

Peer Test Plan

PMoC
3/24/2003
Version 1.1

T-76.115 Peer Test Plan

PMoC

Version	Date	Author	Description
1.1	03.03.2003	Jakobsson	Updated the testcases
1.0	24.02.2003	Jakobsson	First version

Table of Contents

1. Introduction	3
1.1 Purpose and scope of the peer test plan	3
1.2 The product and environment	3
1.3 Contacting PMoC	3
2. Testing instructions	3
2.1 Requirements	3
2.2 Testing tasks	3
2.3 Running the tests	4
3. Item pass/fail criteria	4
3.1 Fault levels	4
3.2 System and acceptance test criteria	5
4. Suspension criteria and resumption requirements	5
5. Test deliverables	5
5.1 Test report	5
5.2 Bug reports	5
7. Risks, contingencies and assumptions	6

1. Introduction

1.1 Purpose and scope of the peer test plan

The purpose of this peer test plan is to describe how testing should be performed by the peer testing group. This plan is a cut back version of the PMoC groups test plan, since there are many things that the peer group neither needs nor wants to know.

1.2 The product and environment

The proconf system is a prototype of a software product that can be used for creating process and automation flow diagrams that could be used with a process simulation software that the customer uses. The new software is used to verify whether or not the GML-specification (Gallery Markup Language) together with graphics defined through SVG is a good solution for creating, storing and managing the configuration data related to simulation models.

The product will mainly be used by the researchers at VTT. As in use of the customer, the target software environment will be Win32 platforms running on normal personal computers, with additional software as needed.

The proconf system utilizes the Batik libraries from Apache for rendering and handling SVG graphics.

1.3 Contacting PMoC

If problems or questions arise during testing, the testers should contact Markus Jakobsson, <majakobs@cc.hut.fi> for help and guidance.

2. Testing instructions

2.1 Requirements

The peer group is required to have a computer with windows 2000 or windows XP operating system, and a Java 1.4.1 runtime environment.

The peer group will be given the following for successfully completing the testing:

- Proconf software and data, including batik libraries
- This peer test plan
- Peer test report template
- User guide
- Access to burana account

2.2 Testing tasks

The test cases for the peer testing are summarized below. The details of the test cases can be found in the test report template. This is to make the testing easier as only one document is needed when running and reporting tests. Detailed instructions for executing the test cases can be found in the user manual.

Table 1: Features to be tested during peer testing in phase I3

Items to be tested	Feature to be tested	Priority
fi.vtt.proconf.*	PTC_1: Drawing free symbols	1
	PTC_2: Moving graphical objects	1
	PTC_3: Make free symbol a terminal symbol	1
	PTC_4: Add terminals to symbol	1
	PTC_5: Attach symbol to type	1
	PTC_6: Create new component	1
	PTC_7: Open net	1
	PTC_8: Add components to net	1
	PTC_9: Move components and graphics in net	1
	PTC_10: Delete component	2

2.3 Running the tests

Installing the proconf system is not part of the testing. Any faults in the installation should be notified to the group but not recorded in Burana.

The test cases should be run in the order they are numbered. This is because the test cases are often dependent on the execution of previous test cases

We ask you to read through the chapters in the user manual before commencing with testing. When testing, first run the test cases as stated in the user manual, after which you should test each test case again with a more destructive attitude. By going through it once you will get a better picture of what the system does and understand when something goes wrong.

3. Item pass/fail criteria

3.1 Fault levels

System and acceptance tests have the following levels of faults.

Table 2: Fault categories

Fault category	Description
Critical	A vital part of the system is not functional.
Serious	Part of the system has a malfunction, causing errors with vital parts of the system in certain situations.
Normal	A non-vital feature of the system is not functioning properly.

Minor	A bug that does not affect functionality.
-------	---

When a bug is discovered in the system, it is entered into the Burana bug tracking system. The bug is given a severity and an internal priority, as shown in the tables above and below. The table also states the required actions of the PMoC group when a bug is encountered. The peer group should not set a responsible person.

Table 3: Internal priorities for bugs

Internal priority	Explanation and required action
Low	The bug is fixed if there is time and no higher priority bugs or features are left. Usually corresponds to fault level Minor.
SemiLow	The responsible person attends to the problem, but the rest of the group can continue as normal. Usually corresponds to fault level Normal.
Before release	The same as the Low or Semilow, but the bug has to be fixed before a release.
High	The development of the part of the system that is not functioning properly is frozen, and all efforts of persons involved with the subsystem are directed to fixing this bug. Usually corresponds to fault level Serious.
Urgent	The bug must be fixed immediately, before developing any other parts of the system. Usually corresponds to fault level Critical

3.2 System and acceptance test criteria

A test is accepted even if it contains a fault of low category. Low category faults (for example small visual bugs) are corrected only if the customer sees them as a real nuisance and there is time to correct them.

4. Suspension criteria and resumption requirements

While running a test, testing can be suspended if the system has a critical fault that prevents the test from being run or if the environment cannot be setup so that the test can be run. The PMoC group should immediately be contacted. Testing is resumed only when the reason for suspension is found and the fault is corrected.

5. Test deliverables

5.1 Test report

The peer group is to deliver a full test report according to the peer test report template.

5.2 Bug reports

Any found bugs shall be reported to Burana according to the fault categories listed in table 2. The PMoC group then sets the internal priorities listed in table 3.

7. Risks, contingencies and assumptions

The table below suggests that ineffective testing due to lack of understanding in the system is the greatest risk associated with the peer testing. The relatively high probability is due to the fact that the domain is abstract and difficult to understand in just a few hours.

The severities of the risks are graded on a scale from 1 to 5 where 5 is the most severe impact on the testing of the project.

Table 5: Risks of the peer testing

Risk	Probability	Severity	Action	Impact
Installation is unsuccessful	10 %	5	The peer group can be provided a working environment in Maarintalo.	Time is wasted for both groups.
Testing is ineffective, due to difficulties in understanding the proconf system	50 %	4	Testers should ask for help from the PMoC group.	Only unimportant minor level bugs are found, while the important parts of the system will be left untested.
No testing motivation.	25%	3	Make test cases easier and fewer.	Peer groups test results are not worth much.