / 15)
MMRE = 17.58695
Pred(25) = 80
Error range = [ -39.23571 .. 69.6777 ]
=====

Trustworthiness vs. RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.959094e-01 1.757217e-01 5.667537 1.448650e-08
x1          -1.786360e-05 7.417221e-06 -2.408395 1.602285e-02
R2log = 0.9434857
Excluded as outliers: Hibernate Httpunit Saxon (3 / 16)
MMRE = 15.74339
Pred(25) = 81.25
Error range = [ -34.22964 .. 67.20574 ]
=====

Trustworthiness vs. avg_minor_release_per_year
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8336724 0.21092029 3.952547 7.732358e-05
```

```

x1      -0.1709522 0.08621789 -1.982793 4.739058e-02
R2log = 0.9199122
Excluded as outliers: Struts PMD Eclipse ( 3 / 13 )
MMRE = 16.02546
Pred(25) = 76.92308
Error range = [ -58.04918 .. 42.83300 ]
=====
Trustworthiness vs. avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0733039428 0.1853673583 0.3954523 0.69250914
x1          0.0008035347 0.0003490147 2.3022949 0.02131855
R2log = 0.9050056
Excluded as outliers: Log4J Eclipse Hibernate ( 3 / 13 )
MMRE = 12.44518
Pred(25) = 92.3077
Error range = [ -18.3153 .. 26.35232 ]
=====
Trustworthiness vs. ECAcritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.006571364 0.1811053039 5.557934 2.729862e-08
x1          -0.002302134 0.0008695641 -2.647458 8.109947e-03
R2log = 0.9108075
Excluded as outliers: HttpUnit Hibernate Eclipse ( 3 / 14 )
MMRE = 14.2219
Pred(25) = 85.71429
Error range = [ -26.01005 .. 69.84993 ]
=====
Trustworthiness vs. ECAeveryCritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.053430796 0.192135040 5.482763 4.187350e-08
x1          -0.004465151 0.001700510 -2.625772 8.645262e-03
R2log = 0.9431824
Excluded as outliers: Hibernate HttpUnit ( 2 / 14 )
MMRE = 14.09593
Pred(25) = 85.71429
Error range = [ -27.42494 .. 72.16417 ]
=====
Trustworthiness vs. LCOM , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.035289e+00 1.852369e-01 5.589000 2.283810e-08
x1          -3.524124e-04 1.628406e-04 -2.164156 3.045236e-02
x2          -3.009276e-06 1.390154e-06 -2.164707 3.041014e-02
R2log = 0.9418732
Excluded as outliers: HttpUnit ( 1 / 16 )
MMRE = 11.76897
Pred(25) = 93.75
Error range = [ -23.73681 .. 70.55883 ]
=====
Trustworthiness vs. eLOC per class , Num. interfaces per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.08170932 0.489735200 -0.1668439 0.867492901
x1          0.02197466 0.009287465 2.3660560 0.017978725
x2          -1.99998553 0.712936259 -2.8052796 0.005027295
R2log = 0.9420866
Excluded as outliers: JFreechart Struts Xerces ( 3 / 16 )
MMRE = 17.53712

```

```

Pred(25) = 68.75
Error range = [ -52.10994 .. 33.56057 ]
=====
Trustworthiness vs. eLOC per class , number_of_developers
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.54953810 0.88622430 2.876854 0.004016611
x1         -0.01528943 0.00778192 -1.964738 0.049444620
x2         -0.04821620 0.02186049 -2.205632 0.027409747
R2log = 0.9519285
Excluded as outliers: HttpUnit JBoss Xerces Hibernate ( 4 / 13 )
MMRE = 19.06270
Pred(25) = 76.92308
Error range = [ -44.01146 .. 99.52835 ]
=====
Trustworthiness vs. Num. attributes per class , Num. interfaces
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.126611075 0.197336499 5.709086 1.135844e-08
x1         -0.025715879 0.012991747 -1.979401 4.777086e-02
x2         -0.001225818 0.000465498 -2.633347 8.454787e-03
R2log = 0.9433258
Excluded as outliers: HttpUnit ( 1 / 16 )
MMRE = 12.60557
Pred(25) = 93.75
Error range = [ -20.08801 .. 71.42751 ]
=====
Trustworthiness vs. Num. attributes per class , avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.298652171 0.3409122878 -0.8760382 0.38100929
x1          0.061186256 0.0286748731 2.1337934 0.03285969
x2          0.001077984 0.0004794049 2.2485876 0.02453874
R2log = 0.9269831
Excluded as outliers: xalan Eclipse JMeter ( 3 / 13 )
MMRE = 13.35098
Pred(25) = 84.61538
Error range = [ -12.79747 .. 61.22068 ]
=====
Trustworthiness vs. Num. interfaces , avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.847074680 0.4334035749 4.261789 2.027967e-05
x1         -0.001410845 0.0005997616 -2.352344 1.865553e-02
x2         -0.170094011 0.0543869838 -3.127476 1.763139e-03
R2log = 0.9508081
Excluded as outliers: Findbugs ( 1 / 13 )
MMRE = 8.832315
Pred(25) = 92.3077
Error range = [ -48.9806 .. 15.42577 ]
=====
Trustworthiness vs. avg_loc_del_per_year , avg_number_of_revisions_per_file
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.879856e+00 4.385666e-01 4.286363 1.816225e-05
x1        -5.371639e-06 2.157395e-06 -2.489873 1.277888e-02
x2        -1.793606e-01 5.659590e-02 -3.169145 1.528882e-03
R2log = 0.9259456

```

```

Excluded as outliers: Eclipse Findbugs JBoss ( 3 / 13 )
MMRE = 9.319058
Pred(25) = 92.3077
Error range = [ -43.95124 .. 23.7274 ]
=====
Trustworthiness vs. avg_minor_release_per_year , 
avg_files_added_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.43508645 0.1874418039 2.321182 0.020277050
x1         -0.18035777 0.0779748736 -2.313024 0.020721310
x2          0.00141138 0.0004683939 3.013234 0.002584799
R2log = 0.9207203
Excluded as outliers: Eclipse Hibernate PMD ( 3 / 13 )
MMRE = 14.97507
Pred(25) = 76.92308
Error range = [ -36.13875 .. 44.93085 ]
=====
Trustworthiness vs. avg_minor_release_per_year , 
avg_files_rem_per_year
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.44920182 0.155388946 2.890822 0.003842354
x1         -0.15366367 0.062602885 -2.454578 0.014105002
x2          0.00255197 0.000909821 2.804914 0.005032995
R2log = 0.9227616
Excluded as outliers: Eclipse Hibernate Saxon ( 3 / 13 )
MMRE = 11.63267
Pred(25) = 92.3077
Error range = [ -6.907151 .. 41.66466 ]
=====
Trustworthiness vs. avg_number_of_revisions_per_file , 
ECACritical
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.064511031 0.470046005 4.392147 1.122369e-05
x1         -0.186129523 0.056968375 -3.267243 1.086004e-03
x2         -0.004765708 0.002111050 -2.257506 2.397646e-02
R2log = 0.9535993
Excluded as outliers: Findbugs Xerces Hibernate ( 3 / 13 )
MMRE = 10.97284
Pred(25) = 84.61538
Error range = [ -45.2087 .. 7.735594 ]
=====
Trustworthiness vs. Comment_lines , Num_attributes_per_class , 
Num_interfaces_per_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.364647e-02 3.588361e-01 0.09376558 0.925295376
x1         1.017502e-05 3.578772e-06 2.84315914 0.004466876
x2          1.307563e-01 4.294309e-02 3.04487320 0.002327784
x3         -2.877423e+00 8.840822e-01 -3.25470085 0.001135119
R2log = 0.9328287
Excluded as outliers: JFreeChart Xalan Struts Checkstyle Eclipse ( 5
/ 16 )
MMRE = 19.78407
Pred(25) = 75
Error range = [ -89.08603 .. 80.86492 ]
=====
Trustworthiness vs. eLOC_per_class , Num_interfaces_per_class , 
Num_parameters_per_method

```

```

      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 4.64758929 1.44738420  3.211027 0.0013226166
x1          0.04575355 0.01348791  3.392189 0.0006933665
x2         -5.31343849 1.45810400 -3.644074 0.0002683565
x3         -5.50426165 1.76048160 -3.126566 0.0017686089
R2log = 0.9224535
Excluded as outliers: JFreeChart Struts Eclipse Ant Findbugs ( 5 /
16 )
MMRE = 21.82098
Pred(25) = 75
Error range = [ -99.73646 .. 27.61901 ]
=====
Trustworthiness vs. McCabe , Num. classes , Num. parameters per
method
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 4.0479654687 1.2227291309  3.310599 0.0009309661
x1          1.7051097710 0.5560074627  3.066703 0.0021643377
x2         -0.0008548898 0.0002706781 -3.158327 0.0015867758
x3         -7.2109771241 2.3530808392 -3.064483 0.0021804629
R2log = 0.944457
Excluded as outliers: Hibernate JFreeChart Findbugs Xalan ( 4 / 16 )
MMRE = 27.57755
Pred(25) = 68.75
Error range = [ -96.38393 .. 33.76057 ]
=====
Trustworthiness vs. NOC , Num. attributes per class , Num.
interfaces per class
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -0.4316035 0.53335075 -0.8092302 0.418382775
x1          0.9376168 0.33215971  2.8227890 0.004760790
x2          0.1451749 0.05628061  2.5794834 0.009894821
x3         -2.1505817 0.82116578 -2.6189373 0.008820417
R2log = 0.9474125
Excluded as outliers: JFreeChart Xalan Struts JMeter Log4J ( 5 /
16 )
MMRE = 24.68306
Pred(25) = 75
Error range = [ -80.56896 .. 81.16524 ]
=====
```

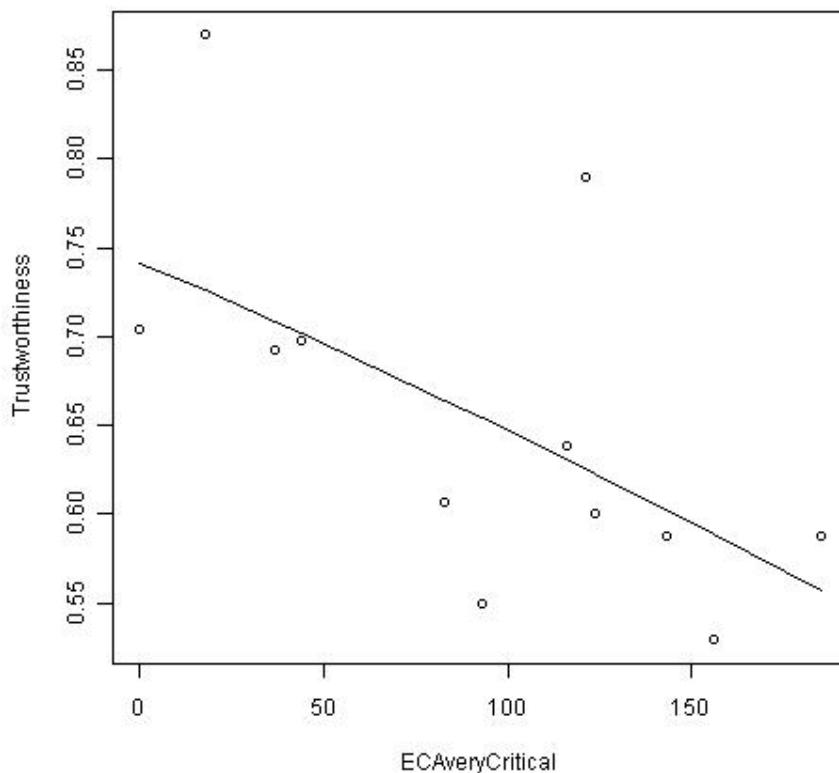
Univariate regressions tend to suggest that the trustworthiness of OSS products decreases with the amount of functionality provided, expressed in terms of number of methods, public methods or interfaces. Moreover, trustworthiness appears correlated to:

- Traditionally accepted indicators of object-oriented design quality, like the cohesion of classes and the response for classes.
- The quality of code (expressed in terms of violations of ECA rules).
- The stability of the application (expressed in terms of minor releases per year).
- The “liveness” of development (expressed in terms of files added per year).

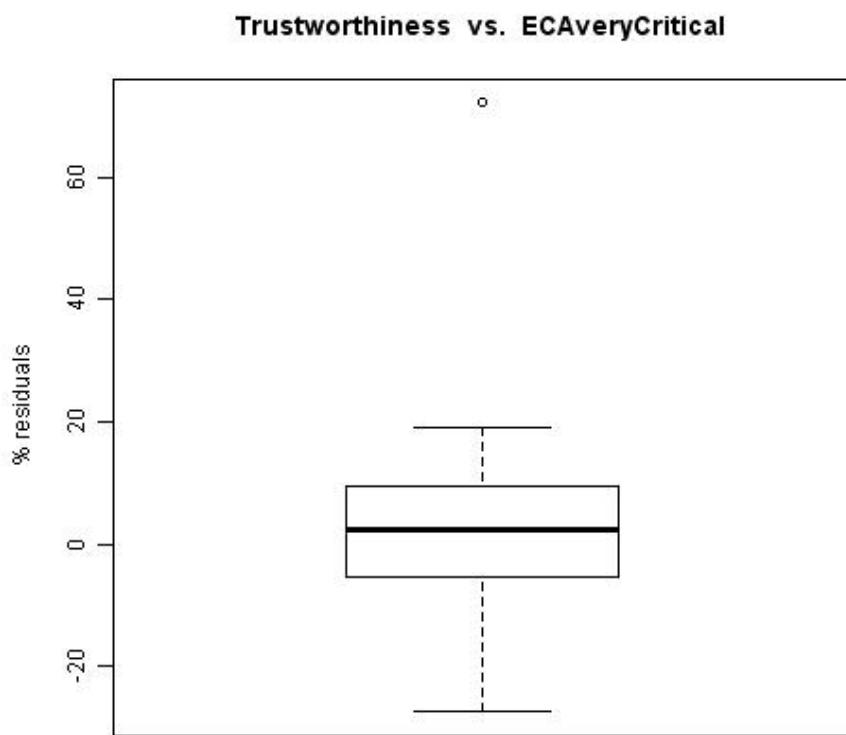
Other interesting models are those that show that a high level of Trustworthiness can be found for products that are well commented (many comment lines) and characterized by classes that have few interfaces (num.

interfaces per class) and are either rich in data (num. attributes per class), or poor in the number of parameters per method.

The trustworthiness vs. Number of very critical ECA rules violated is illustrated in Figure 5. It is fairly precise (MMRE=14.1%). The boxplot of relative residuals is reported in Figure 6.



**Figure 5. Trustworthiness vs. Number of very critical ECA rules violated: regression line**



**Figure 6. Trustworthiness vs. Number of very critical ECA rules violated: boxplot of relative residuals**

### 3 ANALYSIS OF JAVA AND C++ PROGRAM QUALITIES VS. OO CHARACTERISTICS

#### 3.1 Reliability

```
=====
Reliability vs. DIT
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 9.19445  2.344761  3.921274 0.0000880822
x1          -7.82697  2.083891 -3.755940 0.0001726920
R2log = 0.9026896
Excluded as outliers: Eclipse Perl Subversion Mono SpringFramework
Ant ( 6 / 19 )
MMRE = 27.25935
Pred(25) = 73.68421
Error range = [ -39.08723 .. 174.9719 ]
=====

Reliability vs. NOA_tot
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 8.809868e-01 1.423359e-01  6.189493 6.035795e-10
x1          -9.879488e-05 2.977698e-05 -3.317827 9.072052e-04
R2log = 0.9304286
Excluded as outliers: Eclipse xalan Hibernate HttpUnit weka ( 5 / 19 )
)
MMRE = 26.04594
Pred(25) = 57.89474
Error range = [ -78.8904 .. 107.5351 ]
=====

Reliability vs. NOA_class
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.2419884 0.37056837  3.351577 0.0008035272
x1          -0.1942622 0.08040306 -2.416105 0.0156875426
R2log = 0.91724
Excluded as outliers: Eclipse xalan HttpUnit Hibernate Perl Log4J ( 6 / 19 )
)
MMRE = 24.90891
Pred(25) = 68.42105
Error range = [ -84.67373 .. 104.3178 ]
=====

Reliability vs. RFC
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.17033925 0.27082217  4.321431 1.550207e-05
x1          -0.04039102 0.01314747 -3.072151 2.125222e-03
R2log = 0.9024962
Excluded as outliers: Eclipse Subversion Perl ( 3 / 19 )
MMRE = 19.87898
Pred(25) = 68.42105
Error range = [ -56.65978 .. 72.59991 ]
=====

Reliability vs. CBO , num_classes
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 0.237554451 0.2606914759  0.9112475 0.3621649908
x1          0.321682746 0.1100670322  2.9226076 0.0034711366
x2         -0.001218032 0.0003303407 -3.6871982 0.0002267368
```

```
R2log = 0.9314047
Excluded as outliers: Eclipse Hibernate HttpUnit Perl Findbugs weka
( 6 / 19 )
MMRE = 30.66281
Pred(25) = 68.42105
Error range = [ -94.9565 .. 129.7319 ]
=====
Reliability vs. DIT , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.340813e+00 4.434177e-01 5.279024 1.298738e-07
x1          -1.343693e+00 3.275368e-01 -4.102418 4.088552e-05
x2          -7.392357e-06 1.744161e-06 -4.238345 2.251732e-05
R2log = 0.9358413
Excluded as outliers: Eclipse Hibernate HttpUnit weka ( 4 / 19 )
MMRE = 17.57742
Pred(25) = 78.94737
Error range = [ -54.47255 .. 100.5708 ]
=====
Reliability vs. LOC_class , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.774267e-01 1.158948e-01 3.256633 0.001127420
x1          3.007856e-03 9.790248e-04 3.072298 0.002124173
x2          -3.418872e-06 1.191270e-06 -2.869939 0.004105504
R2log = 0.917524
Excluded as outliers: Eclipse Hibernate ( 2 / 19 )
MMRE = 17.70880
Pred(25) = 73.68421
Error range = [ -33.60404 .. 84.76623 ]
=====
Reliability vs. NOC_avg , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.9666834 0.36137575 2.675009 0.0074727220
x1          1.0638592 0.30085597 3.536108 0.0004060686
x2          -0.1387368 0.04285427 -3.237408 0.0012062063
R2log = 0.8897471
Excluded as outliers: Eclipse Perl Subversion Ant Hibernate ( 5 / 19
)
MMRE = 22.92329
Pred(25) = 68.42105
Error range = [ -56.94354 .. 65.9153 ]
=====
Reliability vs. NOC_avg , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.10921501 0.37821376 2.932773 0.003359497
x1          0.89946052 0.32153965 2.797355 0.005152291
x2          -0.06924565 0.02138604 -3.237890 0.001204172
R2log = 0.8927042
Excluded as outliers: Eclipse Perl Subversion Mono Ant Hibernate ( 6
/ 19 )
MMRE = 27.72827
Pred(25) = 68.42105
Error range = [ -89.24164 .. 63.46906 ]
=====
Reliability vs. NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8426405 0.62996236 1.337604 0.1810253989
```

```

x1      -0.2194605 0.06230256 -3.522496 0.0004275039
x2      1.7637168 0.81834471 2.155225 0.0311442479
R2log = 0.9151146
Excluded as outliers: Eclipse Perl Subversion Mono Struts Weka ( 6 /
19 )
MMRE = 30.24023
Pred(25) = 57.89474
Error range = [ -58.66085 .. 174.9980 ]
=====
Reliability vs. CBO , eLOC , NOM_class
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.045533e+00 3.594872e-01 2.908400 0.003632833
x1          1.536938e-01 7.184854e-02 2.139136 0.032424645
x2          -3.563821e-06 1.227180e-06 -2.904073 0.003683424
x3          -1.098609e-01 4.221998e-02 -2.602106 0.009265319
R2log = 0.9054288
Excluded as outliers: Eclipse Perl subversion ( 3 / 19 )
MMRE = 21.45670
Pred(25) = 73.68421
Error range = [ -53.6115 .. 112.0214 ]
=====
Reliability vs. CBO , eLOC , RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.276795e-01 3.719895e-01 2.493832 0.01263723
x1          2.115645e-01 1.015285e-01 2.083795 0.03717886
x2          -5.808341e-06 2.559051e-06 -2.269725 0.02322427
x3          -5.013071e-02 2.141815e-02 -2.340571 0.01925426
R2log = 0.9177348
Excluded as outliers: Eclipse Perl Subversion Mono HttpUnit ( 5 / 19
)
MMRE = 31.35281
Pred(25) = 63.1579
Error range = [ -84.55648 .. 112.0771 ]
=====
Reliability vs. CBO , num_classes , NOM_class
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8015429223 0.3601589958 2.225525 0.026046001
x1          0.3708410018 0.1298290109 2.856380 0.004285018
x2          -0.0007786693 0.0002583005 -3.014586 0.002573298
x3          -0.1124344090 0.0493332816 -2.279078 0.022662414
R2log = 0.907661
Excluded as outliers: Eclipse Perl Hibernate HttpUnit Subversion ( 5
/ 19 )
MMRE = 27.89600
Pred(25) = 68.42105
Error range = [ -69.40517 .. 112.0805 ]
=====
Reliability vs. CBO , num_classes , RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.9880900619 0.3784965318 2.610566 0.00903926
x1          0.3151012567 0.1338405523 2.354303 0.01855746
x2          -0.0005901947 0.0002803687 -2.105066 0.03528555
x3          -0.0622911648 0.0249145569 -2.500192 0.01241262
R2log = 0.9125699
Excluded as outliers: Eclipse Hibernate Perl Subversion HttpUnit Mono
( 6 / 19 )

```

```

MMRE = 30.51855
Pred(25) = 57.89474
Error range = [ -70.35907 .. 112.0805 ]
=====
Reliability vs. DIT , NOC_avg , NOA_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.702083e-01 1.511700e-01 5.756488 8.588187e-09
x1          -2.844113e+00 6.538589e-01 -4.349735 1.363024e-05
x2           3.524763e+00 8.103935e-01  4.349447 1.364814e-05
x3          -8.663225e-05 2.450581e-05 -3.535172 4.075095e-04
R2log = 0.9350407
Excluded as outliers: Eclipse Ant Struts Findbugs SpringFramework
JasperReports ( 6 / 19 )
MMRE = 28.36595
Pred(25) = 57.89474
Error range = [ -87.47118 .. 69.73122 ]
=====
Reliability vs. LOC_class , eLOC , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.098932e+00 3.116668e-01 3.525983 0.0004219142
x1          2.556378e-03 8.405901e-04 3.041171 0.0023565996
x2          -3.132834e-06 1.042929e-06 -3.003881 0.0026655920
x3          -1.423937e-01 6.851466e-02 -2.078296 0.0376821371
R2log = 0.9324595
Excluded as outliers: xalan Eclipse HttpUnit ( 3 / 19 )
MMRE = 21.11316
Pred(25) = 73.68421
Error range = [ -74.75919 .. 108.4027 ]
=====
Reliability vs. eLOC , NOC_avg , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.827714e-01 3.609572e-01 2.722681 0.006475455
x1          -2.288747e-06 1.038534e-06 -2.203825 0.027536682
x2           8.272442e-01 2.611721e-01  3.167430 0.001537929
x3          -1.117035e-01 3.961727e-02 -2.819566 0.004808867
R2log = 0.9021388
Excluded as outliers: Eclipse Perl Subversion Ant ( 4 / 19 )
MMRE = 19.98092
Pred(25) = 68.42105
Error range = [ -59.4227 .. 68.8591 ]
=====
Reliability vs. LOC_class , eLOC , NOC_avg , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.755432e-01 3.616918e-01 2.697167 0.0069932139
x1          6.359476e-03 1.920487e-03 3.311386 0.0009283497
x2          -4.160798e-06 1.386788e-06 -3.000313 0.0026970242
x3           1.029300e+00 2.904751e-01  3.543504 0.0003948469
x4          -1.618864e-01 4.390574e-02 -3.687136 0.0002267923
R2log = 0.922879
Excluded as outliers: Eclipse Ant Perl ( 3 / 19 )
MMRE = 15.04591
Pred(25) = 84.21053
Error range = [ -31.59923 .. 57.60327 ]
=====
Reliability vs. eLOC , NOC_avg , NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)

```

```
(Intercept) 8.235456e-01 3.255412e-01 2.529774 1.141362e-02
x1         -3.604582e-06 1.057859e-06 -3.407431 6.557749e-04
x2          1.002191e+00 2.875965e-01 3.484712 4.926674e-04
x3         -1.185917e-01 3.850887e-02 -3.079594 2.072830e-03
x4          1.772497e-01 3.869357e-02 4.580857 4.630737e-06
R2log = 0.9267199
Excluded as outliers: Eclipse Ant ( 2 / 19 )
MMRE = 15.88928
Pred(25) = 78.94737
Error range = [ -37.67299 .. 69.79773 ]
=====
```

### 3.2 Usability

```
=====
usability vs. NOA_class
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.5003616 0.4036645 3.716853 0.0002017196
x1        -0.3253753 0.0940320 -3.460262 0.0005396505
R2log = 0.9439427
Excluded as outliers: Xalan Log4J HttpUnit Hibernate Xerces Perl ( 6
/ 19 )
MMRE = 25.45628
Pred(25) = 68.42105
Error range = [ -97.69535 .. 86.52027 ]
=====

usability vs. DIT , RFC
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.43184921 0.443911132 3.225531 0.001257391
x1        -0.67610741 0.255879358 -2.642290 0.008234757
x2        -0.02792454 0.009307659 -3.000168 0.002698307
R2log = 0.9383676
Excluded as outliers: Log4J ( 1 / 19 )
MMRE = 18.02903
Pred(25) = 78.94737
Error range = [ -38.12288 .. 68.3278 ]
=====

usability vs. eLOC , NOM_class
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.252137e+00 4.196248e-01 2.983945 0.002845582
x1        -1.837867e-06 8.338661e-07 -2.204032 0.027522113
x2        -1.000181e-01 4.042796e-02 -2.473984 0.013361556
R2log = 0.9370153
Excluded as outliers: Perl Struts Log4J ( 3 / 19 )
MMRE = 19.02303
Pred(25) = 73.68421
Error range = [ -41.32991 .. 69.09885 ]
=====

usability vs. num_abstract_classes , RFC
      Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 2.28439723 0.49184582 4.644539 3.408363e-06
x1        -0.01084170 0.00308439 -3.515023 4.397151e-04
x2        -0.07946134 0.02141907 -3.709840 2.073902e-04
R2log = 0.9422374
Excluded as outliers: Hibernate Subversion Perl Mono Saxon ( 5 / 18
)
MMRE = 26.17127
```

```

Pred(25) = 55.55556
Error range = [ -76.31057 .. 53.94561 ]
=====
Usability vs. NOA_class , NOM_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.528988e+00 4.036597e-01 3.787814 1.519788e-04
x1          -4.135483e-01 1.033948e-01 -3.999700 6.342277e-05
x2           2.907735e-05 1.406882e-05  2.066794 3.875360e-02
R2log = 0.9473835
Excluded as outliers: Xalan Log4J Hibernate Perl Xerces Httpunit ( 6
/ 19 )
MMRE = 28.84795
Pred(25) = 57.89474
Error range = [ -99.22605 .. 80.58475 ]
=====
Usability vs. CBO , DIT , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.25991591 0.66520904 3.397302 6.805385e-04
x1          0.25316271 0.08198439 3.087938 2.015505e-03
x2          -1.68215705 0.41538452 -4.049638 5.129686e-05
x3          -0.06959075 0.02119326 -3.283627 1.024806e-03
R2log = 0.9093855
Excluded as outliers: Eclipse Hibernate SpringFramework Subversion
Log4J ( 5 / 19 )
MMRE = 22.7462
Pred(25) = 63.1579
Error range = [ -39.96112 .. 78.72304 ]
=====
Usability vs. CBO , NOC_avg , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.480510262 0.2687844232 -1.787716 7.382181e-02
x1          0.751091142 0.1640279939 4.579042 4.671095e-06
x2          -0.750256657 0.2533676417 -2.961138 3.065042e-03
x3          -0.001598018 0.0003385618 -4.720019 2.358223e-06
R2log = 0.9295715
Excluded as outliers: Eclipse Subversion Hibernate HttpUnit
JFreeChart Findbugs ( 6 / 19 )
MMRE = 33.78214
Pred(25) = 63.1579
Error range = [ -97.00711 .. 134.7171 ]
=====
Usability vs. DIT , NOM_tot , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.127249e+00 6.333400e-01 3.358779 0.0007828768
x1          -1.288806e+00 3.727459e-01 -3.457599 0.0005450116
x2           3.454618e-05 1.550598e-05  2.227926 0.0258854343
x3          -5.389230e-02 1.835524e-02 -2.936072 0.0033239756
R2log = 0.9340052
Excluded as outliers: Log4J Hibernate SpringFramework Subversion ( 4
/ 19 )
MMRE = 21.40191
Pred(25) = 63.1579
Error range = [ -34.37344 .. 65.53428 ]
=====
Usability vs. LOC_class , eLOC , NOM_class
      Estimate Std. Error z value Pr(>|z|)

```

```
(Intercept) 1.890125e+00 4.504706e-01 4.195890 2.718027e-05
x1          2.424297e-02 8.494319e-03 2.854022 4.316951e-03
x2          -7.563611e-06 2.032655e-06 -3.721049 1.983969e-04
x3          -3.050423e-01 7.810142e-02 -3.905720 9.394539e-05
R2log = 0.9354754
Excluded as outliers: Perl subversion Mono Struts Hibernate ( 5 / 19 )
)
MMRE = 27.96435
Pred(25) = 57.89474
Error range = [ -38.61756 .. 106.5220 ]
=====
Usability vs. LOC_class , num_abstract_classes , num_classes
Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.421457532 0.3370908881 4.216838 2.477523e-05
x1          -0.014156066 0.0036184940 -3.912143 9.148066e-05
x2          -0.026547141 0.0059579972 -4.455716 8.361385e-06
x3          0.001163845 0.0003009015 3.867860 1.097945e-04
R2log = 0.9369254
Excluded as outliers: Perl Subversion Hibernate SpringFramework ( 4
/ 18 )
)
MMRE = 23.77415
Pred(25) = 66.66667
Error range = [ -95.88068 .. 52.09951 ]
=====
Usability vs. LOC_class , num_abstract_classes , NOM_tot
Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.607303e+00 5.390005e-01 4.837292 1.316202e-06
x1          -2.703778e-02 6.464437e-03 -4.182541 2.882687e-05
x2          -2.299565e-02 5.727707e-03 -4.014808 5.949410e-05
x3          8.525322e-05 2.428398e-05 3.510678 4.469657e-04
R2log = 0.9443437
Excluded as outliers: Perl Subversion Mono HttpUnit ( 4 / 18 )
)
MMRE = 29.71893
Pred(25) = 61.11111
Error range = [ -99.9134 .. 94.46588 ]
=====
Usability vs. num_abstract_classes , NOA_class , RFC
Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.16270948 0.447170565 4.836431 1.321913e-06
x1          -0.01046420 0.002709151 -3.862537 1.122154e-04
x2          -0.17605560 0.078344996 -2.247184 2.462829e-02
x3          -0.02680148 0.006575370 -4.076041 4.580889e-05
R2log = 0.9491878
Excluded as outliers: Hibernate Xalan HttpUnit ( 3 / 18 )
)
MMRE = 21.78578
Pred(25) = 66.66667
Error range = [ -78.67393 .. 92.19081 ]
=====
Usability vs. num_abstract_classes , num_classes , RFC
Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.159438564 0.4435793652 4.868212 1.126123e-06
x1          -0.025919391 0.0053830706 -4.814982 1.472128e-06
x2          0.001120332 0.0002939805 3.810905 1.384587e-04
x3          -0.090135329 0.0202497546 -4.451181 8.539920e-06
R2log = 0.911589
```

```

Excluded as outliers: Hibernate Subversion Perl Eclipse
SpringFramework ( 5 / 18 )
MMRE = 18.14438
Pred(25) = 83.33333
Error range = [ -63.12723 .. 24.56751 ]
=====
Usability vs. num_abstract_classes , NOM_class , NPM_avg
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.920868511 0.475202355 4.042212 0.0000529494
x1         -0.009060714 0.003039549 -2.980941 0.0028736456
x2         -0.114136771 0.039685170 -2.876056 0.0040267825
x3         -0.133093100 0.035460799 -3.753246 0.0001745594
R2log = 0.9382102
Excluded as outliers: Perl Hibernate ( 2 / 18 )
MMRE = 19.53203
Pred(25) = 72.22222
Error range = [ -66.0782 .. 58.13719 ]
=====
Usability vs. CBO , DIT , eLOC , NOM_class
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.734273e+00 6.069186e-01 4.505173 6.631898e-06
x1         3.142382e-01 8.396065e-02 3.742684 1.820648e-04
x2        -2.068539e+00 4.402387e-01 -4.698676 2.618539e-06
x3        -8.347764e-06 1.918508e-06 -4.351176 1.354092e-05
x4        -1.000845e-01 3.648032e-02 -2.743522 6.078402e-03
R2log = 0.9206368
Excluded as outliers: Eclipse Subversion ( 2 / 19 )
MMRE = 18.95395
Pred(25) = 84.21053
Error range = [ -20.30330 .. 78.72338 ]
=====
Usability vs. CBO , NOA_class , num_classes , NOM_class
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.674316576 0.3811721419 1.769060 7.688378e-02
x1         0.700906694 0.1557591872 4.499938 6.797327e-06
x2        -0.092085078 0.0415497677 -2.216260 2.667372e-02
x3        -0.001196221 0.0003093007 -3.867503 1.099555e-04
x4        -0.164927536 0.0524740120 -3.143033 1.672072e-03
R2log = 0.915366
Excluded as outliers: Eclipse Hibernate Perl Subversion HttpUnit
Findbugs ( 6 / 19 )
MMRE = 29.02868
Pred(25) = 68.42105
Error range = [ -81.97341 .. 113.6418 ]
=====
Usability vs. DIT , LOC_class , NOA_class , NOM_tot
          Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.868630e+00 1.211943e+00 3.192090 0.0014124733
x1        -1.620172e+00 7.853040e-01 -2.063115 0.0391017215
x2        -8.349439e-03 4.090354e-03 -2.041251 0.0412259134
x3        -4.508927e-01 1.163252e-01 -3.876140 0.0001061265
x4         5.137588e-05 1.866151e-05  2.753040 0.0059044696
R2log = 0.950866
Excluded as outliers: Perl xalan Log4J Hibernate Xerces HttpUnit ( 6
/ 19 )
MMRE = 30.92624

```

```

Pred(25) = 57.89474
Error range = [ -99.4033 .. 85.27701 ]
=====
Usability vs. DIT , NOA_class , NOM_tot , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.382857e+00 2.125340e+00 3.003217 2.671422e-03
x1          -3.387992e+00 1.446997e+00 -2.341396 1.921178e-02
x2          -5.487864e-01 1.286208e-01 -4.266701 1.983847e-05
x3           3.821324e-05 1.623030e-05  2.354438 1.855071e-02
x4          -5.412408e-01 2.331386e-01 -2.321541 2.025767e-02
R2log = 0.9518973
Excluded as outliers: Perl xalan Log4J Hibernate Xerces HttpUnit ( 6
/ 19 )
MMRE = 32.43370
Pred(25) = 57.89474
Error range = [ -99.85002 .. 92.01862 ]
=====
Usability vs. num_abstract_classes , NOA_class , num_classes ,
RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.920047140 0.5389172240 5.418359 6.014845e-08
x1          -0.022947204 0.0052900995 -4.337764 1.439395e-05
x2          -0.279856392 0.0906047498 -3.088761 2.009933e-03
x3           0.000818638 0.0002880858  2.841647 4.488117e-03
x4          -0.048852790 0.0102253535 -4.777614 1.773877e-06
R2log = 0.9410119
Excluded as outliers: xalan HttpUnit Hibernate Eclipse ( 4 / 18 )
MMRE = 18.37164
Pred(25) = 88.88889
Error range = [ -89.46024 .. 81.66713 ]
=====
```

### 3.3 Portability

```

Portability vs. DIT
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.38866   2.494510  3.763729 0.0001673980
x1          -8.12162   2.241206 -3.623772 0.0002903380
R2log = 0.9150252
Excluded as outliers: Subversion Perl Mono Eclipse SpringFramework
Xerces ( 6 / 19 )
MMRE = 24.75527
Pred(25) = 63.1579
Error range = [ -37.82363 .. 108.3158 ]
=====
Portability vs. NOC_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.1032689  0.1786349 -0.5781004 0.563196355
x1           0.5217145  0.1711997  3.0474026 0.002308283
R2log = 0.9310593
Excluded as outliers: Perl Subversion ( 2 / 19 )
MMRE = 12.82051
Pred(25) = 78.94737
Error range = [ -34.5749 .. 54.75781 ]
=====
Portability vs. LOC_class , NOM_class , NPM_avg
```

```

              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.70825093 0.557700797 3.063024 0.002191125
x1          0.01537493 0.006309317 2.436862 0.014815331
x2         -0.19683862 0.071871877 -2.738743 0.006167452
x3         -0.52788552 0.203991351 -2.587784 0.009659557
R2log = 0.9240858
Excluded as outliers: Perl Struts Eclipse ( 3 / 19 )
MMRE = 17.11030
Pred(25) = 78.94737
Error range = [ -78.76592 .. 54.75556 ]
=====
Portability vs. eLOC , NOC_avg , NPM_avg
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -2.532902e-01 2.054608e-01 -1.232791 2.176537e-01
x1          -1.894730e-06 9.345268e-07 -2.027475 4.261381e-02
x2           7.348003e-01 1.826247e-01  4.023555 5.732625e-05
x3           1.250116e-01 3.010493e-02  4.152529 3.288205e-05
R2log = 0.9420108
Excluded as outliers: ( 0 / 19 )
MMRE = 9.63466
Pred(25) = 94.73684
Error range = [ -23.77546 .. 59.20577 ]
=====
Portability vs. eLOC , num_abstract_classes , RFC
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.617593e+00 3.501351e-01 4.619911 3.839054e-06
x1          5.866984e-06 2.124657e-06 2.761379 5.755776e-03
x2         -1.012255e-02 3.339107e-03 -3.031515 2.433299e-03
x3         -5.213867e-02 1.569777e-02 -3.321405 8.956543e-04
R2log = 0.926683
Excluded as outliers: Hibernate Mono Eclipse ( 3 / 18 )
MMRE = 16.14806
Pred(25) = 77.77778
Error range = [ -25.09373 .. 72.99319 ]
=====
Portability vs. num_abstract_classes , num_classes , NOM_class
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.599505136 0.4331144510 3.693031 0.0002215969
x1          -0.015474471 0.0058382937 -2.650513 0.0080369748
x2           0.000585534 0.0002406130  2.433510 0.0149532418
x3          -0.107607067 0.0419112745 -2.567497 0.0102435710
R2log = 0.9359667
Excluded as outliers: Perl Subversion Mono Ant ( 4 / 18 )
MMRE = 16.03865
Pred(25) = 83.33333
Error range = [ -21.70406 .. 77.01724 ]
=====
Portability vs. num_abstract_classes , num_classes , RFC
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.6726374730 0.381925678 4.379484 1.189605e-05
x1          -0.0146242928 0.005326025 -2.745817 6.036035e-03
x2           0.0005486435 0.000217172  2.526309 1.152682e-02
x3          -0.0582754839 0.016826226 -3.463372 5.334498e-04
R2log = 0.93448
Excluded as outliers: Perl Subversion Ant ( 3 / 18 )
MMRE = 13.978

```

```

Pred(25) = 83.33333
Error range = [ -49.97226 .. 45.05065 ]
=====
Portability vs. num_abstract_classes , NOM_tot , NOM_class
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.394492e+00 5.665513e-01 4.226435 2.374225e-05
x1          -1.959248e-02 6.591813e-03 -2.972245 2.956312e-03
x2           8.278948e-05 2.989903e-05  2.768968 5.623410e-03
x3          -2.053544e-01 5.967256e-02 -3.441354 5.788107e-04
R2log = 0.9386772
Excluded as outliers: Perl Subversion Mono Ant SpringFramework ( 5 / 18 )
MMRE = 16.50353
Pred(25) = 83.33333
Error range = [ -25.0264 .. 79.71536 ]
=====
Portability vs. num_abstract_classes , NOM_tot , RFC
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.144172e+00 5.497651e-01 3.900161 9.612886e-05
x1          -1.533894e-02 5.731753e-03 -2.676135 7.447670e-03
x2           5.866474e-05 2.408404e-05  2.435835 1.485745e-02
x3          -8.444293e-02 2.597280e-02 -3.251206 1.149166e-03
R2log = 0.934367
Excluded as outliers: Hibernate Perl Subversion Ant Mono ( 5 / 18 )
MMRE = 17.10712
Pred(25) = 77.77778
Error range = [ -78.21121 .. 39.35438 ]
=====
```

### 3.4 How well are functional requirements satisfied

```

=====
Functionality vs. DIT
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.80991  2.327406 3.355628 0.0007918504
x1          -6.64712  2.070755 -3.209999 0.0013273554
R2log = 0.896114
Excluded as outliers: Perl Subversion Mono Ant Eclipse
SpringFramework ( 6 / 19 )
MMRE = 34.71871
Pred(25) = 57.89474
Error range = [ -30.71730 .. 177.6646 ]
=====
Functionality vs. NOA_class
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.936778 0.29046335 3.225116 0.001259217
x1          -0.164842 0.06778973 -2.431666 0.015029545
R2log = 0.9437502
Excluded as outliers: Log4J Xalan HttpUnit Perl ( 4 / 19 )
MMRE = 23.49420
Pred(25) = 78.94737
Error range = [ -75.84607 .. 95.59488 ]
=====
Functionality vs. CBO , eLOC
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.602959e-02 2.683812e-01 0.05972696 0.9523730960
x1          2.460468e-01 9.044267e-02  2.72047234 0.0065188725
```

```

x2      -1.147935e-05 3.074780e-06 -3.73338848 0.0001889209
R2log = 0.9168487
Excluded as outliers: Eclipse Perl Subversion Mono Hibernate HttpUnit
( 6 / 19 )
MMRE = 31.63941
Pred(25) = 63.1579
Error range = [ -91.62975 .. 118.6485 ]
=====
Functionality vs. CBO , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.026376000 0.2586579245 -0.1019725 0.9187784887
x1          0.318369059 0.1045261741  3.0458310 0.0023203824
x2          -0.001044272 0.0002923314 -3.5722193 0.0003539688
R2log = 0.9287306
Excluded as outliers: Eclipse Hibernate HttpUnit Findbugs Perl ( 5 /
19 )
MMRE = 27.0261
Pred(25) = 68.42105
Error range = [ -91.51855 .. 124.8756 ]
=====
Functionality vs. DIT , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.891210e+00 4.062171e-01 4.655664 3.229389e-06
x1          -1.092263e+00 3.009675e-01 -3.629173 2.843305e-04
x2          -6.183014e-06 1.589698e-06 -3.889428 1.004809e-04
R2log = 0.932447
Excluded as outliers: Eclipse Hibernate HttpUnit ( 3 / 19 )
MMRE = 16.47422
Pred(25) = 84.21053
Error range = [ -31.44239 .. 97.75406 ]
=====
Functionality vs. eLOC , num_abstract_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.357412e-01 1.857019e-01 5.038941 4.681145e-07
x1          -2.542819e-06 8.371699e-07 -3.037400 2.386288e-03
x2          -7.420412e-03 2.673559e-03 -2.775481 5.512016e-03
R2log = 0.9304327
Excluded as outliers: Hibernate Eclipse HttpUnit ( 3 / 18 )
MMRE = 21.54753
Pred(25) = 77.77778
Error range = [ -53.73469 .. 103.8268 ]
=====
Functionality vs. eLOC , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.789777e+00 4.982321e-01 3.592254 0.0003278295
x1          -3.806529e-06 1.056818e-06 -3.601878 0.0003159271
x2          -1.085516e-01 4.769387e-02 -2.276008 0.0228455184
R2log = 0.9124732
Excluded as outliers: Perl Subversion Struts HttpUnit Eclipse Ant ( 6 /
19 )
MMRE = 20.07539
Pred(25) = 84.21053
Error range = [ -36.92347 .. 99.00563 ]
=====
Functionality vs. NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)

```

```
(Intercept) 0.2032856 0.65337256 0.3111328 0.755699687
x1         -0.1753745 0.05685192 -3.0847601 0.002037164
x2          1.8437905 0.74825358  2.4641252 0.013734813
R2log = 0.9069901
Excluded as outliers: Perl Subversion Mono Struts Eclipse Log4J ( 6
/ 19 )
MMRE = 29.0479
Pred(25) = 63.1579
Error range = [ -32.08693 .. 177.7762 ]
=====
Functionality vs. DIT , LOC_class , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.61896794 1.128919362 -1.434086 0.1515475993
x1          4.39746481 1.304007638  3.372269 0.0007455154
x2          0.01949890 0.006142386  3.174484 0.0015010336
x3          -0.45477411 0.117305290 -3.876842 0.0001058209
R2log = 0.9273051
Excluded as outliers: Eclipse Struts JFreeChart Perl ( 4 / 19 )
MMRE = 19.58057
Pred(25) = 73.68421
Error range = [ -43.90468 .. 86.44864 ]
=====
Functionality vs. LOC_class , NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.29201625 0.683267890 1.890937 5.863281e-02
x1          0.01914562 0.007812127  2.450757 1.425563e-02
x2          -0.37640149 0.095985558 -3.921439 8.802182e-05
x3          1.49870052 0.761111616  1.969094 4.894227e-02
R2log = 0.9108565
Excluded as outliers: Perl Subversion Mono Struts Eclipse Ant ( 6 /
19 )
MMRE = 31.45285
Pred(25) = 63.1579
Error range = [ -19.9579 .. 177.7774 ]
=====
Functionality vs. eLOC , NOC_avg , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.173669e+00 4.819173e-01  4.510460 6.468731e-06
x1          -3.240013e-06 1.195289e-06 -2.710653 6.715088e-03
x2          -4.804739e-01 1.574993e-01 -3.050641 2.283537e-03
x3          -2.335761e-01 7.816823e-02 -2.988120 2.806992e-03
R2log = 0.9542656
Excluded as outliers: Log4J xalan HttpUnit Struts SpringFramework ( 5
/ 19 )
MMRE = 24.73709
Pred(25) = 78.94737
Error range = [ -90.8226 .. 142.3043 ]
=====
Functionality vs. eLOC , num_abstract_classes , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.240146e+00 4.985475e-01  4.493344 7.011324e-06
x1          -2.729755e-06 8.416989e-07 -3.243149 1.182162e-03
x2          -6.659970e-03 1.986902e-03 -3.351937 8.024823e-04
x3          -2.746686e-01 9.436995e-02 -2.910552 3.607912e-03
R2log = 0.9543392
```

```

Excluded as outliers: Hibernate HttpUnit Xalan SpringFramework ( 4 / 18 )
MMRE = 23.19727
Pred(25) = 77.77778
Error range = [ -92.50951 .. 143.7929 ]
=====
Functionality vs. num_abstract_classes , num_classes , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.0429054459 0.4374768391 4.669745 3.015733e-06
x1          -0.0129100450 0.0048032082 -2.687796 7.192527e-03
x2           0.0004920086 0.0002502017 1.966448 4.924689e-02
x3          -0.0781696608 0.0194000125 -4.029361 5.592859e-05
R2log = 0.905329
Excluded as outliers: Hibernate Subversion Perl Eclipse ( 4 / 18 )
MMRE = 19.72865
Pred(25) = 72.22222
Error range = [ -73.2239 .. 60.13232 ]
=====
Functionality vs. CBO , LOC_class , eLOC , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.636136e-01 3.802728e-01 1.482129 0.138305843
x1          1.325895e-01 6.374153e-02 2.080112 0.037515232
x2           3.231289e-03 1.016146e-03 3.179944 0.001473035
x3          -3.837420e-06 1.420207e-06 -2.702016 0.006892054
x4          -1.841956e-01 7.052175e-02 -2.611898 0.009004105
R2log = 0.937908
Excluded as outliers: Eclipse HttpUnit xalan Log4J ( 4 / 19 )
MMRE = 23.19516
Pred(25) = 73.68421
Error range = [ -77.21106 .. 110.8443 ]
=====
Functionality vs. DIT , NOC_avg , NOA_tot , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.5563969770 4.036881e-01 3.855444 0.0001155196
x1          0.8368505783 4.153996e-01 2.014568 0.0439499543
x2          -0.8315102131 3.797015e-01 -2.189905 0.0285311126
x3          -0.0000565031 2.664229e-05 -2.120805 0.0339382037
x4          -0.2139029383 7.958940e-02 -2.687581 0.0071971682
R2log = 0.9553053
Excluded as outliers: Xalan Log4J HttpUnit Struts SpringFramework
JasperReports ( 6 / 19 )
MMRE = 27.77935
Pred(25) = 63.1579
Error range = [ -96.69165 .. 151.5041 ]
=====
Functionality vs. DIT , NOC_avg , NOA_class , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.623348951 0.4233207710 3.834796 0.0001256684
x1          -0.743200124 0.2651694779 -2.802736 0.0050671080
x2           1.126200483 0.3577329461 3.148160 0.0016430184
x3          -0.203440019 0.0925919915 -2.197166 0.0280085590
x4          -0.000386465 0.0001791340 -2.157408 0.0309739077
R2log = 0.9358661
Excluded as outliers: Hibernate Findbugs Ant Eclipse HttpUnit xalan
( 6 / 19 )
MMRE = 26.20090

```

```

Pred(25) = 63.1579
Error range = [ -70.75003 .. 116.7326 ]
=====
Functionality vs. LOC_class , NOC_avg , NOM_tot , NPM_avg
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.947712e+00 7.756593e-01 -2.511041 0.012037587
x1          -3.106904e-02 9.778644e-03 -3.177234 0.001486869
x2           7.553499e-01 3.185310e-01  2.371355 0.017723019
x3           7.346473e-05 2.600489e-05  2.825035 0.004727546
x4           3.149868e+00 9.784739e-01  3.219164 0.001285648
R2log = 0.9383142
Excluded as outliers: Hibernate Perl Subversion Mono Log4J Xalan ( 6
/ 19 )
MMRE = 35.24786
Pred(25) = 57.89474
Error range = [ -40.87512 .. 177.7776 ]
=====
Functionality vs. LOC_class , NOC_avg , NOM_class , NPM_avg
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.44482597 0.898828572 2.720014 6.527922e-03
x1          0.02929907 0.009491701 3.086809 2.023175e-03
x2          -0.97229783 0.374522233 -2.596102 9.428813e-03
x3          -0.49190805 0.113714039 -4.325834 1.519558e-05
x4           1.55836885 0.755211732  2.063486 3.906647e-02
R2log = 0.9243234
Excluded as outliers: Perl Subversion Mono Struts Eclipse ( 5 / 19 )
MMRE = 27.76491
Pred(25) = 78.94737
Error range = [ -24.03254 .. 177.7778 ]
=====
Functionality vs. num_abstract_classes , NOA_class , num_classes ,
RFC
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.6334153706 0.5220222268 5.044642 4.543707e-07
x1          -0.0175343102 0.0052881295 -3.315787 9.138551e-04
x2          -0.0832851016 0.0386833634 -2.152995 3.131904e-02
x3           0.0008224689 0.0002977105 2.762647 5.733475e-03
x4          -0.0913108675 0.0204931207 -4.455684 8.362618e-06
R2log = 0.9117899
Excluded as outliers: Hibernate Subversion Perl Eclipse ( 4 / 18 )
MMRE = 20.31626
Pred(25) = 66.66667
Error range = [ -76.10393 .. 63.0015 ]
=====
Functionality vs. num_abstract_classes , NOA_class , NOM_tot ,
RFC
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.223262e+00 7.265397e-01 4.436457 9.145159e-06
x1          -1.665141e-02 5.269090e-03 -3.160206 1.576577e-03
x2          -7.580258e-02 3.753450e-02 -2.019544 4.343070e-02
x3           7.189186e-05 2.635808e-05  2.727507 6.381482e-03
x4          -1.226107e-01 3.400100e-02 -3.606091 3.108437e-04
R2log = 0.9121011
Excluded as outliers: Subversion Hibernate Perl Eclipse Mono ( 5 /
18 )
MMRE = 24.27811

```

```
Pred(25) = 66.66667
Error range = [ -93.06743 .. 66.13784 ]
=====
```

### 3.5 Interoperability

```
Interoperability vs. NOA_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.640683e-01 1.365658e-01 2.665882 0.00767867
x1          -6.194192e-05 2.523441e-05 -2.454661 0.01410175
R2log = 0.9493036
Excluded as outliers: xalan Hibernate Httpunit (3 / 19)
MMRE = 18.94534
Pred(25) = 78.94737
Error range = [ -67.95742 .. 131.5935 ]
=====
```

```
Interoperability vs. CBO , DIT , NOC_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.6588168 0.26708468 2.466696 0.01363659
x1          -0.1450580 0.06243951 -2.323176 0.02016971
x2          -0.4913874 0.24720011 -1.987812 0.04683244
x3          0.7844460 0.32993225 2.377597 0.01742584
R2log = 0.9240956
Excluded as outliers: Eclipse Ant (2 / 19)
MMRE = 19.93456
Pred(25) = 84.21053
Error range = [ -99.80164 .. 98.01738 ]
=====
```

### 3.6 Security

```
Security vs. NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.972188 0.4140592 -2.347945 0.01887733
x1          1.068608 0.4705738 2.270862 0.02315531
R2log = 0.9325839
Excluded as outliers: Perl Subversion Mono (3 / 19)
MMRE = 25.88932
Pred(25) = 78.94737
Error range = [ -13.93843 .. 139.6872 ]
=====
```

```
Security vs. eLOC , num_abstract_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.611793e-01 2.376701e-01 2.781920 0.005403828
x1          -2.064449e-06 9.936063e-07 -2.077733 0.037733963
x2          -7.211312e-03 2.432309e-03 -2.964801 0.003028786
R2log = 0.939565
Excluded as outliers: Hibernate Log4J (2 / 18)
MMRE = 17.80256
Pred(25) = 77.77778
Error range = [ -51.14429 .. 31.005817]
=====
```

```
Security vs. NOA_tot , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2838648343 1.870344e-01 -1.517714 0.129086416
```

```

x1           -0.0001103598 3.419449e-05 -3.227415 0.001249140
x2            0.2046376486 6.654220e-02  3.075306 0.002102864
R2log = 0.9412001
Excluded as outliers: Hibernate Log4J Struts ( 3 / 19 )
MMRE = 18.77020
Pred(25) = 73.68421
Error range = [ -50.71715 .. 49.6564085 ]
=====
```

### 3.7 Performance (in terms of speed)

```

Speed vs. CBO
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.1317698 0.19794710 0.6656817 0.50561459
x1          -0.1028166 0.04649061 -2.2115556 0.02699739
R2log = 0.9314538
Excluded as outliers: Eclipse ( 1 / 19 )
MMRE = 18.31345
Pred(25) = 84.21053
Error range = [ -98.296 .. 34.45808 ]
=====

Speed vs. DIT
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 4.589609  2.283683  2.009740 0.04445875
x1          -4.294063  2.048340 -2.096362 0.03605008
R2log = 0.9131231
Excluded as outliers: Eclipse Subversion Perl Mono SpringFramework
Hibernate ( 6 / 19 )
MMRE = 36.96037
Pred(25) = 57.89474
Error range = [ -29.70468 .. 163.9780 ]
=====

Speed vs. NOA_tot
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.739712e-02 1.141618e-01 -0.1523900 0.878879376
x1          -5.480147e-05 1.950049e-05 -2.8102617 0.004950124
R2log = 0.9355728
Excluded as outliers: Eclipse xalan ( 2 / 19 )
MMRE = 17.03771
Pred(25) = 84.21053
Error range = [ -54.07845 .. 75.09686 ]
=====

Speed vs. num_classes
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.0054106985 0.1384318460 0.03908565 0.96882210
x1          -0.0002859882 0.0001380472 -2.07166880 0.03829634
R2log = 0.931425
Excluded as outliers: Hibernate Eclipse ( 2 / 19 )
MMRE = 16.23726
Pred(25) = 84.21053
Error range = [ -38.11884 .. 66.74786 ]
=====
```

```

Speed vs. CBO , NOC_avg
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.04022432 0.23772785 0.1692032 0.86563678
x1          -0.17618620 0.06563509 -2.6843293 0.00726755
=====
```

```

x2          0.49898227 0.24986904  1.9969752 0.04582788
R2log =  0.9176431
Excluded as outliers: Eclipse Subversion Perl  ( 3 / 19 )
MMRE = 16.72473
Pred(25) = 78.94737
Error range = [ -99.94777 .. 29.45293 ]
=====
Speed vs. DIT , eLOC
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 5.656807e-01 3.324043e-01 1.701785 0.088795676
x1         -5.216015e-01 2.402986e-01 -2.170639 0.029958461
x2        -3.626609e-06 1.274735e-06 -2.844990 0.004441278
R2log = 0.9343495
Excluded as outliers: Eclipse ( 1 / 19 )
MMRE = 13.59051
Pred(25) = 84.21053
Error range = [ -26.8506 .. 56.26816 ]
=====
Speed vs. DIT , NOM_class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.3930354 0.44242555 0.888365 0.37434441
x1         0.4009058 0.18331909 2.186929 0.02874773
x2        -0.1045751 0.05306838 -1.970573 0.04877278
R2log = 0.927589
Excluded as outliers: Eclipse Perl Hibernate Struts ( 4 / 19 )
MMRE = 22.66100
Pred(25) = 68.42105
Error range = [ -40.9379 .. 123.9643 ]
=====
Speed vs. NOC_avg , NOA_class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.2447889 0.33382556 0.7332839 0.46338528
x1         0.3778499 0.16486697 2.2918474 0.02191445
x2        -0.1620681 0.07574333 -2.1397007 0.03237896
R2log = 0.9314903
Excluded as outliers: Eclipse Hibernate Xalan Perl ( 4 / 19 )
MMRE = 22.37267
Pred(25) = 84.21053
Error range = [ -71.57422 .. 122.8487 ]
=====
Speed vs. DIT , NOC_avg , NOA_class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.7258431 0.41116367 1.765339 0.07750688
x1        -0.5864456 0.29914965 -1.960375 0.04995194
x2         0.7447946 0.31800118 2.342113 0.01917493
x3        -0.2037367 0.08285182 -2.459050 0.01393054
R2log = 0.9287745
Excluded as outliers: Eclipse Hibernate Xalan Subversion Struts ( 5
/ 19 )
MMRE = 21.74765
Pred(25) = 78.94737
Error range = [ -79.50218 .. 130.8008 ]
=====
```

### 3.8 Usefulness of the product developer community

```
Community vs. DIT , LOC_class , NOA_class , NOM_class
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -5.01831583 1.758419432 -2.853879 0.004318901
x1          5.42989989 1.774875372  3.059313 0.002218450
x2          0.02652300 0.008735521  3.036224 0.002395614
x3         -0.18569058 0.089154544 -2.082794 0.037269977
x4         -0.26053959 0.083520787 -3.119458 0.001811840
R2log = 0.9352957
Excluded as outliers: xalan Eclipse Mono Perl ( 4 / 19 )
MMRE = 20.25957
Pred(25) = 78.94737
Error range = [ -91.4526 .. 42.36572 ]
=====
```

### 3.9 Documentation Quality

---

```
DocQuality vs. DIT
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 5.300517  2.40524  2.203737 0.02754283
x1        -5.077207  2.16627 -2.343756 0.01909067
R2log = 0.911836
Excluded as outliers: Perl Subversion Mono Eclipse SpringFramework
Struts ( 6 / 19 )
MMRE = 38.66069
Pred(25) = 52.63158
Error range = [ -54.43391 .. 161.1926 ]
=====
```

```
DocQuality vs. NPM_avg
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.3182307 0.07295552 -4.361983 1.288888e-05
x1          0.0604969 0.01403287  4.311084 1.624560e-05
R2log = 0.933459
Excluded as outliers: ( 0 / 19 )
MMRE = 25.59031
Pred(25) = 57.89474
Error range = [ -34.07555 .. 182.2980 ]
=====
```

```
DocQuality vs. DIT , eLOC
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.679398e+00 4.172383e-01  4.025032 5.696747e-05
x1        -1.342252e+00 3.103882e-01 -4.324430 1.529267e-05
x2        -5.832184e-06 1.646693e-06 -3.541755 3.974744e-04
R2log = 0.9308174
Excluded as outliers: Eclipse Hibernate HttpUnit ( 3 / 19 )
MMRE = 26.92643
Pred(25) = 73.68421
Error range = [ -28.68929 .. 235.2284 ]
=====
```

```
DocQuality vs. DIT , NOA_tot
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 6.623799e-01 1.522961e-01  4.349290 0.0000136579
x1        -3.219087e-01 1.192671e-01 -2.699058 0.0069536133
x2        -9.974602e-05 2.533375e-05 -3.937278 0.0000824112
R2log = 0.9486115
Excluded as outliers: xalan Hibernate Httpunit ( 3 / 19 )
MMRE = 32.32802
=====
```

```

Pred(25) = 57.89474
Error range = [ -80.7758 .. 260.6603 ]
=====
DocQuality vs. DIT , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.6974912762 0.1720205533 4.054697 5.019926e-05
x1         -0.3590234092 0.1306731939 -2.747491 6.005318e-03
x2         -0.0004867045 0.0001418686 -3.430671 6.020908e-04
R2log = 0.9301247
Excluded as outliers: Hibernate Eclipse HttpUnit ( 3 / 19 )
MMRE = 30.47274
Pred(25) = 63.1579
Error range = [ -71.75627 .. 248.1265 ]
=====
DocQuality vs. LOC_class , NOA_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1924219508 1.662527e-01 1.157406 2.471064e-01
x1         0.0014126720 5.275031e-04 2.678036 7.405533e-03
x2        -0.0000996399 2.531320e-05 -3.936283 8.275342e-05
R2log = 0.948559
Excluded as outliers: xalan Hibernate HttpUnit ( 3 / 19 )
MMRE = 31.78516
Pred(25) = 63.1579
Error range = [ -81.32186 .. 255.0391 ]
=====
DocQuality vs. eLOC , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.413016e-01 3.290035e-01 2.253173 0.024248259
x1        -3.728005e-06 1.222559e-06 -3.049345 0.002293412
x2        -1.489117e-01 6.892028e-02 -2.160637 0.030723365
R2log = 0.9431522
Excluded as outliers: Subversion Perl Xalan HttpUnit Mono ( 5 / 19 )
MMRE = 36.70243
Pred(25) = 57.89474
Error range = [ -78.99707 .. 276.6553 ]
=====
DocQuality vs. NOC_avg , NOA_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.463870e-01 1.372017e-01 3.982364 6.823315e-05
x1        -3.049559e-01 1.218521e-01 -2.502673 1.232594e-02
x2        -8.543775e-05 2.721482e-05 -3.139383 1.693039e-03
R2log = 0.947922
Excluded as outliers: xalan Hibernate HttpUnit ( 3 / 19 )
MMRE = 34.08316
Pred(25) = 63.1579
Error range = [ -77.22348 .. 273.3272 ]
=====
DocQuality vs. NOC_avg , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8044595 0.32407454 2.482329 0.0130526853
x1        -0.4498690 0.12898487 -3.487765 0.0004870751
x2        -0.1491313 0.06981644 -2.136049 0.0326754179
R2log = 0.943051
Excluded as outliers: xalan HttpUnit Perl Ant ( 4 / 19 )
MMRE = 31.60297
Pred(25) = 73.68421

```

```
Error range = [ -81.7365 .. 251.2494 ]
=====
DocQuality vs. NOC_avg , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.7060823999 0.1656047138 4.263661 2.011043e-05
x1         -0.5725506116 0.1753885502 -3.264470 1.096692e-03
x2         -0.0005324732 0.0001571871 -3.387512 7.052973e-04
R2log = 0.9326148
Excluded as outliers: Hibernate Eclipse Ant Httpunit Saxon ( 5 / 19 )
)
MMRE = 34.41970
Pred(25) = 63.1579
Error range = [ -80.54424 .. 257.9454 ]
=====
DocQuality vs. NOA_tot , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.2245994807 1.588751e-01 1.413686 0.1574541764
x1         -0.0000931313 2.589538e-05 -3.596445 0.0003225958
x2          0.0405978728 1.501036e-02 2.704656 0.0068375073
R2log = 0.9486625
Excluded as outliers: xalan Hibernate Httpunit ( 3 / 19 )
MMRE = 31.64586
Pred(25) = 63.1579
Error range = [ -78.25039 .. 256.1731 ]
=====
DocQuality vs. NOA_tot , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.2037708259 1.724029e-01 1.181946 2.372273e-01
x1         -0.0001133860 2.484085e-05 -4.564499 5.006893e-06
x2          0.0100545146 4.255897e-03 2.362490 1.815264e-02
R2log = 0.9474826
Excluded as outliers: xalan Hibernate Httpunit ( 3 / 19 )
MMRE = 34.56141
Pred(25) = 57.89474
Error range = [ -85.50784 .. 279.1381 ]
=====
DocQuality vs. NOA_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.83287778 0.37422666 2.225597 0.026041179
x1         -0.22498248 0.07930681 -2.836862 0.004555928
x2          0.04585352 0.01560644 2.938116 0.003302132
R2log = 0.9307277
Excluded as outliers: xalan Httpunit Eclipse Hibernate ( 4 / 19 )
MMRE = 35.14279
Pred(25) = 63.1579
Error range = [ -89.80825 .. 275.6048 ]
=====
DocQuality vs. num_classes , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0529935005 0.1899580552 0.2789747 0.780264204
x1         -0.0002919926 0.0001329153 -2.1968324 0.028032415
x2          0.0472390905 0.0157420432 3.0008233 0.002692507
R2log = 0.9477819
Excluded as outliers: Hibernate Httpunit Struts ( 3 / 19 )
MMRE = 27.30236
Pred(25) = 68.42105
```

```
Error range = [ -49.09187 .. 226.4560 ]
=====
DocQuality vs. NOM_tot , NPM_avg
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 3.596449e-02 0.1852560729  0.1941339 0.846071012
x1          -2.857308e-05 0.0000132269 -2.1602248 0.030755269
x2           4.743259e-02 0.0157037320  3.0204659 0.002523861
R2log = 0.9477115
Excluded as outliers: Hibernate Struts HttpUnit ( 3 / 19 )
MMRE = 27.24583
Pred(25) = 68.42105
Error range = [ -43.93959 .. 222.248 ]
=====

DocQuality vs. CBO , DIT , NOA_class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.4679609 0.38033337  3.859669 0.0001135407
x1          -0.1329709 0.04820567 -2.758407 0.0058083843
x2           -0.3761452 0.12871057 -2.922411 0.0034733269
x3           -0.1553328 0.06793296 -2.286560 0.0222214979
R2log = 0.9340693
Excluded as outliers: Eclipse Xalan HttpUnit ( 3 / 19 )
MMRE = 36.12909
Pred(25) = 57.89474
Error range = [ -99.76374 .. 242.9392 ]
=====

DocQuality vs. CBO , NOC_avg , num_classes
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.150884770 0.2621948903  0.575468 0.564974825
x1          0.370464194 0.1747965167  2.119403 0.034056456
x2          -1.066095579 0.3538385531 -3.012944 0.002587271
x3          -0.001086714 0.0003436431 -3.162333 0.001565107
R2log = 0.9180141
Excluded as outliers: Eclipse Hibernate Subversion HttpUnit Ant ( 5
/ 19 )
MMRE = 43.93054
Pred(25) = 63.1579
Error range = [ -94.78446 .. 333.0698 ]
=====

DocQuality vs. CBO , NOA_class , RFC
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.07322347 0.363923014  2.949040 0.0031876286
x1          -0.17563569 0.050071589 -3.507692 0.0004520127
x2          -0.17878535 0.069427891 -2.575123 0.0100204498
x3           0.01387892 0.004905174  2.829445 0.0046628783
R2log = 0.9336109
Excluded as outliers: Eclipse Xalan HttpUnit ( 3 / 19 )
MMRE = 36.79397
Pred(25) = 57.89474
Error range = [ -99.97306 .. 259.4721 ]
=====

DocQuality vs. DIT , LOC_class , NOM_class
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.96285702 0.71285655 -2.753509 0.0058960127
x1           2.62756194 0.81106284  3.239653 0.0011967535
x2           0.01393058 0.00365093  3.815626 0.0001358383
x3           -0.23151450 0.06195684 -3.736706 0.0001864469
=====
```

```
R2log = 0.9261612
Excluded as outliers: Eclipse ( 1 / 19 )
MMRE = 22.3862
Pred(25) = 68.42105
Error range = [ -30.01002 .. 103.6979 ]
=====
DocQuality vs. DIT , eLOC , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.811307e+00 4.223608e-01 4.288530 1.798599e-05
x1          -9.171743e-01 1.780592e-01 -5.150953 2.591665e-07
x2          -3.549420e-06 1.046773e-06 -3.390820 6.968384e-04
x3          -1.546001e-01 6.797397e-02 -2.274401 2.294189e-02
R2log = 0.9491078
Excluded as outliers: xalan HttpUnit ( 2 / 19 )
MMRE = 30.09908
Pred(25) = 73.68421
Error range = [ -78.92565 .. 277.8256 ]
=====
DocQuality vs. DIT , NOA_class , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.3189939076 0.3682168555 3.582112 0.000340828
x1          -0.3460203003 0.1216454179 -2.844499 0.004448129
x2          -0.1603338021 0.0775052672 -2.068683 0.038575876
x3          -0.0003808496 0.0001177822 -3.233507 0.001222805
R2log = 0.9494587
Excluded as outliers: Hibernate xalan HttpUnit ( 3 / 19 )
MMRE = 33.41
Pred(25) = 68.42105
Error range = [ -81.75902 .. 290.9034 ]
=====
DocQuality vs. LOC_class , NOA_class , num_classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8427398100 0.3542733105 2.378784 0.017369834
x1          0.0015687218 0.0005417703 2.895548 0.003784974
x2          -0.1699701800 0.0778069667 -2.184511 0.028924715
x3          -0.0003748264 0.0001180413 -3.175384 0.001496385
R2log = 0.9496783
Excluded as outliers: Hibernate Xalan Httpunit ( 3 / 19 )
MMRE = 32.85275
Pred(25) = 68.42105
Error range = [ -84.36783 .. 286.4350 ]
=====
DocQuality vs. NOA_class , num_classes , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8214573203 0.3556102300 2.309994 0.020888514
x1          -0.1533069733 0.0770613200 -1.989415 0.046655376
x2          -0.0003530395 0.0001204523 -2.930950 0.003379272
x3          0.0436461298 0.0152477646 2.862461 0.004203651
R2log = 0.9495594
Excluded as outliers: Hibernate Xalan HttpUnit ( 3 / 19 )
MMRE = 32.78596
Pred(25) = 68.42105
Error range = [ -80.1802 .. 285.1971 ]
=====
DocQuality vs. NOA_class , num_classes , RFC
      Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) 0.8715757991 0.3556425971 2.450707 0.0142575961
x1         -0.1720720719 0.0785079572 -2.191779 0.0283954866
x2         -0.0004412438 0.0001142038 -3.863651 0.0001117048
x3          0.0108696931 0.0043642178 2.490639 0.0127513447
R2log = 0.9482047
Excluded as outliers: Hibernate Xalan HttpUnit (3 / 19)
MMRE = 35.8614
Pred(25) = 68.42105
Error range = [ -84.54491 .. 311.6622 ]
=====
DocQuality vs. NOA_class , NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.0782997 0.56267110 -1.916394 0.0553149209
x1          0.2023511 0.09336814 2.167240 0.0302165729
x2         -0.1729703 0.04976842 -3.475503 0.0005098959
x3          1.7163604 0.65395019 2.624604 0.0086749735
R2log = 0.9424738
Excluded as outliers: Perl Subversion Mono Xalan SpringFramework
Xerces (6 / 19)
MMRE = 41.7257
Pred(25) = 52.63158
Error range = [ -36.73217 .. 162.4956 ]
=====
DocQuality vs. DIT , LOC_class , NOA_tot , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.709249e+00 2.592978e+00 2.201812 0.02767859
x1        -7.428913e+00 3.447910e+00 -2.154613 0.03119212
x2        -4.671239e-02 2.061756e-02 -2.265661 0.02347216
x3        -3.743203e-05 1.740177e-05 -2.151048 0.03147245
x4          6.279168e-01 3.076132e-01 2.041255 0.04122553
R2log = 0.9211576
Excluded as outliers: Eclipse HttpUnit Perl Subversion Struts
JFreeChart (6 / 19)
MMRE = 52.49287
Pred(25) = 57.89474
Error range = [ -98.334 .. 451.3785 ]
=====
DocQuality vs. LOC_class , num_abstract_classes , NOM_class ,
NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.993515944 0.645614553 1.538869 0.123836386
x1        -0.041394962 0.013645425 -3.033615 0.002416429
x2        -0.003491686 0.001745327 -2.000591 0.045436432
x3          0.461775075 0.176116975 2.621979 0.008742072
x4        -2.421235992 1.223254319 -1.979340 0.047777758
R2log = 0.9508642
Excluded as outliers: HttpUnit Subversion Perl Mono JasperReports (5 / 18)
MMRE = 53.28058
Pred(25) = 72.22222
Error range = [ -100 .. 452.1754 ]
=====
```

### 3.10 Trustworthiness with respect to non Open Source (closed source) products

```
CssCompetitors vs. DIT
    Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.1437290  0.1444319  7.918814 2.397861e-15
x1          -0.4466761  0.1398885 -3.193086 1.407611e-03
R2log = 0.935111
Excluded as outliers: Mono ( 1 / 19 )
MMRE = 14.49223
Pred(25) = 78.94737
Error range = [ -18.27618 .. 51.66207 ]
=====
CssCompetitors vs. NOC_avg
    Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 0.1726643  0.1944626  0.8879051 0.37459179
x1          0.5176016  0.2172251  2.3827887 0.01718205
R2log = 0.8988215
Excluded as outliers: Subversion Perl Eclipse ( 3 / 19 )
MMRE = 15.01813
Pred(25) = 78.94737
Error range = [ -28.11849 .. 42.41935 ]
=====
CssCompetitors vs. DIT , NOC_avg
    Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 1.0349997  0.1350992  7.661035 1.844400e-14
x1          -0.6883398  0.2126237 -3.237362 1.206404e-03
x2          0.3893296  0.1950353  1.996201 4.591210e-02
R2log = 0.9325078
Excluded as outliers: ( 0 / 19 )
MMRE = 13.75802
Pred(25) = 84.21053
Error range = [ -17.21351 .. 47.56928 ]
=====
CssCompetitors vs. DIT , NPM_avg
    Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 4.479621  2.0727501  2.161197 0.030680116
x1          -5.316160  1.9027273 -2.793968 0.005206561
x2          2.304726  0.9486427  2.429499 0.015119709
R2log = 0.9124765
Excluded as outliers: Subversion Perl Mono Eclipse JFreeChart ( 5 /
19 )
MMRE = 18.99531
Pred(25) = 73.68421
Error range = [ -35.47561 .. 100 ]
=====
CssCompetitors vs. eLOC , NOC_avg
    Estimate Std. Error   z value   Pr(>|z|)
(Intercept) 2.299333e-01 2.086647e-01  1.101927 0.27049330
x1          -3.395373e-06 1.724595e-06 -1.968795 0.04897664
x2          7.148023e-01 2.678129e-01  2.669036 0.00760692
R2log = 0.9320783
Excluded as outliers: Subversion Perl Mono ( 3 / 19 )
MMRE = 17.40498
Pred(25) = 73.68421
Error range = [ -54.31288 .. 33.74327 ]
=====
CssCompetitors vs. NOC_avg , NPM_avg
    Estimate Std. Error   z value   Pr(>|z|)
```

```
(Intercept) -1.957704  1.0134814 -1.931663  0.05340117
x1          0.741721  0.3311419  2.239888  0.02509817
x2          2.092774  0.9600741  2.179805  0.02927195
R2log =  0.9076888
Excluded as outliers: Subversion Perl Mono JFreeChart Eclipse Struts
( 6 / 19 )
MMRE =  20.40786
Pred(25) =  78.94737
Error range = [ -20.13194 .. 99.99953 ]
=====
CssCompetitors vs. NOM_class , NPM_avg
      Estimate Std. Error   z value Pr(>|z|)
(Intercept)  0.8131470  0.4309850  1.886718  0.05919832
x1          -0.1519168  0.0601533 -2.525494  0.01155358
x2           1.3859929  0.6986342  1.983861  0.04727139
R2log =  0.9325387
Excluded as outliers: Subversion Perl Mono Struts ( 4 / 19 )
MMRE =  18.75357
Pred(25) =  73.68421
Error range = [ -20.89437 .. 99.98653 ]
=====
CssCompetitors vs. CBO , DIT , eLOC
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) 1.752994e+00 3.627305e-01  4.832772 1.346447e-06
x1          1.550459e-01 6.956129e-02  2.228910 2.581987e-02
x2          -1.266581e+00 3.396434e-01 -3.729150 1.921270e-04
x3          -5.303589e-06 1.853458e-06 -2.861457 4.216988e-03
R2log =  0.9181042
Excluded as outliers: Eclipse ( 1 / 19 )
MMRE =  12.75812
Pred(25) =  84.21053
Error range = [ -14.12123 .. 51.49709 ]
=====
CssCompetitors vs. CBO , LOC_class , num_classes
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.0004707295 0.2735400395 -0.001720880 0.998626937
x1          0.3312303151 0.1111724404  2.979428301 0.002887868
x2          0.0020022083 0.0006833384  2.930039252 0.003389192
x3          -0.0009895835 0.0003246508 -3.048147333 0.002302570
R2log =  0.9191811
Excluded as outliers: Eclipse Hibernate HttpUnit Findbugs Ant ( 5 /
19 )
MMRE =  19.48783
Pred(25) =  73.68421
Error range = [ -87.4929 .. 51.77117 ]
=====
CssCompetitors vs. CBO , num_classes , NPM_avg
      Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.0108418500 0.2733203774 -0.03966719 0.968358464
x1          0.3577090748 0.1122598365  3.18643859 0.001440361
x2          -0.0009996757 0.0003250814 -3.07515488 0.002103933
x3           0.0584237648 0.0192728474  3.03140287 0.002434202
R2log =  0.9200359
Excluded as outliers: Eclipse Hibernate HttpUnit Findbugs Ant ( 5 /
19 )
MMRE =  19.72963
```

```

Pred(25) = 73.68421
Error range = [ -86.94322 .. 53.669 ]
=====
CssCompetitors vs. LOC_class , eLOC , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.696175e+00 4.363672e-01 3.887035 0.0001014761
x1          4.035776e-03 1.072830e-03 3.761802 0.0001686934
x2         -3.259009e-06 1.320332e-06 -2.468326 0.0135746777
x3         -1.152094e-01 4.422591e-02 -2.605021 0.0091868508
R2log = 0.936664
Excluded as outliers: struts (1 / 19)
MMRE = 11.81559
Pred(25) = 78.94737
Error range = [ -15.14697 .. 44.33873 ]
=====
CssCompetitors vs. LOC_class , NOC_avg , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.553236729 0.317420629 1.742914 0.0813487087
x1          0.003977344 0.001042093 3.816687 0.0001352553
x2          0.615072112 0.243506652 2.525894 0.0115404134
x3         -0.079461654 0.038487008 -2.064636 0.0389574731
R2log = 0.9176858
Excluded as outliers: Eclipse (1 / 19)
MMRE = 12.41722
Pred(25) = 84.21053
Error range = [ -18.10814 .. 40.87262 ]
=====
CssCompetitors vs. LOC_class , NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.08606014 0.69402324 1.564876 0.117612045
x1          0.02960836 0.01184741 2.499143 0.012449420
x2         -0.46399279 0.14557734 -3.187260 0.001436277
x3          2.03649885 0.88416896 2.303292 0.021262447
R2log = 0.9148677
Excluded as outliers: Perl subversion Mono Struts Eclipse HttpUnit (6 / 19)
MMRE = 18.19113
Pred(25) = 73.68421
Error range = [ -33.26122 .. 100 ]
=====
CssCompetitors vs. NOC_avg , NOA_tot , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.294542e+00 9.060903e-01 -2.532355 0.0113299215
x1          1.359853e+00 4.011515e-01 3.389874 0.0006992483
x2         -4.086605e-05 2.003193e-05 -2.040045 0.0413458092
x3          2.068821e+00 7.478790e-01 2.766251 0.0056704914
R2log = 0.9436392
Excluded as outliers: Perl subversion Mono JFreeChart Struts Findbugs (6 / 19)
MMRE = 18.60225
Pred(25) = 68.42105
Error range = [ -34.97817 .. 99.9989 ]
=====
CssCompetitors vs. num_abstract_classes , NOA_tot , NOA_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.1165152460 5.472809e-01 3.867329 0.0001100341

```

```

x1      -0.0218368736 6.900719e-03 -3.164435 0.0015538462
x2      0.0001840121 7.968543e-05  2.309232 0.0209307374
x3      -0.3083698590 1.391138e-01 -2.216674 0.0266454002
R2log = 0.9005944
Excluded as outliers: Subversion Mono Ant Hibernate Eclipse ( 5 / 18 )
)
MMRE = 17.84377
Pred(25) = 66.66667
Error range = [ -48.55602 .. 75.72637 ]
=====
CssCompetitors vs. num_abstract_classes , NOA_class , num_classes
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.6912597173 0.442970447 3.817997 0.0001345397
x1          -0.0217462691 0.006447623 -3.372757 0.0007441951
x2          -0.2246642012 0.103386669 -2.173048 0.0297766994
x3          0.0009579255 0.000339281  2.823399 0.0047517453
R2log = 0.937157
Excluded as outliers: xalan Ant Mono Subversion Hibernate ( 5 / 18 )
MMRE = 19.16762
Pred(25) = 72.22222
Error range = [ -79.04625 .. 75.44633 ]
=====
CssCompetitors vs. num_abstract_classes , num_classes , NOM_class
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.7227869456 0.4489647897 3.837243 0.0001244234
x1          -0.0142765782 0.0061651445 -2.315692 0.0205750707
x2          0.0005232428 0.0002526464  2.071048 0.0383543169
x3          -0.0994379792 0.0434188545 -2.290203 0.0220095670
R2log = 0.9346103
Excluded as outliers: Perl Subversion Mono Ant ( 4 / 18 )
MMRE = 13.24973
Pred(25) = 94.44444
Error range = [ -18.64748 .. 72.06345 ]
=====
CssCompetitors vs. num_abstract_classes , num_classes , RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.8254993844 0.4005170860 4.557856 5.167834e-06
x1          -0.0130060884 0.0056245971 -2.312359 2.075790e-02
x2          0.0004688211 0.0002284782  2.051929 4.017660e-02
x3          -0.0561115470 0.0176593668 -3.177438 1.485823e-03
R2log = 0.9328286
Excluded as outliers: Subversion Perl Ant ( 3 / 18 )
MMRE = 13.05055
Pred(25) = 88.88889
Error range = [ -56.7418 .. 20.09059 ]
=====
CssCompetitors vs. num_abstract_classes , NOM_tot , RFC
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 2.911289e+00 7.317075e-01 3.978761 0.0000692753
x1          -2.247386e-02 8.099466e-03 -2.774734 0.0055246904
x2          5.524749e-05 2.561362e-05  2.156958 0.0310089660
x3          -8.574168e-02 2.453855e-02 -3.494162 0.0004755523
R2log = 0.9304233
Excluded as outliers: Subversion Perl Ant Hibernate Log4J ( 5 / 18 )
MMRE = 16.24853
Pred(25) = 88.88889

```

```
Error range = [ -71.30661 .. 23.02565 ]
=====
CssCompetitors vs. CBO , eLOC , NOM_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.108287e+00 5.215418e-01 2.125021 0.0335848650
x1          1.921109e-01 8.850113e-02 2.170717 0.0299525351
x2         -5.097758e-06 1.707540e-06 -2.985440 0.0028317038
x3          -1.126090e-01 4.458328e-02 -2.525813 0.0115430948
x4          1.578174e-01 4.059001e-02 3.888084 0.0001010388
R2log = 0.9206606
Excluded as outliers: Eclipse Struts ( 2 / 19 )
MMRE = 11.5023
Pred(25) = 84.21053
Error range = [ -20.07521 .. 51.541 ]
=====
```

### 3.11 Trustworthiness with respect to Open Source products

```
=====
OssCompetitors vs. NOC_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2082583 0.3356401 -0.620481 0.534941176
x1          0.9015753 0.3412375 2.642076 0.008239963
R2log = 0.9091802
Excluded as outliers: Subversion Eclipse Perl Struts SpringFramework
Mono ( 6 / 19 )
MMRE = 15.8684
Pred(25) = 84.21053
Error range = [ -39.28659 .. 32.54347 ]
=====

OssCompetitors vs. LOC_class , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.40554604 0.396638942 3.543641 0.0003946421
x1          0.02660823 0.008778278 3.031145 0.0024362819
x2         -0.27711737 0.085240042 -3.251023 0.0011499042
R2log = 0.9140257
Excluded as outliers: Perl subversion Mono Eclipse SpringFramework ( 5 / 19 )
MMRE = 18.23934
Pred(25) = 73.68421
Error range = [ -27.14013 .. 77.58913 ]
=====

OssCompetitors vs. DIT , LOC_class , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.48734938 2.663330354 2.811273 0.004934586
x1         -5.99624197 2.273424476 -2.637537 0.008351043
x2          0.01789952 0.007064122 2.533863 0.011281276
x3         -1.53869620 0.531897956 -2.892841 0.003817745
R2log = 0.9281545
Excluded as outliers: Perl SpringFramework Struts Eclipse Weka ( 5 /
19 )
MMRE = 18.90718
Pred(25) = 73.68421
Error range = [ -99.44842 .. 44.58648 ]
=====
```

```
OssCompetitors vs. CBO , DIT , eLOC , NPM_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.557558e+00 2.884176e+00 2.967072 0.003006504
x1          -4.571151e-01 1.645090e-01 -2.778664 0.005458295
x2          -8.125457e+00 2.707119e+00 -3.001515 0.002686399
x3           9.007565e-06 3.766258e-06  2.391649 0.016772894
x4          2.606437e+00 1.000374e+00  2.605462 0.009175054
R2log = 0.9181163
Excluded as outliers: Eclipse Perl Subversion Mono Saxon
SpringFramework (6 / 19)
MMRE = 24.14073
Pred(25) = 73.68421
Error range = [ -100 .. 77.77778 ]
=====
OssCompetitors vs. CBO , eLOC , NOA_class , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.526096e-01 4.900212e-01 1.535872 0.12456989
x1          2.752658e-01 1.171049e-01 2.350592 0.01874357
x2          -2.859612e-06 1.396508e-06 -2.047688 0.04059054
x3           2.670564e-01 1.106122e-01 2.414349 0.01576336
x4          -2.378693e-01 8.511025e-02 -2.794838 0.00519258
R2log = 0.9118133
Excluded as outliers: Eclipse Perl xalan SpringFramework Subversion
JFreeChart (6 / 19)
MMRE = 19.54657
Pred(25) = 73.68421
Error range = [ -63.92 .. 53.88595 ]
=====
```

### 3.12 Trustworthiness

```
=====
Trustworthiness vs. CBO
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.0898005 0.2500808 4.357793 1.313805e-05
x1          -0.1206116 0.0562384 -2.144648 3.198103e-02
R2log = 0.9198285
Excluded as outliers: Eclipse Perl saxon (3 / 19)
MMRE = 17.77963
Pred(25) = 73.68421
Error range = [ -99.39152 .. 52.6094 ]
=====
Trustworthiness vs. DIT
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.974197  2.322762  3.002545 0.002677323
x1          -5.725856  2.083825 -2.747762 0.006000347
R2log = 0.9074196
Excluded as outliers: subversion Perl Mono SpringFramework Eclipse
Struts (6 / 19)
MMRE = 23.19333
Pred(25) = 63.1579
Error range = [ -35.23401 .. 92.12724 ]
=====
Trustworthiness vs. eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.044713e-01 1.194857e-01 7.569701 3.740833e-14
```

```

x1      -2.603099e-06 9.396003e-07 -2.770432 5.598198e-03
R2log = 0.9370628
Excluded as outliers: Perl Subversion HttpUnit ( 3 / 19 )
MMRE = 13.23626
Pred(25) = 89.47368
Error range = [ -21.31357 .. 66.54776 ]
=====
Trustworthiness vs. NOA_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.66779e-01 1.340631e-01 6.465454 1.009952e-10
x1         -4.33111e-05 2.035776e-05 -2.127498 3.337871e-02
R2log = 0.9425177
Excluded as outliers: Perl HttpUnit xalan ( 3 / 19 )
MMRE = 13.61876
Pred(25) = 89.47368
Error range = [ -23.75661 .. 64.84206 ]
=====
Trustworthiness vs. NOM_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.032448e+00 1.679008e-01 6.149156 7.789617e-10
x1         -3.632595e-05 1.481411e-05 -2.452118 1.420182e-02
R2log = 0.906677
Excluded as outliers: Hibernate Perl Saxon Eclipse HttpUnit
Subversion ( 6 / 19 )
MMRE = 16.39437
Pred(25) = 78.94737
Error range = [ -38.47132 .. 68.59724 ]
=====
Trustworthiness vs. RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.17351870 0.26169090 4.484370 7.312983e-06
x1         -0.02734316 0.01247506 -2.191827 2.839203e-02
R2log = 0.9007448
Excluded as outliers: Subversion Perl Eclipse ( 3 / 19 )
MMRE = 14.71718
Pred(25) = 78.94737
Error range = [ -31.40370 .. 51.89447 ]
=====
Trustworthiness vs. CBO , DIT
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.6071625 2.36605573 3.215124 0.001303882
x1         -0.1465589 0.06215524 -2.357949 0.018376197
x2         -5.7262202 2.04798543 -2.796026 0.005173526
R2log = 0.9167052
Excluded as outliers: Eclipse Perl subversion Mono SpringFramework
Saxon ( 6 / 19 )
MMRE = 25.82200
Pred(25) = 63.1579
Error range = [ -99.90245 .. 92.1479 ]
=====
Trustworthiness vs. CBO , NOC_avg
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.9951516 0.25053969 3.972032 7.126224e-05
x1         -0.2194019 0.07270511 -3.017695 2.547047e-03
x2          0.6123313 0.26158818 2.340822 1.924136e-02
R2log = 0.9099186

```

```

Excluded as outliers: Eclipse Perl Subversion Saxon ( 4 / 19 )
MMRE = 18.54393
Pred(25) = 78.94737
Error range = [ -99.9947 .. 45.6629 ]
=====
Trustworthiness vs. NOC_avg , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.4144994 0.42551450 3.324210 0.0008866951
x1          0.4605055 0.18572388 2.479517 0.0131560588
x2         -0.1199083 0.05238826 -2.288840 0.0220886634
R2log = 0.916493
Excluded as outliers: Perl Eclipse Hibernate JFreeChart ( 4 / 19 )
MMRE = 15.77349
Pred(25) = 84.21053
Error range = [ -38.98247 .. 41.84654 ]
=====
Trustworthiness vs. num_abstract_classes , NOM_tot
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.800744e-01 1.763134e-01 5.558706 2.717817e-08
x1          5.363685e-03 2.640211e-03 2.031536 4.220060e-02
x2         -5.542834e-05 1.860861e-05 -2.978640 2.895307e-03
R2log = 0.9472414
Excluded as outliers: Hibernate Perl Saxon HttpUnit ( 4 / 18 )
MMRE = 16.13976
Pred(25) = 77.77778
Error range = [ -38.67931 .. 72.01058 ]
=====
Trustworthiness vs. CBO , num_classes , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.1238816221 0.3664302478 3.067109 0.002161398
x1          0.3994356425 0.1634344572 2.444011 0.014524976
x2         -0.0007405181 0.0002854275 -2.594418 0.009475125
x3         -0.1255450707 0.0566894116 -2.214612 0.026786684
R2log = 0.9107728
Excluded as outliers: Eclipse Perl Subversion Hibernate HttpUnit
Xalan ( 6 / 19 )
MMRE = 21.28769
Pred(25) = 68.42105
Error range = [ -53.22242 .. 80.83116 ]
=====
Trustworthiness vs. CBO , LOC_class , num_classes , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.2361483845 0.3738102688 3.306887 0.0009433888
x1          0.3708361471 0.1485368880 2.496593 0.0125392802
x2         -0.0036123567 0.0012017100 -3.006014 0.0026469707
x3         -0.0005900905 0.0002621771 -2.250733 0.0244024847
x4         -0.1095336517 0.0508753102 -2.152982 0.0313200552
R2log = 0.9308123
Excluded as outliers: Eclipse Perl Xalan HttpUnit Hibernate ( 5 / 19
)
MMRE = 19.47218
Pred(25) = 73.68421
Error range = [ -64.88881 .. 82.64803 ]
=====
Trustworthiness vs. CBO , eLOC , NOA_class , NOM_class
      Estimate Std. Error z value Pr(>|z|)

```

```
(Intercept) 1.403783e+00 7.003792e-01 2.004318 4.503600e-02
x1          3.194640e-01 1.146557e-01 2.786290 5.331520e-03
x2          -4.616468e-06 1.185942e-06 -3.892660 9.915098e-05
x3          3.359901e-01 1.076638e-01 3.120735 1.804005e-03
x4          -3.463321e-01 9.313220e-02 -3.718715 2.002390e-04
R2log = 0.937181
Excluded as outliers: Eclipse Perl Xalan SpringFramework JFreeChart
Struts ( 6 / 19 )
MMRE = 21.59807
Pred(25) = 73.68421
Error range = [ -81.4272 .. 81.27204 ]
=====
Trustworthiness vs. CBO , NOA_class , num_classes , NOM_class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.4092283188 0.3864543688 3.646558 0.0002657764
x1          0.3836338350 0.1552724665 2.470714 0.0134843666
x2          -0.0998738591 0.0413426434 -2.415759 0.0157024546
x3          -0.0006656367 0.0002732111 -2.436346 0.0148364874
x4          -0.1081264903 0.0528521288 -2.045830 0.0407730628
R2log = 0.9111478
Excluded as outliers: Eclipse Perl Subversion HttpUnit Hibernate ( 5
/ 19 )
MMRE = 17.34947
Pred(25) = 68.42105
Error range = [ -38.05296 .. 88.17485 ]
=====
```

## 4 ANALYSIS OF C++ PRODUCTS VS. SPECIFIC CODE MEASURES

In this section the correlation between subjectively evaluated product qualities and static C++ code measures is studied.

The explanation of the measure names whose meaning is not obvious is given in appendix 7.3.

### 4.1 Reliability

```
=====
Reliability vs. ACC
    Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.91346579 0.19354873  4.719565 2.363497e-06
x1          -0.08611095 0.03982475 -2.162247 3.059913e-02
R2log = 0.9525058
Excluded as outliers: 23 6 33 7 ( 4 / 14 )
MMRE = 39.41277
Pred(25) = 57.14286
Error range = [ -45.24351 .. 208.9873 ]
=====

=====
Reliability vs. CBO
    Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.8901774 0.18135041  4.908604 9.172675e-07
x1          -0.0836722 0.03820374 -2.190157 2.851282e-02
R2log = 0.9525867
Excluded as outliers: 23 6 33 7 ( 4 / 14 )
MMRE = 38.96113
Pred(25) = 57.14286
Error range = [ -45.03412 .. 206.9204 ]
=====

=====
Reliability vs. Num. attributes per class
    Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.65761181 0.097243743  6.762510 1.356210e-11
x1          -0.01338205 0.006239805 -2.144626 3.198277e-02
R2log = 0.945489
Excluded as outliers: 23 6 33 ( 3 / 14 )
MMRE = 33.2816
Pred(25) = 64.28571
Error range = [ -36.54978 .. 164.7415 ]
=====

=====
Reliability vs. eLOC per method , Num. attributes per class
    Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.34960280 0.142589082  2.451820 0.014213565
x1          0.02282025 0.008222203  2.775443 0.005512660
x2          -0.02346744 0.007960171 -2.948107 0.003197262
R2log = 0.948413
Excluded as outliers: 23 6 ( 2 / 14 )
MMRE = 28.94756
Pred(25) = 64.28571
Error range = [ -43.50952 .. 118.5081 ]
=====

=====
Reliability vs. eLOC per method , RFC
    Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.328125954 0.149166845  2.199724 0.02782645
x1          0.022082234 0.009124209  2.420181 0.01551279
x2          -0.004489265 0.001900828 -2.361741 0.01818932
R2log = 0.9468959
```

```
Excluded as outliers: 23 6 21 ( 3 / 14 )
MMRE = 32.00535
Pred(25) = 57.14286
Error range = [ -31.60512 .. 152.2526 ]
=====
```

## 4.2 Usability

---

Usability vs. LCOM

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.43681770	0.14935508	-2.924693	0.0034479655
x1	0.08995364	0.02365335	3.802997	0.0001429559

R2log = 0.954874  
 Excluded as outliers: 26 23 12 11 ( 4 / 14 )  
 MMRE = 12.99803  
 Pred(25) = 78.57143  
 Error range = [ -31.04423 .. 48.1741 ]

---

Usability vs. Num. attributes per class

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.22204403	0.11586654	-1.916378	0.05531705
x1	0.01957454	0.00776761	2.520021	0.01173480

R2log = 0.952939  
 Excluded as outliers: 23 11 26 31 ( 4 / 14 )  
 MMRE = 16.01321  
 Pred(25) = 85.71429  
 Error range = [ -24.89848 .. 52.41722 ]

---

Usability vs. Num. attribute per method

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.4913189	0.2228903	-2.204308	0.02750268
x1	0.3214012	0.1443871	2.225969	0.02601626

R2log = 0.9513997  
 Excluded as outliers: 23 31 26 ( 3 / 14 )  
 MMRE = 15.73335  
 Pred(25) = 78.57143  
 Error range = [ -31.82454 .. 45.60825 ]

---

Usability vs. ACC , eLOC

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	8.724003e-01	1.823106e-01	4.785242	1.707811e-06
x1	-2.192346e-01	4.901286e-02	-4.473001	7.712934e-06
x2	4.396464e-07	1.850783e-07	2.375462	1.752702e-02

R2log = 0.9633624  
 Excluded as outliers: 23 33 6 ( 3 / 14 )  
 MMRE = 16.19986  
 Pred(25) = 78.57143  
 Error range = [ -57.2537 .. 64.96081 ]

---

Usability vs. ACC , Num. methods

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	8.072334e-01	1.762147e-01	4.580965	4.628353e-06
x1	-2.205313e-01	4.662974e-02	-4.729414	2.251686e-06
x2	2.289943e-05	8.404434e-06	2.724684	6.436299e-03

R2log = 0.9644928  
 Excluded as outliers: 23 6 33 ( 3 / 14 )  
 MMRE = 16.99904  
 Pred(25) = 78.57143  
 Error range = [ -52.46419 .. 77.16923 ]

---

Usability vs. ACC , Num. public methods

```

                Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.158145e-01 1.766366e-01 4.618603 3.863314e-06
x1          -2.235443e-01 4.734560e-02 -4.721543 2.340623e-06
x2           2.358181e-05 8.657949e-06  2.723718 6.455168e-03
R2log = 0.964489
Excluded as outliers: 23 6 33 ( 3 / 14 )
MMRE = 17.05076
Pred(25) = 78.57143
Error range = [ -52.7305 .. 77.87682 ]
=====
=====

Usability vs. CBO , eLOC
                Estimate Std. Error z value Pr(>|z|)
(Intercept) 8.028734e-01 1.668030e-01 4.813303 1.484562e-06
x1          -2.003409e-01 4.391046e-02 -4.562487 5.055133e-06
x2           3.555104e-07 1.720663e-07  2.066124 3.881676e-02
R2log = 0.9638507
Excluded as outliers: 23 33 6 ( 3 / 14 )
MMRE = 14.58373
Pred(25) = 78.57143
Error range = [ -54.84987 .. 60.24986 ]
=====
=====

Usability vs. CBO , Num. methods
                Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.369666e-01 1.650452e-01 4.465241 7.997860e-06
x1          -1.944938e-01 4.136683e-02 -4.701684 2.580240e-06
x2           1.724152e-05 7.733264e-06  2.229527 2.577885e-02
R2log = 0.9642966
Excluded as outliers: 23 33 6 ( 3 / 14 )
MMRE = 16.76150
Pred(25) = 71.42857
Error range = [ -49.82752 .. 68.89863 ]
=====
=====

Usability vs. CBO , Num. public methods
                Estimate Std. Error z value Pr(>|z|)
(Intercept) 7.434389e-01 1.649598e-01 4.506788 6.581641e-06
x1          -1.970393e-01 4.191890e-02 -4.700488 2.595404e-06
x2           1.782232e-05 7.951941e-06  2.241255 2.500959e-02
R2log = 0.9643288
Excluded as outliers: 23 33 6 ( 3 / 14 )
MMRE = 16.78464
Pred(25) = 71.42857
Error range = [ -50.08411 .. 69.5543 ]
=====
```

### 4.3 Portability

```

Portability vs. Num. methods per class
                Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.16764556 0.25023579 -4.666181 3.068490e-06
x1           0.06720192 0.01108669  6.061496 1.348615e-09
R2log = 0.9577369
Excluded as outliers: 26 12 30 21 ( 4 / 14 )
MMRE = 17.95409
Pred(25) = 64.28571
Error range = [ -52.7869 .. 43.36979 ]
=====
=====

Portability vs. LCOM , McCabe
                Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1377386 0.20605546  0.6684538 5.038439e-01
x1           0.1053370 0.02571181  4.0968321 4.188425e-05
x2           -0.1776655 0.07301470 -2.4332841 1.496255e-02
R2log = 0.9505414
```

```
Excluded as outliers: 6 12 38 26 ( 4 / 14 )
MMRE = 21.25693
Pred(25) = 71.42857
Error range = [ -28.87038 .. 101.1187 ]
=====
```

#### 4.4 How Well Are Functional Requirements Satisfied

---

```
Functionality vs. ACC
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.09495566 0.15316421 -0.6199598 0.535284240
x1          0.09262021 0.02938224  3.1522513 0.001620168
R2log = 0.9519274
Excluded as outliers: 23 11 31 26 ( 4 / 14 )
MMRE = 14.81292
Pred(25) = 85.71429
Error range = [ -17.42741 .. 55.07268 ]
=====
```

```
Functionality vs. CBO
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.09479071 0.15298589 -0.6196043 0.535518334
x1          0.09265841 0.02936296  3.1556217 0.001601564
R2log = 0.9519462
Excluded as outliers: 23 11 26 31 ( 4 / 14 )
MMRE = 14.96079
Pred(25) = 85.71429
Error range = [ -17.44796 .. 55.08617 ]
=====
```

```
Functionality vs. eLOC per method
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.11039782 0.153035437 -0.7213873 0.470671271
x1          0.01915716 0.006123815  3.1283050 0.001758176
R2log = 0.9525849
Excluded as outliers: 23 26 31 ( 3 / 14 )
MMRE = 15.0428
Pred(25) = 78.57143
Error range = [ -25.60657 .. 55.52895 ]
=====
```

```
Functionality vs. McCabe
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.07590812 0.1644303 -0.461643 0.644337342
x1          0.17946645 0.0607990  2.951799 0.003159282
R2log = 0.950971
Excluded as outliers: 23 26 11 38 ( 4 / 14 )
MMRE = 17.54617
Pred(25) = 85.71429
Error range = [ -19.89249 .. 66.49178 ]
=====
```

```
Functionality vs. Num. public methods
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) 2.191009e-01 1.243452e-01 1.762037 0.07806307
x1          2.045245e-05 9.340875e-06 2.189565 0.02855580
R2log = 0.9567552
Excluded as outliers: 23 6 11 ( 3 / 14 )
MMRE = 20.85635
Pred(25) = 64.28571
Error range = [ -9.03449 .. 81.73682 ]
=====
```

```
Functionality vs. ACC , LCOM
  Estimate Std. Error   z value Pr(>|z|)
(Intercept) -0.35883458 0.26157811 -1.371807 0.17012364
```

```

x1          0.11846781 0.04668115  2.537808 0.01115490
x2          0.06219426 0.02739112  2.270600 0.02317123
R2log = 0.9575212
Excluded as outliers: 23 11 33 ( 3 / 14 )
MMRE = 16.61442
Pred(25) = 78.57143
Error range = [ -7.442391 .. 57.07726 ]
=====
=====

Functionality vs. LCOM , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.64751779 0.35211448 -1.838941 0.065923846
x1          0.33420139 0.10892891  3.068069 0.002154471
x2          -0.04181575 0.01351686 -3.093599 0.001977443
R2log = 0.9574533
Excluded as outliers: 31 23 12 33 ( 4 / 14 )
MMRE = 18.42919
Pred(25) = 71.42857
Error range = [ -30.94845 .. 62.0385 ]
=====

Functionality vs. eLOC per method , eLOC per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.1503298185 0.1625709749 -0.9247027 0.355120578
x1          0.0331825698 0.0092090709  3.6032484 0.000314265
x2          -0.0008606183 0.0004175914 -2.0609101 0.039311617
R2log = 0.9565738
Excluded as outliers: 21 23 26 31 ( 4 / 14 )
MMRE = 17.92958
Pred(25) = 71.42857
Error range = [ -54.74645 .. 49.00091 ]
=====

Functionality vs. eLOC per method , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.01291205 0.141025978  0.09155795 0.927049260
x1          0.02616528 0.007695074  3.40026367 0.000673209
x2          -0.01654921 0.007586458 -2.18141529 0.029152714
R2log = 0.9544243
Excluded as outliers: 23 26 ( 2 / 14 )
MMRE = 14.61859
Pred(25) = 78.57143
Error range = [ -38.11871 .. 47.51121 ]
=====

Functionality vs. eLOC , Num. methods per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.861262e-01 3.913439e-01 -2.008786 0.044559821
x1          1.230460e-06 4.052053e-07  3.036633 0.002392363
x2          3.567668e-02 1.250492e-02  2.853011 0.004330710
R2log = 0.9583719
Excluded as outliers: 23 11 30 21 ( 4 / 14 )
MMRE = 19.85899
Pred(25) = 71.42857
Error range = [ -27.99154 .. 76.8662 ]
=====

Functionality vs. McCabe , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.08119489 0.15571578 -0.5214301 6.020672e-01
x1          0.42356272 0.10047029  4.2158008 2.488934e-05
x2          -0.04761877 0.01204921 -3.9520254 7.749250e-05
R2log = 0.9625412
Excluded as outliers: 23 12 27 ( 3 / 14 )
MMRE = 15.43948
Pred(25) = 78.57143
Error range = [ -56.20872 .. 47.23569 ]
=====
```

---

## 4.5 Interoperability

---

Interoperability vs. CBO , Num. methods per class

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.90223626	0.25261593	-3.571573	0.0003548435
x1	0.07831970	0.01940581	4.035889	0.0000543960
x2	0.02287279	0.01104113	2.071599	0.0383028347

R2log = 0.9531449  
 Excluded as outliers: 26 31 30 12 ( 4 / 14 )  
 MMRE = 13.13501  
 Pred(25) = 78.57143  
 Error range = [ -39.77479 .. 24.05713 ]

---

Interoperability vs. LCOM , eLOC

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-5.376699e-01	2.236527e-01	-2.404040	0.016215007
x1	7.824550e-02	2.632056e-02	2.972790	0.002951063
x2	8.366915e-07	2.570572e-07	3.254885	0.001134383

R2log = 0.9623516  
 Excluded as outliers: 23 11 30 6 ( 4 / 14 )  
 MMRE = 18.56473  
 Pred(25) = 71.42857  
 Error range = [ -29.90824 .. 80.73459 ]

---

Interoperability vs. LCOM , Num. public methods

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-3.823804e-01	1.972813e-01	-1.938250	0.052592785
x1	5.355102e-02	2.453868e-02	2.182311	0.029086614
x2	2.975123e-05	9.920541e-06	2.998952	0.002709101

R2log = 0.9610928  
 Excluded as outliers: 23 11 6 30 ( 4 / 14 )  
 MMRE = 20.38997  
 Pred(25) = 64.28571  
 Error range = [ -25.06132 .. 76.19507 ]

---

Interoperability vs. eLOC , Num. methods per class

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-7.919337e-01	4.051218e-01	-1.954804	0.050606206
x1	1.281041e-06	4.216898e-07	3.037876	0.002382521
x2	2.551186e-02	1.217675e-02	2.095129	0.036159496

R2log = 0.9587668  
 Excluded as outliers: 23 11 30 6 ( 4 / 14 )  
 MMRE = 22.17888  
 Pred(25) = 71.42857  
 Error range = [ -36.45152 .. 93.03068 ]

---

## 4.6 Security

---

Security vs. McCabe

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.2073593	0.17560733	-1.180813	0.237677203
x1	0.1563689	0.06016332	2.599074	0.009347573

R2log = 0.9576711  
 Excluded as outliers: 38 23 6 30 ( 4 / 14 )  
 MMRE = 18.18531  
 Pred(25) = 71.42857  
 Error range = [ -26.82613 .. 42.62553 ]

---

## 4.7 Performance (in terms of speed)

No valid model for speed was found.

## 4.8 Usefulness of the Product Developer Community

---



---

```
Community vs. ACC
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.58740552 0.12962802 -4.531470 5.857456e-06
x1          0.08234206 0.01777073  4.633578 3.593994e-06
R2log = 0.9547405
Excluded as outliers: 31 26 (2 / 14)
MMRE = 10.17461
Pred(25) = 92.85714
Error range = [-28.82385 .. 15.61735]
=====
=====

Community vs. CBO
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.5870887 0.12952371 -4.532674 5.824178e-06
x1          0.0823230 0.01775849  4.635700 3.557322e-06
R2log = 0.9547553
Excluded as outliers: 31 26 (2 / 14)
MMRE = 10.34233
Pred(25) = 92.85714
Error range = [-28.84201 .. 15.62564]
=====
=====

Community vs. LCOM
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.9732497 0.20630826 -4.717454 2.388147e-06
x1          0.1259511 0.02479839  5.079002 3.794229e-07
R2log = 0.954188
Excluded as outliers: 12 26 30 40 (4 / 14)
MMRE = 12.54779
Pred(25) = 92.85714
Error range = [-35.5852 .. 13.62179]
=====
=====

Community vs. eLOC per class
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.360054981 0.1489386742 -2.417471 0.015628763
x1          0.001328723 0.0004110311  3.232658 0.001226443
R2log = 0.9636467
Excluded as outliers: 21 38 11 6 (4 / 14)
MMRE = 22.64649
Pred(25) = 71.42857
Error range = [-9.006752 .. 113.6507]
=====
=====

Community vs. Num. methods per class
Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.12424641 0.142021759  0.8748406 0.38166062
x1          -0.01595695 0.008058194 -1.9802146 0.04767942
R2log = 0.961628
Excluded as outliers: 23 31 33 6 (4 / 14)
MMRE = 13.08653
Pred(25) = 71.42857
Error range = [-28.77573 .. 50.75046]
=====
=====

Community vs. LCOM , Num. attributes per class
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.81447634 0.163799366 -4.972402 6.612831e-07
x1          0.20005395 0.048581791  4.117879 3.823749e-05
x2          -0.02424760 0.009896842 -2.450034 1.428426e-02
```

```
R2log = 0.9612146
Excluded as outliers: 31 12 ( 2 / 14 )
MMRE = 10.11723
Pred(25) = 85.71429
Error range = [ -31.12256 .. 34.29227 ]
=====
=====
Community vs. LCOM , Num. classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.6738207342 2.196017e-01 -3.068377 0.002152251
x1          0.0814517360 2.671835e-02  3.048532 0.002299622
x2          0.0001101306 5.278197e-05  2.086520 0.036931562
R2log = 0.959998
Excluded as outliers: 23 6 ( 2 / 14 )
MMRE = 14.29262
Pred(25) = 85.71429
Error range = [ -10.84559 .. 90.38774 ]
=====
=====
Community vs. LCOM , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.347267046 0.133464151 -2.601950 0.0092695372
x1          0.103582841 0.027107852  3.821138 0.0001328373
x2          -0.003334854 0.001461017 -2.282557 0.0224564786
R2log = 0.9615928
Excluded as outliers: 31 6 ( 2 / 14 )
MMRE = 11.10023
Pred(25) = 92.85714
Error range = [ -11.21702 .. 47.82311 ]
=====
=====
Community vs. eLOC , McCabe
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.494036e-01 1.650298e-01 -3.329118 0.0008712148
x1          -5.096624e-07 2.009426e-07 -2.536358 0.0112012144
x2           2.659012e-01 6.910741e-02  3.847652 0.0001192554
R2log = 0.9524312
Excluded as outliers: 23 38 26 ( 3 / 14 )
MMRE = 18.89881
Pred(25) = 78.57143
Error range = [ -96.60586 .. 53.85015 ]
=====
```

## 4.9 Documentation Quality

```
DocQuality vs. LCOM
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.1761374 0.20640259 -5.698269 1.210303e-08
x1          0.1543135 0.02493453  6.188746 6.064469e-10
R2log = 0.951804
Excluded as outliers: 26 12 40 30 ( 4 / 14 )
MMRE = 24.56469
Pred(25) = 64.28571
Error range = [ -46.73282 .. 115.0743 ]
=====
=====
DocQuality vs. LCOM , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.060281719 0.144657312 -0.4167209 0.6768825977
x1          0.070364209 0.020671910  3.4038561 0.0006644173
x2          -0.003189213 0.001244662 -2.5623122 0.0103977803
R2log = 0.9479979
Excluded as outliers: 6 26 ( 2 / 14 )
MMRE = 31.39738
Pred(25) = 71.42857
Error range = [ -13.11330 .. 228.8443 ]
```

---

## 4.10 Trustworthiness with Respect to Non Open Source (Closed Source) Products

---

```
CssCompetitors vs. Num. attribute per method
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.1967843 0.2788195 0.7057767 0.480326992
x1          0.5778684 0.2051677 2.8165667 0.004853996
R2log = 0.9455916
Excluded as outliers: 38 23 31 ( 3 / 14 )
MMRE = 12.82807
Pred(25) = 92.85714
Error range = [ -13.75473 .. 55.43068 ]
```

---

```
CssCompetitors vs. ACC , LCOM
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.81241367 0.18207248 4.462034 8.118516e-06
x1          0.11761397 0.04014289 2.929883 3.390893e-03
x2          -0.09470962 0.04633366 -2.044078 4.094585e-02
R2log = 0.9458702
Excluded as outliers: 31 23 ( 2 / 14 )
MMRE = 11.81406
Pred(25) = 78.57143
Error range = [ -31.08982 .. 42.3786 ]
```

---

```
CssCompetitors vs. ACC , Num. attributes per class
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.76021858 0.146926845 5.174130 2.289755e-07
x1          0.09766702 0.034094393 2.864607 4.175276e-03
x2          -0.02054296 0.007889923 -2.603695 9.222465e-03
R2log = 0.9473084
Excluded as outliers: 23 ( 1 / 14 )
MMRE = 10.14672
Pred(25) = 85.71429
Error range = [ -13.11653 .. 45.0169 ]
```

---

```
CssCompetitors vs. ACC , Num. methods per class
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.90912915 0.201063381 4.521605 6.137255e-06
x1          0.07146669 0.030381788 2.352287 1.865836e-02
x2          -0.01990298 0.008380173 -2.375008 1.754856e-02
R2log = 0.9469983
Excluded as outliers: 23 31 ( 2 / 14 )
MMRE = 10.79340
Pred(25) = 85.71429
Error range = [ -19.74199 .. 50.63433 ]
```

---

```
CssCompetitors vs. ACC , RFC
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.810252111 0.152721869 5.305410 1.124201e-07
x1          0.068533460 0.029901800 2.291951 2.190847e-02
x2          -0.002757649 0.001226429 -2.248520 2.454308e-02
R2log = 0.9460376
Excluded as outliers: 23 ( 1 / 14 )
MMRE = 11.02680
Pred(25) = 85.71429
Error range = [ -9.854596 .. 51.67716 ]
```

---

```
CssCompetitors vs. CBO , Num. attributes per class
  Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) 0.78021113 0.142022476 5.493575 3.938776e-08
x1          0.09793209 0.034113617 2.870763 4.094830e-03
x2          -0.02126357 0.008028686 -2.648450 8.086189e-03
R2log = 0.9473551
Excluded as outliers: 23 (1 / 14)
MMRE = 10.22969
Pred(25) = 85.71429
Error range = [-13.22009 .. 45.6625]
=====
=====
CssCompetitors vs. CBO , Num. methods per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.93804751 0.193806038 4.840136 1.297506e-06
x1          0.06957322 0.029764967 2.337420 1.941737e-02
x2          -0.02054070 0.008424547 -2.438196 1.476077e-02
R2log = 0.9469472
Excluded as outliers: 23 31 (2 / 14)
MMRE = 11.01684
Pred(25) = 85.71429
Error range = [-19.63953 .. 51.79163]
=====
=====
CssCompetitors vs. CBO , Num. public methods
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.254925e-01 3.441033e-01 -0.6553046 0.512271635
x1          2.056646e-01 6.258897e-02 3.2859561 0.001016368
x2          2.667686e-05 1.132542e-05 2.3554859 0.018498500
R2log = 0.947548
Excluded as outliers: 23 11 33 31 (4 / 14)
MMRE = 10.88707
Pred(25) = 85.71429
Error range = [-20.11094 .. 37.30294]
=====
=====
CssCompetitors vs. CBO , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.826363527 0.147495173 5.602648 2.111017e-08
x1          0.067873751 0.029615757 2.291812 2.191649e-02
x2          -0.002831636 0.001236111 -2.290763 2.197714e-02
R2log = 0.9460443
Excluded as outliers: 23 (1 / 14)
MMRE = 11.09106
Pred(25) = 85.71429
Error range = [-9.608151 .. 52.41336]
=====
=====
CssCompetitors vs. LCOM , Num. classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.1206910277 3.243335e-01 -0.3721201 0.709803440
x1          0.1054928918 3.750434e-02 2.8128184 0.004910938
x2          0.0002092773 6.682759e-05 3.1315997 0.001738568
R2log = 0.9507166
Excluded as outliers: 23 6 40 30 (4 / 14)
MMRE = 12.01640
Pred(25) = 78.57143
Error range = [-20.59733 .. 31.36449]
=====
=====
CssCompetitors vs. eLOC per method , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.80183742 0.145301133 5.518453 3.419977e-08
x1          0.03260312 0.009422386 3.460177 5.398205e-04
x2          -0.04501317 0.012269553 -3.668688 2.437979e-04
R2log = 0.9509458
Excluded as outliers: 23 11 (2 / 14)
MMRE = 12.21157
Pred(25) = 78.57143
```

```
Error range = [ -51.63936 .. 37.34865 ]
=====
=====

CssCompetitors vs. eLOC per method , Num. methods per class
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.95394789 0.190135942 5.017189 5.243308e-07
x1          0.01605134 0.006908994 2.323252 2.016560e-02
x2         -0.02551988 0.008850982 -2.883283 3.935538e-03
R2log = 0.945248
Excluded as outliers: 23 31 40 ( 3 / 14 )
MMRE = 12.43937
Pred(25) = 85.71429
Error range = [ -25.64484 .. 51.77958 ]
=====

=====

CssCompetitors vs. eLOC per method , RFC
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.80377663 0.146377383 5.491126 3.993797e-08
x1          0.01996342 0.007856225 2.541095 1.105058e-02
x2         -0.00399077 0.001436051 -2.778989 5.452843e-03
R2log = 0.946928
Excluded as outliers: 23 ( 1 / 14 )
MMRE = 11.40184
Pred(25) = 85.71429
Error range = [ -21.8786 .. 52.13241 ]
=====

=====

CssCompetitors vs. McCabe , Num. attributes per class
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.65900945 0.17663946 3.730817 0.0001908600
x1          0.37306403 0.11494886 3.245478 0.0011725351
x2         -0.05943443 0.01651706 -3.598366 0.0003202233
R2log = 0.9491308
Excluded as outliers: 23 11 12 ( 3 / 14 )
MMRE = 12.39782
Pred(25) = 85.71429
Error range = [ -58.62359 .. 34.73436 ]
=====

=====

CssCompetitors vs. McCabe , Num. methods per class
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.92696071 0.204470613 4.533467 5.802342e-06
x1          0.13232514 0.062069741 2.131878 3.301684e-02
x2         -0.02696131 0.009139742 -2.949898 3.178785e-03
R2log = 0.9444082
Excluded as outliers: 23 31 40 ( 3 / 14 )
MMRE = 12.73944
Pred(25) = 85.71429
Error range = [ -25.52103 .. 53.4052 ]
=====

=====

CssCompetitors vs. Num. attributes per class , Num. attribute per
method
  Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.36109339 0.24855162 1.452790 1.462820e-01
x1          -0.05406766 0.01335819 -4.047531 5.176084e-05
x2          0.82088517 0.22949849  3.576865 3.477394e-04
R2log = 0.9538842
Excluded as outliers: 23 11 33 31 ( 4 / 14 )
MMRE = 13.93870
Pred(25) = 78.57143
Error range = [ -53.65518 .. 36.72223 ]
=====
```

## 4.11 Trustworthiness with Respect to Open Source Products

```
OssCompetitors vs. eLOC per method
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.23408088 0.17720754 1.320942 0.186520706
x1          0.02176670 0.00753464 2.888885 0.003866108
R2log = 0.9386403
Excluded as outliers: 38 23 26 40 ( 4 / 14 )
MMRE = 13.03507
Pred(25) = 85.71429
Error range = [ -18.60721 .. 55.40829 ]
=====
=====
OssCompetitors vs. McCabe
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.3173418 0.18393871 1.725258 0.08448094
x1          0.1395156 0.06310081 2.210996 0.02703614
R2log = 0.9462933
Excluded as outliers: 38 23 40 ( 3 / 14 )
MMRE = 14.45781
Pred(25) = 85.71429
Error range = [ -17.11461 .. 65.59923 ]
=====
=====
OssCompetitors vs. Num. attribute per method
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.09565536 0.2669680 -0.3583028 0.720116745
x1          0.58951143 0.1957686 3.0112673 0.002601597
R2log = 0.94848
Excluded as outliers: 38 23 31 40 ( 4 / 14 )
MMRE = 15.30319
Pred(25) = 85.71429
Error range = [ -15.24604 .. 77.54009 ]
=====
=====
OssCompetitors vs. ACC , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.193124e-01 2.081705e-01 -1.533899 0.125054607
x1          9.512686e-02 3.237944e-02 2.937878 0.003304666
x2          8.237929e-07 2.591264e-07 3.179116 0.001477248
R2log = 0.95981
Excluded as outliers: 23 11 31 30 ( 4 / 14 )
MMRE = 10.40612
Pred(25) = 85.71429
Error range = [ -25.19403 .. 30.04013 ]
=====
=====
OssCompetitors vs. ACC , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.40655808 0.141819950 2.866720 0.0041474982
x1          0.11792319 0.033468869 3.523369 0.0004260971
x2          -0.02172221 0.008016072 -2.709832 0.0067317293
R2log = 0.9510754
Excluded as outliers: 23 40 ( 2 / 14 )
MMRE = 9.966128
Pred(25) = 92.85714
Error range = [ -9.989877 .. 42.72601 ]
=====
=====
OssCompetitors vs. CBO , eLOC per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.4122575172 0.1599775821 2.576971 0.0099670443
x1          0.1210676594 0.0366927598 3.299497 0.0009685818
x2          -0.0009670465 0.0004085868 -2.366808 0.0179422378
R2log = 0.9512633
Excluded as outliers: 23 40 21 31 ( 4 / 14 )
MMRE = 14.42386
Pred(25) = 85.71429
Error range = [ -46.46329 .. 61.37502 ]
```

```
=====
=====
OssCompetitors vs. CBO , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.248452e-01 2.096827e-01 -1.549223 0.1213281834
x1          9.346898e-02 3.184942e-02  2.934716 0.0033385348
x2          8.819743e-07 2.604953e-07  3.385759 0.0007098181
R2log = 0.9598142
Excluded as outliers: 23 11 31 30 ( 4 / 14 )
MMRE = 10.44114
Pred(25) = 85.71429
Error range = [ -25.15033 .. 31.33668 ]
=====
```

```
=====
=====
OssCompetitors vs. CBO , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.43318339 0.137728419 3.145200 1.659735e-03
x1          0.14770516 0.036059105 4.096196 4.199944e-05
x2          -0.02946704 0.008659143 -3.402997 6.665095e-04
R2log = 0.9566234
Excluded as outliers: 23 40 27 ( 3 / 14 )
MMRE = 10.08933
Pred(25) = 78.57143
Error range = [ -7.82335 .. 36.42596 ]
=====
```

```
=====
=====
OssCompetitors vs. LCOM , eLOC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.447879e-01 2.377248e-01 -1.871021 0.0613422130
x1          9.701718e-02 2.750497e-02  3.527260 0.0004198839
x2          1.183040e-06 3.453409e-07  3.425716 0.0006131823
R2log = 0.9529247
Excluded as outliers: 23 11 30 26 ( 4 / 14 )
MMRE = 11.17179
Pred(25) = 78.57143
Error range = [ -25.99168 .. 35.10325 ]
=====
```

```
=====
=====
OssCompetitors vs. LCOM , Num. classes
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.3511150900 2.355584e-01 -1.490565 0.1360757260
x1          0.1065373057 2.950291e-02  3.611078 0.0003049269
x2          0.0001958606 5.563282e-05  3.520595 0.0004305802
R2log = 0.95833
Excluded as outliers: 6 23 30 ( 3 / 14 )
MMRE = 11.79174
Pred(25) = 85.71429
Error range = [ -21.57931 .. 44.4996 ]
=====
```

```
=====
=====
OssCompetitors vs. LCOM , RFC
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.145063140 0.160047281 -0.9063768 3.647365e-01
x1          0.187438475 0.031621691  5.9275285 3.075280e-09
x2          -0.003993762 0.001455453 -2.7439995 6.069563e-03
R2log = 0.9586327
Excluded as outliers: 31 12 30 ( 3 / 14 )
MMRE = 11.48521
Pred(25) = 92.85714
Error range = [ -20.73418 .. 74.66464 ]
=====
```

```
=====
=====
OssCompetitors vs. eLOC per method , Num. attributes per class
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.59804881 0.16652342  3.591380 0.0003289321
x1          0.02357554 0.01057863  2.228600 0.0258405490
x2          -0.04001477 0.01205332 -3.319812 0.0009007799
```

```
R2log = 0.9524063
Excluded as outliers: 23 11 33 40 ( 4 / 14 )
MMRE = 15.28894
Pred(25) = 78.57143
Error range = [ -53.70027 .. 38.27468 ]
=====
```

## 4.12 Trustworthiness

---

```
Trustworthiness vs. ACC
Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.04322005 0.15431496 0.2800769 7.794185e-01
x1 0.11088778 0.02812860 3.9421725 8.074688e-05
R2log = 0.9534642
Excluded as outliers: 23 31 26 27 ( 4 / 14 )
MMRE = 13.12188
Pred(25) = 92.85714
Error range = [ -24.00001 .. 70.41757 ]
=====
```

```
Trustworthiness vs. CBO
Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.04383213 0.15413018 0.2843838 7.761163e-01
x1 0.11081394 0.02809886 3.9437164 8.022853e-05
R2log = 0.9534742
Excluded as outliers: 23 26 31 27 ( 4 / 14 )
MMRE = 13.27053
Pred(25) = 92.85714
Error range = [ -24.01627 .. 70.28749 ]
=====
```

```
Trustworthiness vs. LCOM , Num. classes
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.8795987366 3.018419e-01 -2.914104 3.567112e-03
x1 0.1795246487 3.523546e-02 5.094999 3.487435e-07
x2 0.0003059096 6.165806e-05 4.961389 6.999080e-07
R2log = 0.9565812
Excluded as outliers: 6 23 40 30 ( 4 / 14 )
MMRE = 14.07461
Pred(25) = 92.85714
Error range = [ -19.56384 .. 76.62689 ]
=====
```

## 5 CONCLUSIONS

The activities reported above were largely successful, in the sense that they identified the existence of several statistically valid models of the subjective qualities as functions of the internal, objectively measurable qualities.

For both Java and C++ products, we were able to find several statistically significant correlations. In particular, we were able to find at least one correlation for every subjective quality evaluated by means of questionnaires. In other words, we have a quantitative model for every quality.

### 5.1 Usage of the results

The main result of the activity reported in this document does not consist just in having found that relationships exist between trustworthiness (and other perceived qualities) and objectively measurable characteristics of the OSS. A really important point is that we were able to *quantify* the nature of these relationships.

The quantitative knowledge of the relationships can be beneficial to both the users and the developers of OSS:

- The users can rely on the measures of the software in order to estimate to what extent a given OSS product can be expected to satisfy a given quality aspect (e.g., reliability). In this way, the potential users can get a rough evaluation of OSS without the need to even try the product.
- Developers can derive from their client satisfaction targets (i.e. to what extent users will be satisfied with a given quality of their OSS product) into threshold of quality metrics that must be met by their code.

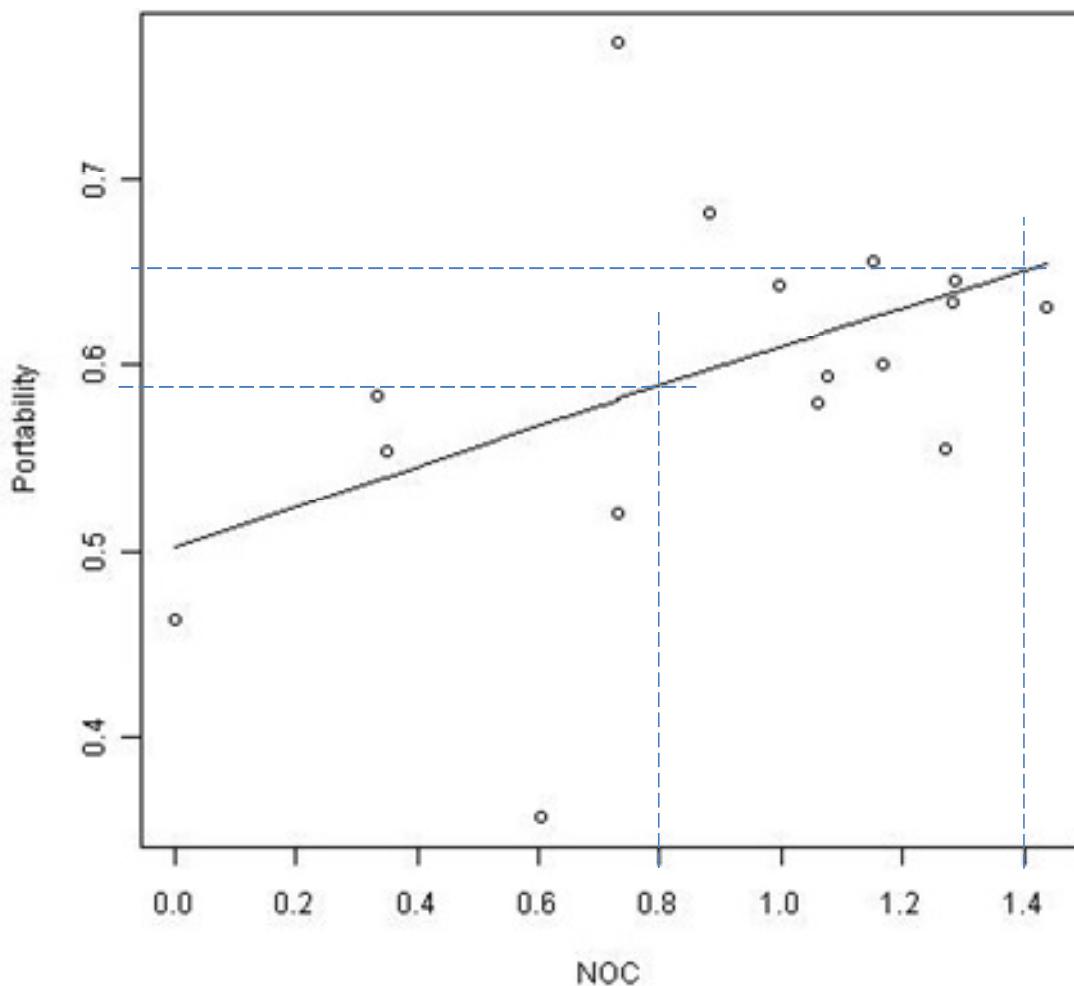
The procedure for using the quantitative knowledge of relations is exemplified below, considering the dependency of Portability on the average number of children of classes in Java products. The equation that describes the correlation found is the following:

$$\text{Portability} = \frac{1}{1 + e^{-(0.0061183 + 0.44062 \text{ NOC})}}$$

For users the procedure is simple: given a Java product, if its average number of children per class is 0.8 then a user can expect that the product's portability will be satisfactory with probability around 60%, i.e., 60% of the user will be satisfied with the product's portability. In fact,  $\frac{1}{1 + e^{-(0.0061183 + 0.44062 \cdot 0.8)}} = 0.5887$ .

Instead, if the average number of children per class is 1.4, then the user can expect that the product's portability will be satisfactory with probability around 65%. In fact,  $\frac{1}{1 + e^{-(0.0061183 + 0.44062 \cdot 1.4)}} = 0.651$ .

**Figure 7** shows the relation between portability and NOC, highlighting the values of portability for NOS = 0.8 and NOC = 1.4.



**Figure 7. Portability vs. NOC (number of children) for Java products.**

For developers the procedure is reversed, since they have to establish what is the minimum value of NOC to achieve a target portability probability. By solving with respect to NOC the equation that describes the relation between portability and NOC we get:

$$NOC = 0.44062^{-1} \left( -0.0061183 + \log \left( \frac{\text{Portability}_{\text{target}}}{1 - \text{Portability}_{\text{target}}} \right) \right)$$

If the goal of the developer is that the product is considered trustworthy by over 55% of the users, he/she must aim at an average NOC greater than 0.44. In fact,

$$0.44062^{-1} \left( -0.0061183 + \log \left( \frac{0.55}{1-0.55} \right) \right) = 0.44.$$

In this sense, NOC=0.44 can be seen as a “threshold” below which developers should not take their code.

Other thresholds may be established too, by users and developers alike. For instance, one may decide that a product’s quality is

- “good” if the product’s portability is expected to be rated satisfactory by at least 75% of users

- “acceptable” if the product’s portability is expected to be rated satisfactory by 25% to 75% of users
- “poor” if the product’s portability is expected to be rated satisfactory by at most 25% of users.

By solving the logistic regression formula for the value of the objectively measurable characteristic and substituting 0.75 and 0.25, one obtains the threshold values for the objectively measurable characteristic that correspond to the 75% and 25% values of satisfied users, so one finds the thresholds on the range of the objectively measurable characteristic. For instance, these thresholds can be displayed by Spago4Q, which is used for result visualization in our tool set.

In conclusion, our analyses let users and developers perform the needed evaluations on the basis of clear quantitative data.

## 5.2 Improvements over the state of the art

We also would like to point out that we used the theoretically sound way to build models belonging to MOSST, which are relevant to developers and users (e.g., reliability, usability, portability, etc. as listed in Section 0). All of these characteristics are classified as “external” software qualities, as opposed to “internal” software quality in the experimental software engineering literature [31]:

- an “internal” quality of a software product can be quantified based on the knowledge of the software product alone. For instance, one can measure software size based only on the source code. An “internal” software quality has no practical interest *per se*, but it can be used to predict some interesting product (e.g., fault proneness) or process (e.g., effort) quality. “Internal” software qualities are usually easy to measure, possibly with the aid of automated tools.
- an “external” quality of a software product can only be quantified based on the knowledge of the software product and additional information. For instance, one cannot measure software usability based only on the source code or its user interface, as usability also depends on the skills and knowledge of specific users and the way they interact with the product. “External” software qualities are practically relevant, though it is commonly said that they are usually very hard to measure (and even define).

However, the distinction between “internal” and “external” qualities is entirely peculiar to the software measurement literature. No such distinction exists in the authoritative general references on measurement [32][33][34]. So, this distinction between “internal” and “external” qualities in software engineering literature can be only accepted as an illustration convention used to explain the differences in the nature of software qualities and somehow organize them in categories.

At any rate, the authoritative, foundational work on measurement already shows how these “external” qualities can be measured. Technically, this is carried out

via so-called “Probability Representations.” From a practical point of view, this amounts to measuring “external” qualities via prediction models based on “internal” qualities, as we have shown elsewhere [35] and done in our work here, as MOSST is the ensemble of a set of prediction models.

Thus, our approach is different from other approaches that have been used in the quantification of qualities of OSS, like OpenBRR [37], QSOS [36], OSMM [38], OpenBQR [39], which are typically based on weighted sums of directly measurable characteristics. However, these models are not theoretically valid, nor are they validatable, as they provide a *definition* of a quality. From a practical point of view, these models do not provide any reliable indication on whether the directly measurable characteristics they use actually influence the qualities of interest, nor on the values for the weights of the directly measurable characteristics. So, the choice of directly measurable characteristics and the values used for their weights are fairly subjective, and, as a result, so is the definition of the quality. Instead, the solid statistical analysis used in MOSST shows which directly measurable characteristics are truly influential on the OSS qualities of interest and which weights should be used, based on an extensive set of data coming from the field and not on some fairly subjective analysis.

So, unlike existing quality models for OSS, MOSST is built by means of a theoretically valid approach and solid statistical techniques that use evidence coming from OSS stakeholders and the analysis of actual OSS products and projects. We collected data from 694 OSS stakeholders and obtained 4101 evaluations on 22 Java and 22 C++ programs. This has allowed us to build statistically valid models to quantitatively predict the impact of objectively observable characteristics of OSS products and projects on qualities of practical interest, instead of collecting only data on objectively observable qualities of the code that may only be conjectured to influence qualities of practical interest.

## 6 REFERENCES

- [1] V. Basili and D. Weiss, "A methodology for collecting valid software engineering data," IEEE Transactions on Software Engineering, vol. SE-10, no. 6, pp. 728-738, 1984.
- [2] V. Basili and D. Rombach, "The TAME project: towards improvement-oriented software environments," IEEE Transactions on Software Engineering, vol. 14, no. 6, pp. 758-773, 1988.
- [3] V. Basili, G. Caldiera, and D. Rombach, "Goal/Question/Metric Paradigm," in Encyclopedia of Software Engineering, vol. 1, J. C. Marciniak, Ed.: John Wiley & Sons, 1994, pp. 528-532.
- [4] R. van Solingen and E. Berghout, The Goal/Question/Metric Method, McGraw-Hill, 1999. <http://www.gqm.nl/>
- [5] V.R. Basili, G. Caldiera, H.D. Rombach, "The Goal Question Metric Approach", <http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/gqm.pdf>
- [6] V. del Bianco, M. Chinosi, L. Lavazza, S. Morasca, D. Taibi, "How European software industry perceives OSS trustworthiness and what are the specific criteria to establish trust in OSS", QualiPSO report, October 2007, <http://qualipso.semanticdesktop.org/xwiki/bin/view/Wiki/WP51>.
- [7] V. del Bianco, L. Lavazza, S. Morasca, D. Taibi, "Analysis of relevant open-source projects and artifacts", Working document wd 5.2.1, Version 0.1 - QualiPSO report, February 2008.
- [8] V. del Bianco, L. Lavazza, S. Morasca, D. Taibi, "Definition of trustworthiness of software products and artefacts, Version 2.0", QualiPSO Working document wd 5.3.1, 31/10/2008.
- [9] V. del Bianco, L. Lavazza, S. Morasca, D. Taibi, "Identification of factors that influence the trustworthiness of software products and artefacts, Version 2.0", QualiPSO Working document wd 5.3.2, 31/10/2008.
- [10] L. Lavazza, S. Morasca, D. Taibi, D. Tosi, "Definition and identification of trustworthiness of software products and artefacts and its influencing factors", Working document wd 5.3.1, Version 4.0 - QualiPSO report, January 2011.
- [11] ISO/IEC 9126-1:2001 "Software Engineering—Product Quality—Part 1: Quality model", June 2001.
- [12] <http://www.r-project.org/>
- [13] <http://www.cs.waikato.ac.nz/~ml/weka/>
- [14] K. Hornik, C. Buchta and A. Zeileis, "Open-source machine learning: R meets Weka", Computational Statistics, Physica Verlag, An Imprint of Springer-Verlag GmbH, May 14, 2008.

- [15] L. Lavazza, S. Morasca, D. Taibi, D. Tosi, "Experimentation on the trustworthiness of Open Source Software, version 2.0", QualiPSO Working document wd 5.6.1, 30/06/2009.
- [16] L. Lavazza, S. Morasca, D. Taibi, D. Tosi, "Experimentation on the trustworthiness of Open Source Software, version 3.0", QualiPSO Working document wd 5.6.1, 31/01/2011.
- [17] The R Development Core Team, "R: A Language and Environment for Statistical Computing - Reference Index, Version 2.9.0 (2009-04-17)", R Foundation for Statistical Computing, 2009.
- [18] V. del Bianco, L. Lavazza, S. Morasca, D. Taibi, "Trustworthiness models for Open Source Software, version 1.0", QualiPSO Working document wd 5.6.2, 31/10/2008.
- [19] Lionel C. Briand, Sandro Morasca, and Victor R. Basili, "Defining and Validating Measures for Object-Based High-Level Design", IEEE Transactions on Software Engineering, Vol. 25, No. 5, September/October 1999
- [20] del Bianco, V., Chinosi, M., Lavazza, L., Morasca, S., and Taibi, D. 2008. How European software industry perceives OSS trustworthiness and what are the specific criteria to establish trust in OSS, QualiPSO report, October 2008.
- [21] del Bianco, V., Lavazza, L., Morasca, S., Taibi, D., and Tosi, D. 2010. An investigation of the users' perception of OSS quality. The 6th International Conference on Open Source Systems, OSS 2010, 30 May - 2 June 2010, Notre Dame, IN, USA.
- [22] QualiPSO project portal. <http://www.qualipso.org>.
- [23] PMD web site: <http://pmd.sourceforge.net/>
- [24] del Bianco, V., Lavazza, L., Morasca, S., Taibi, D., and Tosi, D. 2010. The QualiSPo approach to OSS product quality evaluation. Workshop on Emerging Trends in FLOSS Research and Development (FLOSS-3), Cape Town, South Africa, May 8, 2010.
- [25] Crisà, A.F., del Bianco, V., Lavazza, L. 2006. A tool for the measurement, storage, and pre-elaboration of data supporting the release of public datasets, Workshop on Public Data about Software Development (WoPDaSD 2006), Como, June 10, 2006.
- [26] PC-lint web site: <http://www.gimpel.com/>
- [27] SISSy web site: <http://sissy.fzi.de>
- [28] Findbugs web site: <http://findbugs.sourceforge.net/>
- [29] Gruber, H., Hentschel, A., Plösch, R. 2009. Experiment for Comparing the Automatically Assessed Source Code Quality with Experts' Opinions. SQMB 2009 Workshop, Kaiserslautern, March 3<sup>rd</sup> 2009, T.U. Munich Technical Report TUM-I0917.

- [30] Ayewah, N., Pugh, W., Morgenthaler, J. D., Penix, J., and Zhou, Y. 2007. Evaluating static analysis defect warnings on production software. 7<sup>th</sup> ACM SIGPLAN-SIGSOFT Workshop on Program Analysis For Software Tools and Engineering (San Diego, California, USA, June 13 - 14, 2007).
- [31] Norman Fenton and Shari Lawrence Pfleeger. Software Metrics (2nd Ed.): A Rigorous and Practical Approach. PWS Pub. Co., Boston, MA, USA. 1997.
- [32] Roberts F.S., Measurement Theory: With Applications to Decision Making, Utility, and the Social Sciences, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1979.
- [33] Tversky A., Luce R.D., Suppes P., Krantz D., Foundations of Measurement vol. I: Additive and Polynomial Representations, Academic Press, 1971.
- [34] Tversky A., Luce R.D., Suppes P., Krantz D., Foundations of Measurement vol. II: Geometrical, Threshold, and Probabilistic Representations, Academic Press, 1989.
- [35] Sandro Morasca. A probability-based approach for measuring external attributes of software artifacts. In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM '09). IEEE Computer Society, Washington, DC, USA, 2009.
- [36] Atos Origin, "Method for Qualification and Selection of Open Source software (QSOS), version 1.6", <http://www.qsos.org/download/qzos-1.6-en.pdf>
- [37] "Business Readiness Rating for Open Source - A Proposed Open Standard to Facilitate Assessment and Adoption of Open Source Software", BRR 2005 - RFC 1, <http://www.openbrr.org>.
- [38] "Making Open Source Ready for the Enterprise: The Open Source Maturity Model", from "Succeeding with Open Source" by Bernard Golden, Addison-Wesley, 2005, available form <http://www.navicasoft.com>
- [39] D. Taibi, L. Lavazza, S. Morasca, "OpenBQR: a framework for the assessment of OSS", Open Source Software 2007, Limerick, June 2007.

## 7 APPENDIXES

### 7.1 The logistic function

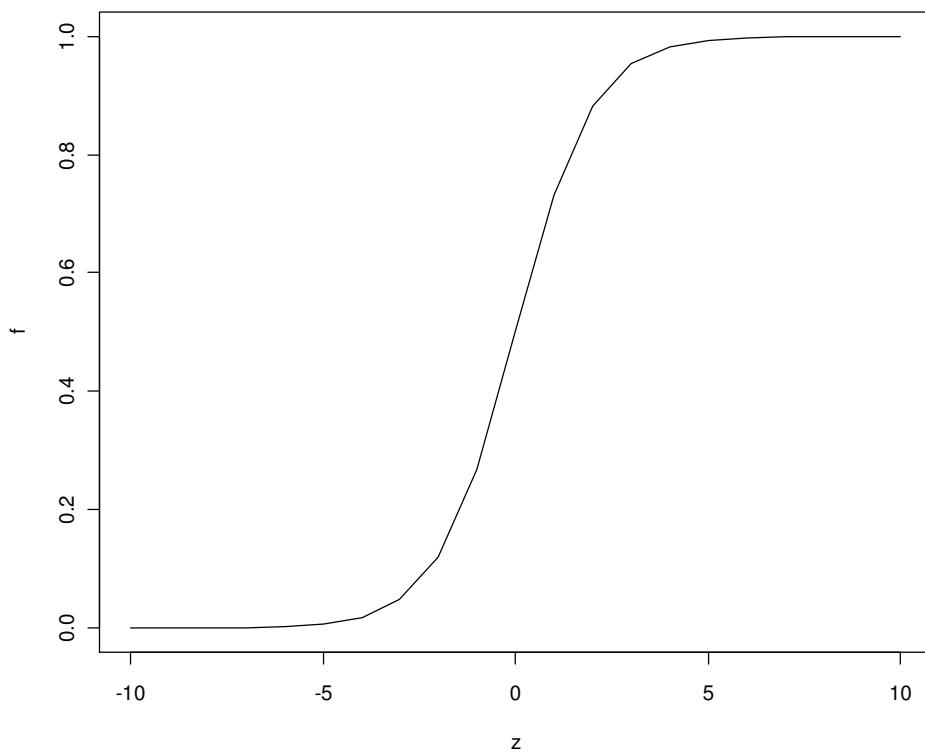
The logistic function is defined as follows:

$$f(z) = \frac{1}{1 + e^{-z}}$$

The variable  $z$  is usually defined as  $z = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k$ .

A graph of the function is shown in Figure 8. Note that when  $z$  is inversely proportional to  $X$ , the function starts at one and decreases towards zero.

$f(z)$  is used to estimate a dichotomous dependent variable  $Y$  (i.e., a response variable that can only take two values  $a$  and  $b$ ).  $f(z)$  is the probability that  $Y = a$  (therefore  $1-f(z)$  is the probability that  $Y = b$ ). In our case, we want to estimate the probability that a user was or was not satisfied with the trustworthiness of an OSS product.



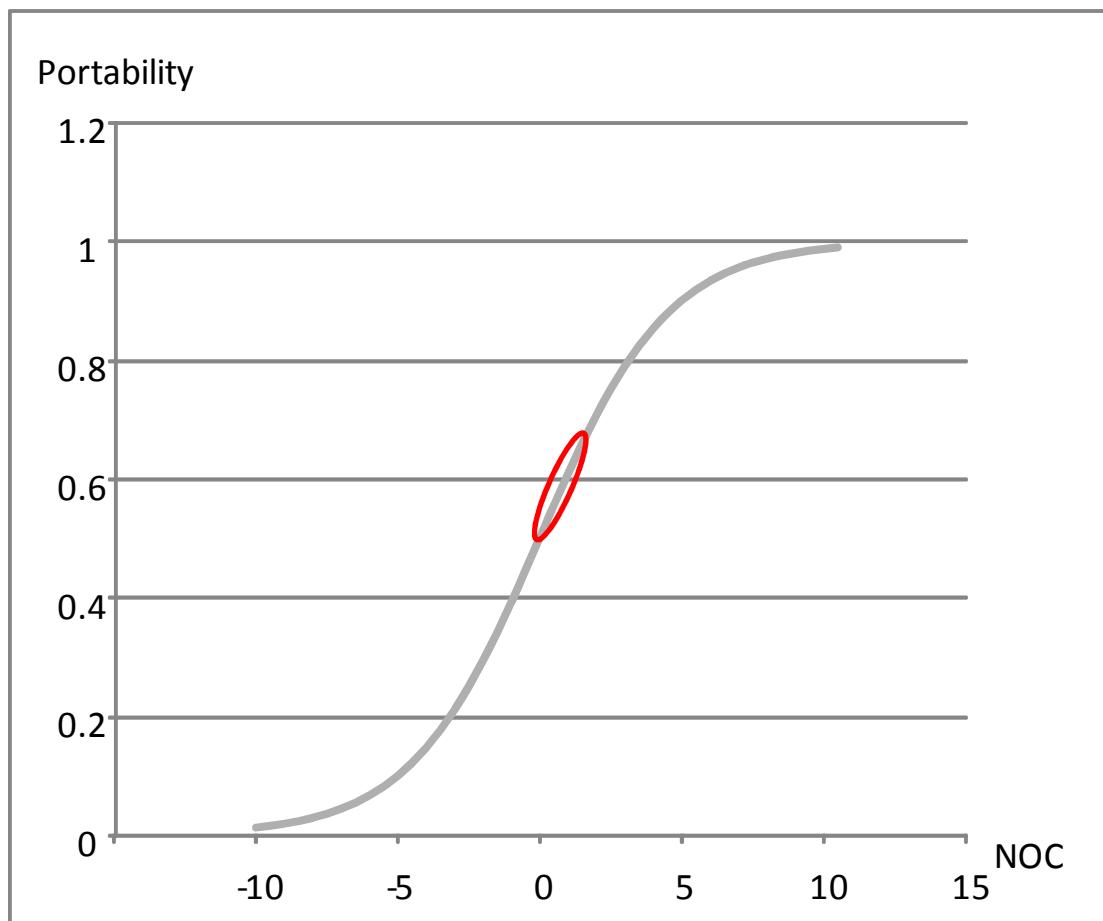
**Figure 8. The logistic function.**

Logistic regression has many analogies to linear regression. Unlike the latter, however, logistic regression does not assume linearity of relationship between the independent variables and the dependent, does not require normally distributed variables, does not assume homoscedasticity, and in general has less stringent requirements. It does, however, require that observations be independent and that the independent variables be linearly related to the logit of the dependent variable ( $\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = -z$ ).

The logistic curve, illustrated in Figure 8, is better for modelling binary dependent variables coded 0 or 1 because it comes closer to hugging the  $y=0$  and  $y=1$  points on the y axis. Even more, the logistic function is bounded by 0 and 1, whereas the linear regression function may predict values above 1 and below 0.

You may have noticed that the correlations reported in the paper do not look S-shaped like Figure 8. The reason is that our correlations are defined for a relatively small interval of values of the independent variables.

Consider for instance the function  $\text{Portability} = \frac{1}{1+e^{-(0.0061185+0.44062 \cdot \text{NOC})}}$  which is reported in **Figure 7**, limitedly to the 0..1.4 range. The same function, plotted for whole x axis is reported in **Figure 9**. The highlighted region corresponds to the portion of the function illustrated in **Figure 7**. It is easy to see that in such region the function is approximately linear.



**Figure 9. The logistic function**  $\text{Portability} = \frac{1}{1+e^{-(0.0061185+0.44062 \cdot \text{NOC})}}$

## 7.2 Elementary Code Assessment Rules

We checked the ability of the chosen OSS products to satisfy a set of rules that are measurable via code static analysis and purport to indicate the probability of run-time errors. We checked whether the number of rule violations is actually

statistically correlated with the quality of the OSS products, as reported by users.

Elementary Code Assessment (ECA) is a pragmatic approach for providing a quick estimate of the internal quality of a software product with little effort. ECA provides software quality evaluations with less time and effort than other techniques. However, ECA is simpler than most existing techniques, so it may also provide less information. This simplification is highlighted by the term “elementary” in the name of the approach. Also, ECA takes into account only the “internal” quality of a software product and thus excludes “external” qualities such as usability and performance.

The ECA approach, originally developed for internal use within Siemens AG [28], was then refined and broadened to support the needs of the evaluation of OSS products. A small set of rules and measures, which can be evaluated by means of software tools, was defined for C/C++ and Java. The rules were defined by a team of experts who have many years of experience in the areas of software review, static analysis tool application, design and implementation, as well as in the creation of coding guidelines in various application areas. The team of experts focused on a few rules whose violation had been often observed to be connected with serious errors. Clearly, the ECA approach may not detect all problems, but it may help find significant defects [30]. Despite being based on static analysis, ECA rules address the run-time behavior of the software and the likelihood of run-time errors. The set of measures obtained can be used to estimate the internal quality of OSS products in many domains, without requiring special customization and tuning of the set.

The set of ECA measures can be automatically quantified through simple configurations of well-known static analysis tools, such as PC-lint [26], SISSy [27], Findbugs [28] and PMD [23]. In the work reported here, we selected the subset of rules evaluated by PMD, and we focused on OSS Java products. PMD is an OSS tool that scans Java source code and looks for potential causes of correctness, readability, modifiability, and efficiency problems in the program [23], such as empty try statements, unused local variables, unused parameters, unused private methods, wasteful string or buffer usage, duplicate code. In the context of QualiPSO, PMD has been integrated into our static analysis tool MacXim [25][24], which we used to analyze the OSS Java products and compute the subset of ECA measures.

The ECA rules are the following (in PMD terminology):

1. Avoid Catching Throwable
2. Constructor Calls Overridable Method
3. Class Naming Conventions
4. Empty Catch Block
5. Excessive Class Length
6. Excessive Method Length
7. For Loops Must Use Braces
8. If Else Statements Must Use Braces
9. If Statements Must Use Braces
10. Missing Break In Switch

11. Override Both Equals And Hashcode
12. Unused Private Field
13. Unused Private Method
14. Switch Statements Should Have Default
15. Use Equals To Compare Strings
16. While Loops Must Use Braces

### 7.3 Measure names and definitions

A few metrics are named differently in different contexts. The following table provides the mapping of metrics names and definitions.

MOSST	KALIBRO	definition	Type
RFC	rfc	Response For a Class	per Class
ACC	acc	Afferent Connections per Class	per Class
CBO	cbo	Coupling Between Objects	per Class
LCOM	lcom4	Lack of Cohesion of Methods	per Class
eLOC	total_loc	Total Lines of Code	sum/total
eLOC per Class	loc	Lines of Code per Class	per Class
eLOC per Method	amloc	Average Method LOC	per Class
McCabe	accm	Average Cyclomatic Complexity per Method	per Class
Num. Classes	total_modules	total_modules - Total Number of Modules	sum/total
Num. Attributes per Class	noa	Number of Attributes	per Class
Num. Methods per Class	nom	Number of Methods	per Class
NOC	noc	Number of Children	per Class
Num. Methods	total_nom	total_nom - Total Number of Methods	sum/total
Num. Attributes per Method	anpm	Average Number of Parameters per Method	per class