Acceptance Test Report (ATR):
Group 17

ETSF01

Richard Berntsson Svensson

Tommy Ivarsson, ada09tiv
Oscar Olsson, ada09ool
Jonas Klauber, ada09jkl
Fredrik Pettersson, ada09fpe
Erik Westrup, ada09ewe
Simon Thörnqvist, ada09st1

0. This test should be performed by 4 persons (as users) and ¾ of them must be able to install and figure out how to use the tool within 5 minutes.
Requirements: QR-02

1. This test should be performed by 4 persons (users) and ¾ must be able to complete the tasks within 15 minutes. Read and follow the manual and do an estimation. We must be able to understand the information presented. Make sure that the manual and GUI language is in English.
Requirements: QR-01, QR-02, FR-06
Comments:

2. Create a file with 1000 rows (projects) and follow the manual and do an estimation.
Requirements: FR-01

3. Find out, by reading the manual, how we can add up to 20 project characteristics and do so.
Requirements: FR-01, FR-02

4. Run the example .arff file (http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa.arff) and see if the result is presented in all units required.
Requirements: FR-01, FR-03

5. Run the previous testfile and verify that we can set the threshold. Set it to 0 and verify that all projects are used. Set it to 1 and verify that none or few projects are used. Set it to 0.3 and verify that a reasonable amount of projects are used. Verify that the threshold is in the interval [0,1}
Requirements: FR-04, FR-05

6. So create a new arff-file with only three projects in. Then load this file to the program and compare with a project that does not match any of the three projects. Set the threshold to 1 and verify that the program warns.
Requirements: FR-05

7. Delete one row at a time from the sample .arff file and enter its values as the compared project in the GUI. Verify that the estimate is within +- 30% of the actual effort. Do this for 10 projects. Do this with the threshold 0.35 and organic-mode. 7/10 projects should fulfill this requirements as agreed with the developers. The 10 projects that were used are in the Appendix.

Requirements: QR-03

Test no.	Result {Failed, Passed}	Failure severity {Low, Medium, Critical}	Comments
0	PASS	-	4/4 users completed within 5 minutes. Strange that you first need to open a terminal and then double click the .jar file. And it did not work to double click on the .jar, had to right click and "run as".
1	PASS	-	4/4 users completed within 5 minutes.The buttons in the main dialog were not ordered in the order they (usually) are used.
2	PASS	-	Fast!
3	PASS	-	Passed after demo. Perhaps should have been clearer from user manual.
4	PASS	-	Splendid!
5	PASS	-	
6	PASS	-	No problems here!

7	FAIL	LOW	With the test parameters used we could not get estimates within the range. However we don't know what threshold is the optimal and should be used to achieve this requirement. Also the projects chosen might have been the bad ones, we don't know. Also we did not know what project type to use so we used Organic.

Other comments

It would be nice if the individual GUI components and dialogs could be merged to as single window. That would be a lot easier for the users. To avoid confusion it should not possible to open many instances of the dialogs.

Summary

Overall the test went out well and the program works as it was described (for the most part). It seems to give good enough estimations based on previous projects. In the construction of our test we follow the Project Mission to be sure that the testing covered all requirements. A few tweaks in the installation process should be added so that the instructions actually work. Also the cognitive aspects should be considered. As customers we are satisfied with the product. Good work.

Appendix

Data for test 7.

Nominal,Nominal,Nominal,Nominal,Nominal,Low,Nominal,High,Very_High,Very_High,Low,High,High,Nominal,Nominal,190,420   %   instance number: 51 PASS
Nominal,Nominal,High,Nominal,High,Nominal,Nominal,High,High,Nominal,Nominal,High,High,Nominal,High,47.5,252   %   instance number: 52 FAIL
Very_High,Nominal,Extra_High,High,High,Low,Low,Nominal,High,Nominal,Nominal,Nominal,Low,High,Nominal,21,107   %   instance number: 53 FAIL
Low,Nominal,Nominal,Nominal,Nominal,Low,Low,High,High,Very_High,Nominal,High,Low,Low,High,423,2300   %   instance number: 54 FAIL
High,High,Nominal,Nominal,Nominal,Low,Low,Nominal,High,High,Nominal,High,Nominal,Nomin

al,Nominal,79,400   %   instance number: 55 FAIL

High,High,Low,Nominal,Nominal,Nominal,High,High,High,Nominal,Nominal,Nominal,High,Nomin al,Nominal,284.7,973   %   instance number: 56 Not needed to test

Nominal,High,Low,Nominal,Nominal,High,Nominal,High,High,Nominal,Nominal,Nominal,High,Hi gh,Nominal,282.1,1368   %   instance number: 57 Not needed to test

Nominal,High,High,Very_High,Nominal,Nominal,High,High,High,High,Nominal,High,Low,Low,Hi gh,78,571.4   %   instance number: 58 Not needed to test

Nominal,High,High,Very_High,Nominal,Nominal,High,High,High,High,Nominal,High,Low,Low,Hi gh,11.4,98.8   %   instance number: 59 Not needed to test

Nominal,High,High,Very_High,Nominal,Nominal,High,High,High,High,Nominal,High,Low,Low,Hi gh,19.3,155   %   instance number: 60 Not needed to test