

Paper: Reverse Engineering and Software Products Reuse to Teach Collaborative Web Portals: A Case Study with Final-Year Computer Science Students

Authors: Fuensanta Medina-Dominguez, Maribel Sanchez-Segura, Arturo Mora-Soto and Antonio Amescua

FEEDBACK OBTAINED AFTER APPLYING THE TEACHING STRATEGY PROPOSED

In order to validate the teaching strategy described, the following parameters were analyzed: a) Changeability (time taken to make a change), b) Time for project development, c) Correlations between the above parameters, d) Functionalities that are not working properly, e) Students' grades.

In this section, sample 1 (29 groups of three students each) represents the students who used the MOSS manuals. Sample 2 (22 groups of three students each) represents the students who used the new strategy described in this paper.

A)

B)

C) Correlations Between Parameters. A comparison was made of correlations between: Time spent on phase ADD vs Changeability; Time spent on phase ADD vs test; Changeability vs test. The correlation is the degree to which two variables are related; the most common measure of correlation is Pearson's correlation.

1) Time spent on phase ADD vs changeability

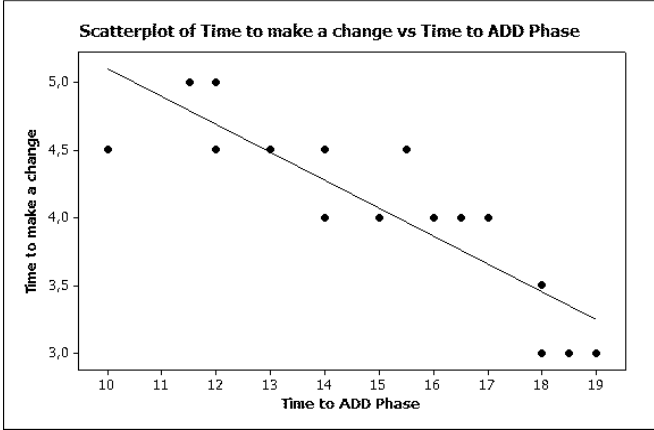


Fig 10 Correlation Time dedicated to ADD vs Time taken to make a change

Pearson -0.889

The correlation is negative, so while the time dedicated to the ADD phase increased, the time taken to make a change decreased. It is worth spending more time during the ADD phase using software engineering techniques to develop projects where the changeability is higher (Fig. 10). A Pearson value of -0.889 can be interpreted as the changeability of a project has a 77.44% dependence on the time dedicated to the ADD phase.

2) Time spent on the ADD phase vs the test phase

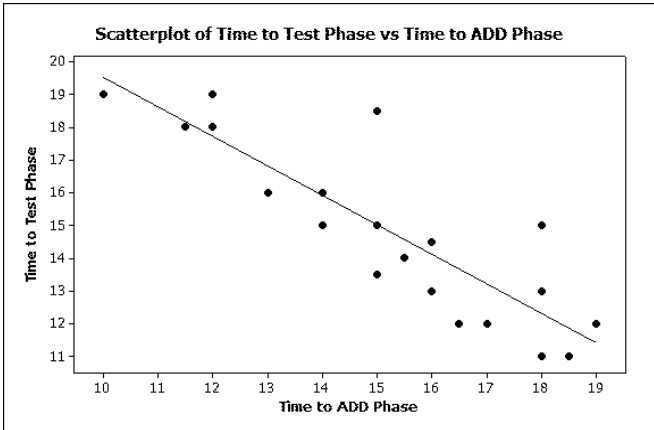


Fig. 11 Correlation Time to ADD phase vs Time to Test Phase

Pearson -0.880

The correlation is negative, so while the time dedicated to the ADD phase increased, the time dedicated to the test phase decreased. It is worth spending more time during the ADD phase using software engineering techniques to develop projects where the test phase is shorter (Fig. 11). A Pearson value of -0.880 can be interpreted as the test phase time of a project has a 75.69% dependence on of the time dedicated to the ADD phase.

3) Changeability vs test

As can be seen in Fig. 12, the time to develop the ADD phase is correlated with the time to make a change, and the time to develop the ADD phase is correlated with the time dedicated to the test phase. A new correlation was identified between the time to make a change and the time to develop the test phase. This is a positive correlation so as the time dedicated to the test phase increases the time spent on making a change also increases with a Pearson coefficient 0.734.

The interpretation of this correlation may be that the less time spent on the ADD phase, the fewer software engineering techniques were used, so the test phase is longer as is the time dedicated to making a change, due to the low maintainability of the software developed.

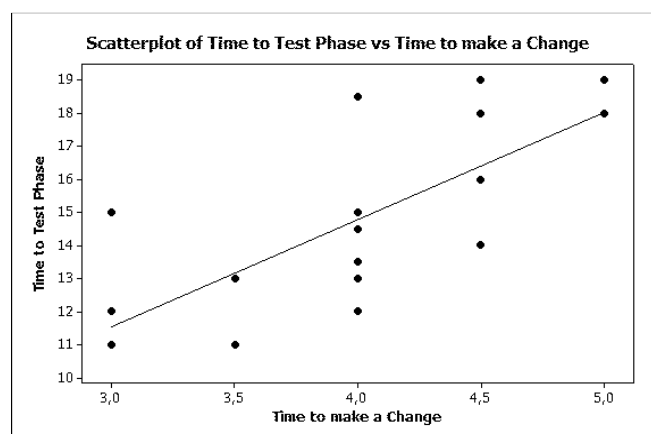


Fig. 12 Correlation between Time spent to make a Change vs Time spent on Test Phase

D) Functionalities Not Working Properly. An analysis was made of the number of functionalities identified as *not working properly*, and the correlations: Time spent on the ADD phase vs number of functionalities not working properly; Time spent on the Test phase vs number of Functionalities not working properly.

1) Number of functionalities *not working properly* according to the web portal requirements specifications.

The number of functionalities not working properly was stored, after developing teaching strategy step 4, and the analysis of the data obtained is summarized in Table X and Fig. 13. A T-test and the representation of the Interval-plot were used. H0: the functionality not working properly of both samples is the same.

Var 1: functionality not working properly in a project where the MOSS manual was used (sample 1).

Var 2: functionality not working properly in a project where new reuse and reengineering were used (sample 2).

Table X Functionality not working properly. Statistics Summary

	Var 1	Var 2
Count	29	22
Average	7.75	6.0
Mode	8	6
Variance	3.97	3.71
Std Deviation	1.99	1.92
Minimum	5	3
Maximum	13	9
Range	14	8
Std. skewness	1.74	0.25
Std. kurtosis	0.61	-1.03
T student	3.18 DF:49	
P - Value	0.003	

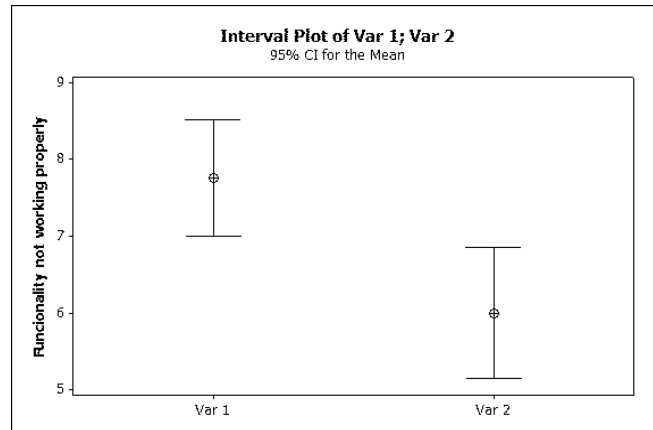


Fig. 13 Interval Plot – Number of functionalities not working properly

With a p-value of 0.003, lower than 0.05, H_0 can be rejected. So the number of functionalities not working properly of both samples is different. And as can be seen in Fig. 13, the number of functionalities not working properly was lower in projects where the new strategy was applied, due to the use of software engineering techniques.

2) Time dedicated to the ADD phase vs number of identified functionalities not working properly

This analysis was made using the data obtained after applying step 4 of the proposed strategy.

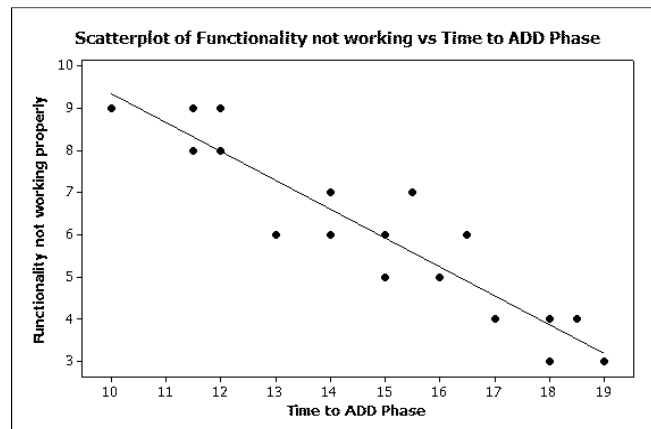


Fig. 14 Correlation Time dedicated to ADD phase vs identified functionalities not working properly

Pearson -0.937

The correlation is negative (Fig. 14). So, while the time dedicated to the ADD phase increased,

the number of functionalities identified whose performance did not meet the project requirements decreases. A Pearson value of -0.937 can be interpreted as the number of functionalities not working properly in a project having an 87.79% dependence on the time dedicated to the ADD phase. As a conclusion, the better the analysis and design of a collaborative web portal, the better will be the translation of the software requirements to collaborative functionalities.

3) Time taken on the test phase vs identified number of functionalities not working properly

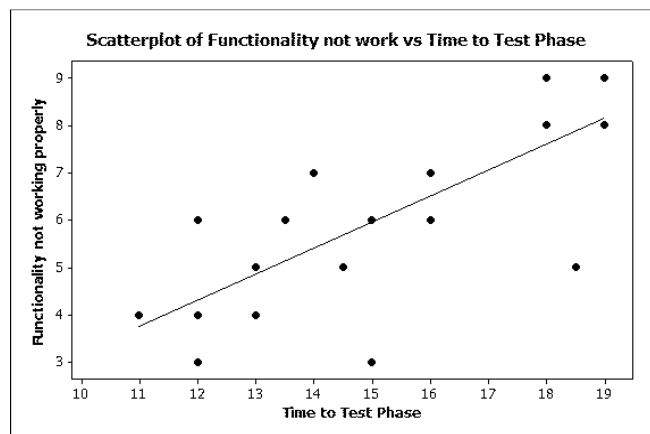


Fig. 15 Correlation Time dedicated to Test phase vs identified Functionalities not working properly

Pearson 0.770

The correlation is positive (Fig. 15). So as the time dedicated to the test phase increased, the functionalities, whose performance was not in keeping with the project requirements, increased.

A Pearson value of 0.770 can be interpreted as the time spent on testing a collaborative web portal having a 60% dependence on the number of identified functionalities that do not satisfy the web portal specifications. This is because the better analyzed and designed the web portal is, the less time needs to be spent on the testing phase, and the fewer number of non-working functionalities are identified because the requirements are better translated into code.