

Use cases –
Process model configuration using SVG
PmoC
14.4.2003
Version 1.4

T-76.115 Use cases

PmoC

Version	Date	Author	Description
1.0	28.10.2002	Björn Forss	First version.
1.1	2.12.2002	Björn Forss	The version after phase I1.
1.2	7.2.2003	Björn Forss	The version after phase I2.
1.3	20.3.2003	Björn Forss	The version after phase I3
1.4	14.4.2003	Björn Forss	Final version

1. Introduction

It is difficult to get a common picture of what requirements the system being developed has. Use cases help the customer and the group members to agree on the requirements and the functionality of the system. Use cases show from a user perspective the functionality of the system. Use cases help to avoid misunderstandings between the customer and the project group.

2 Why use use cases in this project

The requirements documentation is the most important description of what the outcome from the project should be. The project needed an effective and clear way to make the requirements in order to success. Use cases simplify the writing of the requirements documentation. Use cases also give priority to different functionality in the project. They help in planning the implementation order of the different functionalities.

3. Possible solutions

To write the requirements documentation without using use cases seem much more time consuming. The alternative is to picture the functionality only in plain text. Use-cases in form of tables are easier to read, change and find relevant information from. They also decrease the amount of written text needed.

4. Implementing use cases

The use-cases were primary implemented in table form in this project. The table form was easy to edit and maintain. Also the table form was easy to read and find relevant information from. As the project moved forward, more detailed use cases was made to complement and update the requirements.

Some use cases were also implemented as diagrams. These were used to clarify the different roles of different users.

Table 1: Example of use-case table

Name: Name of the case.

Summary: Summary of the case.

Actors: Actors involved in the case.

Preconditions: The situation before this case can happen.

Basic sequence: Detailed description of the actions during the case.

Exceptions: Exceptional situations that can occur if the case could not be fulfilled as planned.

Postconditions: The situation after that the case has happened.

The use cases were also given a priority according to their importance. They were divided into *core functionality* (must be implemented; use of the program requires this), *desirable functionality* (should be implemented; adding this would make the program more useful) and *optional functionality* (can be implemented; may simplify some tasks).

5. Experiences so far

5.1 Planning phase

We noticed that the use of use cases clarify the differing opinions of the customer and the project group. Use cases helped to reach a common understanding of the system functionality and the goals of the project. Use cases demand a lot of writing, but help to get a better understanding of the actual case. The use of diagrams did not seem as meaningful as using the table form. The only advantage we found in using diagrams so far was that it makes it easier to show the different users' different roles.

Writing the use cases may lock the design and functionality of the system. Thinking of individual use cases may also distort the vision of the project as a whole. Alternative ways to react in the situation of the cases may not be considered. It is important that the use cases are there to help develop the common understanding and functionality, not to hide other possibilities.

5.2 Implementation phase 1

No new use cases were created during the implementation phase 1. The use cases created during the planning phase were enough for the implementation phase one and possibly also for implementation phase 2. In the beginning of the implementation phase one where the use cases from planning phase prioritized. The priority helped in planning the implementation and in giving an overview of the importance of different functionality.

5.3 Implementation phase 2

Several use cases were changed during phase 2, course of new agreements on the functionality of the system. The Graphical User Interface (GUI) were discussed in more detail for the first time during this phase, which made it possible to specify the functionality in more detail. The customer had new ideas of the functionality that made it necessary to reconsider the requirements documentation and the use cases. The customer seems to prefer to discuss functionality from the GUI design point of view, instead of discuss use cases. This made the process of using use cases more a group intern process. The use cases were updated according to the GUI design discussions with the customer.

The use cases were used as base for the system tests during the second implementation phase. The functionality of the system were tested according to the functionality described in the requirements document. The use cases worked as the test cases. The use cases were very useful for this purpose and it was easier to see the

big picture of the functionality of the system after these testes. Using the use cases instead of writing special test cases saved us much valuable working time.

5.4 Implementation phase 3

The use cases were well updated thoroughly during implementation phase 2, so there where little need to change them during implementation phase 3. The implementation of functionality according to the use cases worked well during this phase.

As in implementation phase 2 the use cases where used as base for the system tests. The use cases worked as test cases.

5.5 Delivery phase

During the delivery phase naturally no new use cases were created, course no new features were specified for the system. The existing use cases in the requirements document were updated in the beginning of the delivery phase to describe the final agreements and scaling of the project with the customer.

The use cases were also used in this phase (like in earlier phases) as test cases. The functionality was tested according to the use cases in the requirements document.

6. Summary

Making the use cases with the customer clarified a lot of the different opinions between the project group and the customer. The use cases were gone though during customer meetings to make sure that the project group understood the goals of the project. The use cases were also prioritized during customer meetings together with the customer. These customer meetings and the cooperation between the customer and the project group was vital for the project, and the use cases was an important element in the communication between the group and the customer.

During the whole project the use cases were also used as test cases. This was time saving for the project since no extra test cases had to be made. The use cases worked well for testing user functionality of the program.

The use cases used was mostly in the table form shown earlier in this document. The table form turn out to be a good choice. They were easy to read and it was easy to find a relevant use case. The table form was also easy to update which contributed to the frequent updates of the use cases as the project evolved. The diagram form of the use cases were not used as frequent as the table form. They seemed not so relevant to the project. The different actors were identified early in the project. The diagram use cases were also harder to update course of the software for making them was not present on most of the workstations used in the project.