Experiences with a UML-based Development Method for SIL products

Kai Hansen[1], Bente Anda[2], Ingolf Gullesen[1]

[1]ABB Corporate Research Center
P.O. Box 90, N–1361 Billingstad, Norway
{kai.hansen,ingolf.gullesen}@no.abb.com

[2]Simula Research Laboratory,
P.O. Box 134, N–1325 Lysaker, Norway
bentea@simula.no

Abstract. This article describes the results of an investigation into the suitability of methodology based on UML in an IEC 61508 development project that developed SIL products. Feedback was gathered from project members through interviews and questionnaires. The results show that the method led to improvements in some aspects of the development process that are particularly important in the development of SIL products. These improvements were related to documentation, traceability from requirements to code, and the quality of the code. The results also show, however, that the opinions of the project members differed a lot, and consequently that it is difficult to define one method with which a large number of developers with different backgrounds and mind sets will be content.

1. Introduction

ABB is a global company operating in around 100 countries and employing around 103,000 people. ABB is a leader in power and automation technologies and develops software and hardware solutions for these markets. The organization has a large number of development projects, and the majority of the projects develop also embedded software where development of special hardware is included. Many different methods, programming languages and software tools are used, and until recently there have been few attempts at common streamlining of software development.

For development of safety products used in plants or installations where the process can be dangerous to humans or damage the environment, ABB's products must be certified according to the new international standard IEC 61508 [5]. The standard IEC 61508 is becoming a requirement in most countries for process industry and partly also in discrete manufacturing. This standard is a life-cycle standard and includes requirements on software development methodology. For some safety levels of software it highly recommends the use of semi-formal development methods.

ABB has also identified that there is potential for improving development projects by adapting ”state of the art” methodology in analysis and design. ABB already uses a gate model for project business decisions which defines the milestones for decision making in a project [1]. Consequently a UML-based development method which qualifies as a semi-formal method, was defined. The development method was applied in a large, international development project. This project developed a new version of a safety certified product (in addition to other non-safety related functionality) and involved approximately 230 people at four locations in three countries. The development included software, hardware and VHDL code, and much of the development was concerned with modifying or integrating with existing systems.

2. The Development Method

This UML method describes the requirements analysis and design phases in a defined ABB V-model for development. This method was developed internally in ABB [6,7]. It was not based on any particular method for UML-based development, but those responsible for it had experience with development based on UML, and were familiar with basic literature on such development, for example [2,3,4].

The method is document-driven in the sense that documents based on predefined templates are produced, constituting important milestones in the development process.

Requirements analysis:
R1. Identify actors and use cases, and document them
R2. Group use cases and actors into subsystems
R3. Refine the use cases and identify dependencies

Analysis:
A1. Describe flow of events inside the use case (textual)
A2. Create high-level sequence diagrams
A3. Define interfaces between use cases in different subsystems
A4. Describe the activities in the use case in an activity diagram (Optional)
A5. Create high-level class diagrams
A6. Update sequence diagrams with correct high level class and operation names

Detailed Design (Note that the hardware developers did no detailed design):
The goal of this phase is to realize the high-level classes with implementation class diagrams and to group the classes in components. State transition diagrams may be used in the process of elaborating the class diagrams. The detailed classes are connected to the high-level classes through a “realize” association.

Figure 1. The UML-based development method

The main reason why UML-based development was chosen as a basis for a semi-formal development method was the good tool support for modeling with UML. A brief overview of the steps of the method is given in Figure 1. The relations between the V model, the Gate model and the ABB UML method are sketched in Figure 2.

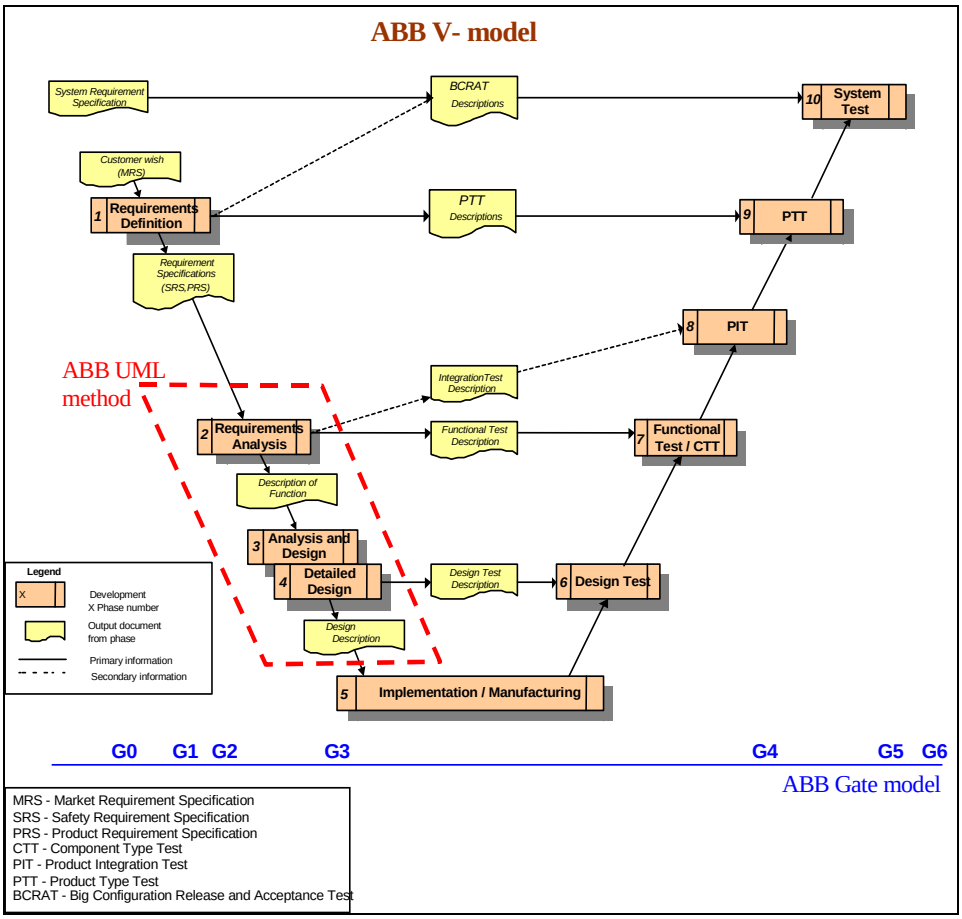

Figure 2 The ABB development models.

3. The Project

The goal of the project was to develop a new version of a safety-critical system based on several existing systems. This software project was ABB's most ambitious project regarding quality assurance in that it followed the requirements of IEC 61508. C and C++ were used in the software implementation. UML version 1.3 and Rational Rose was used for modeling.

The project consisted of several sub-projects. The three sub-projects that developed safety software and applied the ABB UML method are described briefly below.

- Sub-project A, the largest one, developed software based on a comprehensive existing code base. This includes code running both on a Windows PC platform and on an embedded 32 bit RISC processor.
- Sub-project B developed hardware and embedded software for a 32 bit RISC processor. The project was divided into two teams: the hardware team, which dealt with electronic and mechanical design, and the software team. This sub-project had no existing systems to relate to.
- Sub-project C developed C code in the form of embedded software for a 16 bit processor. This sub-project was the only one that generated code automatically from their UML models. It also included development of VHDL code and hardware. This sub-project had no existing systems to relate to.

The product developed here is used in various applications in process industries. Important applications are in chemical industry, petrochemical industry, oil installations and platforms and burner control. This product has a significant part of the world market in these market segments. The majority of the functions are in the safety level which in the IEC 61508 is defined as SIL 2, while some functions require the level SIL 3. The application is typically monitoring fluids or gas or it has sensors indicating fire and will take actions such as changing position of valves or other actuators in a process plant. Development of products according to a specific SIL level will ensure that the engineering of a plant will obtain the quality required.

4. Collection of feedback

ABB wanted to gather experiences from applying the UML-based development method in order to improve the method and extend its use in the organization. This was done in two phases:

1. Interviews were conducted with 16 project managers and developers with experience from the project. The interviewees represent different locations, different kinds of development and different roles in the project. The interviews were performed in the middle of the project in order to establish knowledge of the situation in an early phase after introducing a new development methodology. This shows the situation as a snapshot in a projects live-cycle.
2. A total of 55 project members responded to a questionnaires were at a late stage of the development project. The questionnaire consisted on statements on about UML-based development, the respondents indicated to what extent they agreed to statements on a Likert-scale. An example of a question is given in Figure 3.

All the developers interviewed had applied the ABB UML method. A couple of the managers had applied the method, while others had reviewed project documents with UML. Some of the interviewees had positive experience with applying UML in previous projects, but these projects had been smaller than the project and had not had to satisfy the same safety requirements.

The UML method improved traceability between requirements and code

☐ *Totally agree*	☐ *Partially agree*	☐ *No opinion*	☐ *Partially disagree*	☐ *Disagree*

The UML method improved functional testing

☐ *Totally agree*	☐ *Partially agree*	☐ *No opinion*	☐ *Partially disagree*	☐ *Disagree*

Figure 3. Examples of questions

5. Expectations and Overall opinion

Most of the project members had positive expectations to using the method before the project, and they were also positive to reusing it on future project after this first experience with it, Figure 4.

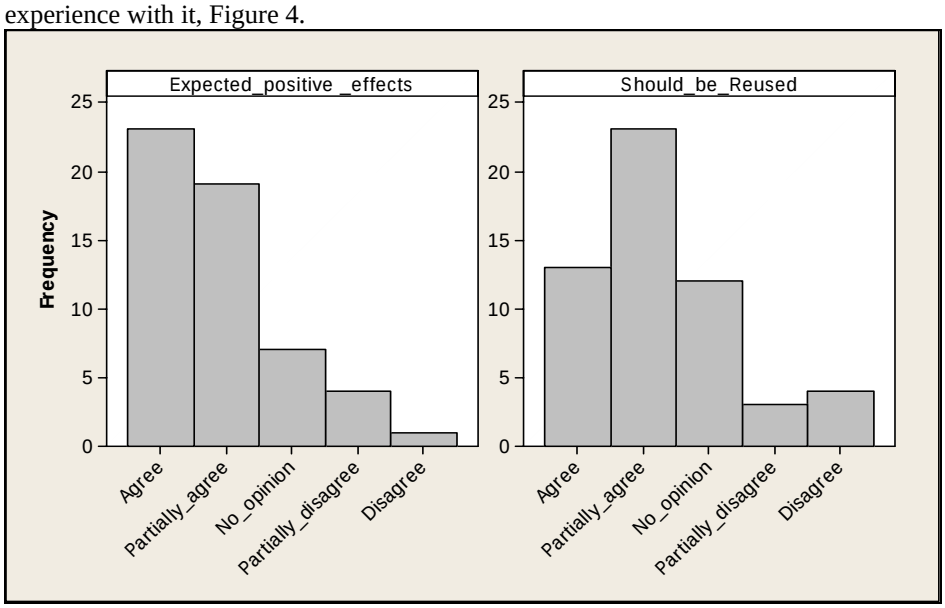

Figure 4 Expectations and opinions on whether the method should be reused

In the interviews we learned that there were three main reasons for the positive expectations that most of the project members had: 1) They experienced a need for more methodological support for software development, 2) The use of UML has become a de facto standard in industry and many of the developers were therefore eager to learn more about it, and 3) Those who had used it before had mainly positive experiences. We see from Figure 3 that most of the project members were positive to reusing the method on future project, but that they were less positive after the project than before. The main reason for this according to the interviews was that they found the use of UML, the method and the tool Rational Rose to be more difficult than what

they had expected initially. Although they had experienced benefits from using the method, these were smaller than expected.

Most problems were related to understanding and applying the method. The project members had not experienced many problems related to the syntax of the UML although some thought that sequence diagrams (in UML version 1.3) lacked some expressive power with respect to loops and conditions. We believe that this reflects the fact that defining and adopting a development method is more challenging than learning the UML notation, but we also expect that the developers may have more opinions on the syntax when they get more familiar with UML. There were also some problems related to learning to use the tool Rational Rose, but these were few compared to the problems with the method.

6. Experience and Training

At the start of the project, most of the developers attended courses of two to five days that covered UML syntax, Rational Rose tools and the ABB UML method. A special team, the UML team, was set up to help the rest of the project with the use of UML, which included responsibility for developing templates and for reviewing documents, with particular focus on the correct use of UML. The number of people in the UML team varied from three to five over the course of the project.

Both the interviews and the questionnaires revealed that the project members were not very satisfied with the training. The interviewees reported that there had been too little training because managers, reviewers and testers, who did not themselves develop, did not receive training even though they had to read and understand the models. Developers who started on the project after the courses did not receive the same training as the others. Furthermore, the training had not been sufficiently adapted to this particular development project.

The questionnaires showed, however, that those who received training were not more positive to the method than were those who had not received training in it. On the other hand, those who had previous experience were more positive than those who had not. In our opinion, these results indicate that all project members should receive some equal basic training adapted to the particular project, but that most must be learned during the project, something which leads to the needs for much mentoring during the project to ensure correct use of such a method.

7. Benefits of the method

The opinions on what were the benefits of using the method are shown in Figure 5. We see from the figure that a majority among the developers and project managers found that the documentation of the project and product had improved when compared to previous ABB projects. They found that it was easier to read documents that had a common format, and also that many of the project members were better at expressing themselves with UML than they had been when they had used plain text in English.

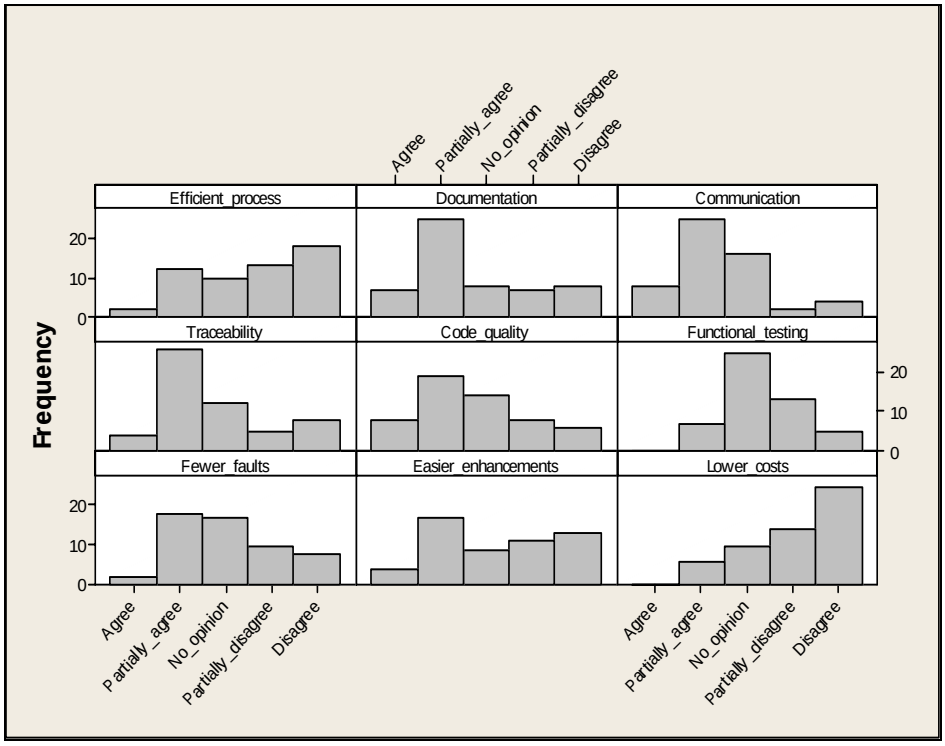

Figure 5 Effects of using the method

The communication in the development teams had improved due to having the UML documents as a basis. It had become easier to trace from requirements to code and test when requirements were mapped to use cases which again were mapped to a set of classes.

There were some improvements in the quality of the code due to more emphasis on design before coding. Figure 5 shows that the method did not improve functional testing as such, but that there are some indications of fewer faults. We know from the interviews, that many of the project members found the UML diagrams, in particular the use cases, to be valuable input to making functional test cases, and that this had a positive effect on the coverage of the test cases and consequently on the fault rate in the project.

Nevertheless, the interviews also revealed that the UML diagrams were often not used as input to testing because the method lacked guidelines on how this should be done, and the testers often lacked training in UML. We believe that this explains the results from the questionnaire on this issue.

Some of the project members expected that future enhancements to the system would be easier than what was typically the case due to having good documentations. Others thought that since the method was not particularly aimed at maintenance, enhancements would probably not be easier.

Both in the interviews and in the questionnaire, the project members expressed that there had been high costs involved in introducing this method. Many of the interviewees thought that they had spent approximately twice as much effort as usual on analysis and design. However, most of the project members thought that the costs had not been too high since they realized that there must necessarily be high learning costs associated with the introduction of any development method.

8. Challenges

The main challenge experienced in this project was that of having to use the method not only in developing new functionality, but also in modifying existing systems. Even though systems development is often about modifying existing system, there is little methodological support on how to use UML in such development. The developers who had been involved in modifying and integrating with legacy software had in general obtained fewer benefits from using the method than had those who had developed from scratch. Examples of problems related to having legacy code were:

- It was very time consuming and consequently costly to reverse engineer the existing system to UML diagrams, so that was only done to a limited extent.
- It was very difficult to describe functionality at an appropriate level of abstraction when code was already available. For example, in the analysis phase they were required to make high-level classes while they already had design classes, an activity that was considered superfluous by many of the developers.
- In the reviews many of the reviewers tended to know the existing code and therefore focused more on that than on functionality

9. Conclusions

The investigations discussed here do not give uniform answers to the questions raised. The project members had widely different points of view on many of the topics. This probably reflects that software developers have different background, mindset and ways to work, and that there so far is not a comprehensive set of methodology elements that for everyone yields the best results of software products and the safest software.

However, some statements about improvements due to the method had a clear majority of support. This was the case for documentation, communication, traceability and the quality of the code. All these aspects are important for the safety of the final products, and we can thus state that the investigation give support to a claim of better safety abilities in equipment developed according to IEC 61508. It was less agreement around the effect that the method would have on future enhancements of the product, and thus on future costs. It was agreement that the work process used is not more efficient than traditional methods. These negative results are problematic for the product and project from a financial point of view, but are not so important viewed from the safety side.

One positive remark is that in general the developers did experience the method as a positive thing and would consider using it again in a new project.

We think that the progress of safety software development must go through such deployment of methods at large scale projects followed by an investigation as done here. Taking feedback from the investigation into modifying the method in small steps by correcting problems and keep the positive parts is the way forward into improved safety software development. We believe that UML used in a well defined way is a good way to develop safety software.

10. References

1. ABB Gate Model for Product Development 1.1 tech. report 9AAD102113, ABB/GP-PMI, Västerås, Sweden, 2001.
2. Booch, G., Rumbaugh, J. and Jacobson, I. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.
3. Douglass, B.P. *Real Time UML: Advances in the UML for Real-Time Systems*. 3rd Edition, Addison-Wesley, Boston, MA, 2004.
4. Fowler, M. UML Distilled. *A Brief Guide to the Standard Object Modelling Language*, 3rd edition. Addison-Wesley, 2003.
5. IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. 1998. (http://www.iec.ch/).
6. The ABB Instruction “Software and Hardware development”, 2001.
7. The ABB Guideline “Guideline for use of semi-formal methods in Software and Hardware design”, 2003.