REAL-TIME CONTROL SYSTEMS: A "ONE DOCUMENT" OBJECT ORIENTED DEVELOPMENT PROCESS

G.Chiozzi, ESO, München, Germany
J.M.Filgueira, GTC, Tenerife, Spain

Abstract

The Object Oriented community is converging towards a unified development process driven by Use Cases[1], where the Unified Modelling Language (UML)[3] has become the standard analysis and design language.

Control systems for scientific experiments are typically designed and developed in parallel by many different groups and user requirements evolve with time.

It is therefore necessary to adapt the process to these constraints.

Our experience shows that:

- Use Cases are an effective way of capturing requirements and are well accepted by the customer as a base for discussion.
- Use Cases are also a good base for modular and integration testing and for requirements tracing.
- All project's documentation must be for the whole team "one document", easily accessible online for consultation and update. Hypertext navigability provides road maps through the document and prevents information redundancy and obsolescence. The Web can be a good solution, but also case tools can provide the same functionality.
- The process must be able to produce the standard paper deliverables, defined in IEEE[5] or similar Project Management Plans.

A wide collaboration between organisations in the definition of a unified development process for such control systems can provide significant timesaving in the projects and must be pursued.

1 INTRODUCTION

Recognising the need to cooperate between different organisations in the definition of a Development Process for Control Systems, an inter-organisational working group has been founded. It is formed by the control software development teams of European Southern Observatory, Gran Telescopio Canarias, Synchrotron of Trieste and Astronomical Observatory of Trieste.

Purpose of the working group is not to identify a unique SW Development Process, suitable for all our organisations and projects, but to adapt the core Unified Software Development Process[4] to our specific needs.

During the first phase, the collaboration has involved mainly ESO and GTC, since these two teams are in a more advanced stage in the application of the new Development Process. In particular the whole SW development for the GTC is based on the Unified Process, while ESO is adopting the process for the development of control system for the Auxiliary Telescopes[6] of the VLTI (Very Large Telescope Interferometer). We now plan to have the other two teams more and more involved in the discussions.

2 ONE DOCUMENT CONCEPT

We have learned from the experience of our previous projects that keeping the documentation up to date and readily available to all the interested parties is a major problem.

We have then put as our objective that of having *one document*. This means that:

- All the relevant documentation of the project is kept in one single, centralised, repository
- No information is duplicated, but it can be simply cross-referenced wherever needed.
- The latest version of the whole documentation is always available for immediate access.
- Changes can be easily done, but everything is, at the same time, kept under configuration control.
- Requirement and design documents and the actual implementation code are seamlessly integrated. It is possible to navigate from one to the other and back.

We have seen that this objective can be actually reached in different ways:

- The ESO group has chosen an HTML based approach. HTML pages are at the core of the documentation system. Various tools are used to produce the HTML documentation and a commercial case tool is used to produce the UML model, that is then exported to HTML pages.
- The GTC has preferred to base the whole documentation system on a commercial case tool. The whole documentation is kept inside the UML model provided by the tool.

The first approach is more expensive in terms of initial setup, since it requires selecting a pool of tools to produce the HTML pages and integrating them in a coherent development environment.

On the other hand, navigating through HTML documents is typically easier and more intuitive, in particular for people not expert in the usage of case tools, like project reviewers.

3 USE CASE DRIVEN PROCESS

3.1 The Process in Brief

The Unified SW Development Process[4] is an iterative and incremental Process. The process is very well described by Figure 1, from [4].

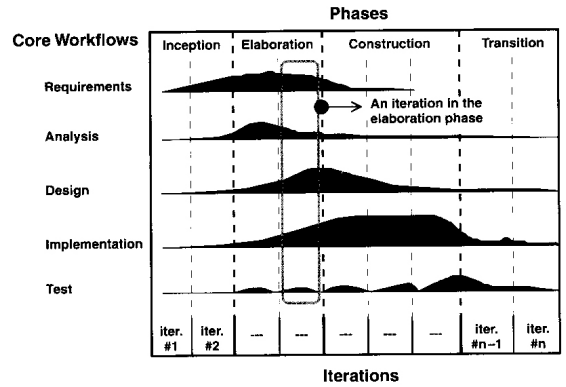

Figure 1 - Unified SW Development Process

The work necessary to build the final system is divided into smaller slices. The driving concept consists in the idea that every slice must be a mini-project in itself, which has well defined goals and which can be measured and controlled. Each mini-project is an **iteration** in the development, since it performs a complete workflow.

If the iteration meets its goals, the system is released and the development proceeds with the next iteration.

The whole life cycle of a project, that culminate with a major product release, is divided in four **phases** that group together a number of iterations with the main focus on the same worksflows:

- Inception
- Elaboration
- Construction
- Transition

3.2 Use Cases Drive the Process

The whole development process follows a flow that is driven by Use Cases[1] [2]. At the same time, Use Cases mature and evolve during the life cycle of the project.

Actually, the purpose of each iteration is the realisation of a set of Use Cases. These Use Cases are first identified and specified. Then a design is created according to the chosen architecture for the whole system. The development team implements the design and verifies that the system satisfies the Use Cases.

- In the *Inception Phase*, high level Use Cases are developed to identify what is in the scope of the project and to capture the basic requirements.
- In the *Elaboration Phase*, more detailed Use Cases contribute to the baseline architecture and to the risk analysis.
- In the *Construction Phase*, Use Cases will be used as the starting point for detailed design and for developing test plans.
- In the *Transition Phase*, Use Cases are the core items of the Acceptance Test. They are also used to develop user guides and for training.

In our experience, it is very effective to write Use Cases as structured text using a formal template. The usage of plain text facilitates the understanding of the Use Case by people with different background. A formal template forces a recognisable and common layout; relevant information is easier to find and communication is optimised.

The template we have defined is largely based on [2] and [7].

4 PROJECT MILESTONES

Every phase is closed with an official project milestone.

Such milestones are extremely important, since they give to managers and customers the possibility of making decisions before work can proceed to the next phase and to monitor the progress of the work.

These milestones are fixed points in the planning and their existence helps the development team in focusing the work toward specific dates.

For this reason we consider essential that milestones are not moved in case of delays in the project, but rather the scope of the milestone is clearly and officially restricted.

We consider very important to define our milestones to match the standards used by the other teams involved in the project and to match the naming commonly used by the external reviewers, like the once defined in IEEE[5] or similar Project Management Plans.

We have anyway verified that the milestones defined in the Unified Process[4] map quite well with the standards used in both our projects.

With respect to the Unified Process, we have in particular added the Software Requirements Review very early in the process, just at the end of the first Iteration in the Inception Phase. We consider it very important for a good start of a project, since it provides the opportunity for an official clarification on the basic characteristics of the system using the semi-formal representation of Use Cases.

5 PRINTABLE DOCUMENTS

We are well aware that online documentation alone is not enough. Printed documents are essential for reviewers and for members of the development team.

Online documentation is very good to:

- Navigate through relationships
- Quickly look for information with a search engine.
- Use it as an up to date reference.

On the other hand,

- A big amount of information can be read conveniently only on paper
- Most people like to keep printed reference documents available at hand
- Many of us do some work offline, not sitting in front of our computer screens.

In order to make available both online and printed documentation, two main problems must be taken into account and solved:

- Information consistency
- Differences in the printed and online media

The GTC group bases the printable documents on the reporting functions provided by the UML Case Tools.

The ESO group extracts the printable documents directly from the HTML documentation. An automatic update procedure has to take care of updating transparently all the places where the information is used.

It is important to note that Web and printed paper are very different media. Just printing a Web tree is typically bad. Information must be kept as atomic as possible. The elementary information units can then be used as building blocks and inserted in very different documents.

6 CONCLUSION

The Unified SW Development Process[4] is meant to be tuned and to be adapted to the specific needs of a project.

Control systems of scientific experiments are very different from other software projects and in particular from commercial application software.

Most of the books focus on commercial applications, since they cover a larger fraction of the software development market.

Here are some important aspects to keep in mind:

- A scientific experiment is always at the limits of knowledge and technology. Requirements are unclear and change in the course of the project.
- The software to be developed is highly linked to the hardware and to the electronics. For example software deadlines actually depend and are affected by hardware delivery dates.
- The system is unique. There will be just one or very few installations. We cannot count on thousands of beta testers to debug the software.
- When it comes to final system integration, the control software is used also to validate hardware and electronic performances. The software team responsible for the final integration becomes automatically also responsible for the verification of the whole system.
- Interfaces with hardware and electronics have a big influence on the system and we have to spend a big effort in making them clear and stable.

Regular discussions among independent teams help in identifying which aspects of the SW Development Process are more important for every specific project.

7 ACKNOWLEDGEMENTS

This work is the result of many hours of discussions, trials and test inside our groups. We want to thank here all our colleagues for the important contribution to the definition of the SW Development Process for our projects.

REFERENCES

[1] I.Jacobson, "Object-Oriented Software Engineering, a Use Case Driven Approach", Addison Wesley, 1992

[2] G.Schneider J.P.Winters I.Jacobson, "Applying Use Cases: A Practical Guide", Addison Wesley, 1998

[3] G.Booch J.Rumbaugh I.Jacobson, “The Unified Modelling Language User Guide”, Addison Wesley, 1998.

[4] I.Jacobson G.Booch J.Rumbaugh, “The Unified Software Development Process”, Addison Wesley, 1999.

[5] IEEE 1058.1, 1987 - "Standard for Software Project Management Plans"

[6] G.Chiozzi P.Duhoux R.Karban, "ATCS System Design Description", VLT-SPE-ESO-15151-1795, ESO, 1999

[7] L.Mattingly H.Rao, Writing Effective Use Cases and Introducing Collaboration Cases, JOOP, Oct. 1998