Proc of Int. Conference on Product Focussed Software Process Improvement
PROFES' 99, Oulu, Finland, ISBN 951-38-5271-7, 135-149.

Developing a Change Request Management Tool for a Distributed Environment

Horst Lichter
Department of Computer Science
Aachen University of Technology
D-52056 Aachen
lichter@informatik.rwth-aachen.de

Manfred Zeller
ABB Utility Automation GmbH
P.O. Box 10 03 51
D-68128 Mannheim
manfred.zeller@dekwl.mail.abb.com

Abstract

This paper presents the experience we obtained in a project aiming at developing and introducing a tool supported systematic approach of change request management. First, we briefly present product and software development at ABB Utility Automation GmbH - Fossil Power Plants and its effort to continuously improve its software engineering capabilities. Then we describe the tool developed to support change management by presenting the basic requirements, its overall architecture, the workflow defined to process change requests and some aspects of the user interface and of tool administration. Afterwards, we discuss the experience we obtained during the project grouped in two categories. The paper closes by presenting our future plans.

Key words: software process improvement, change request management, development tools, workflow, Lotus Notes.

1 Background and Motivation

ABB Utility Automation GmbH - Fossil Power Plants (UTA/F for short) is a company of the worldwide ABB group, employing 800 people, 70 of them in research and development (R&D). The project presented here is focused on the R&D organization. Their business is developing control and monitoring systems for fossil power plants. The products range from controllers to operator stations and engineering systems. Software is a major part of these products. Regarding software development projects two different project types can be distinguished: projects developing basic functionality and components and projects adapting these components according to specific needs of customers. Due to UTA/Fs organization product and software development is distributed over eight sites in five different countries.

Having recognized the impact of good software engineering on software quality as well as on software costs, ABB Kraftwerksleittechnik GmbH, the predecessor organization of UTA/F, started in 1991 an initiative to improve its software process maturity. This process has been continued by UTA/F.

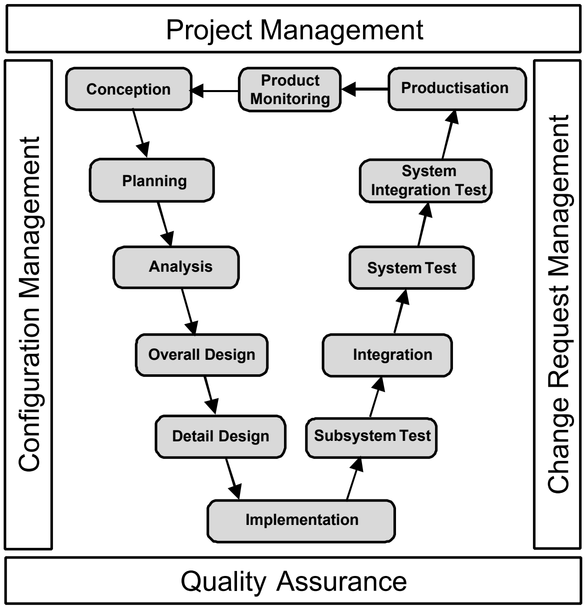


Figure 1: Software process model

A major and important result of its continuous software process improvement initiative is a software process model which takes into account all major aspects of software projects: e.g. organization, planning, realization and control. The quality assurance is integrated in this process model as well. This model is embedded in UTA/Fs overall product life model. The software process model is based on the traditional phased V-like model described e.g. in Bröhl (1995) and defines the development phases as shown in figure 1.The process model is applied by every development project and is tailored to the specific needs of individual projects. More information on

UTA/Fs improvement activities can be found in Lichter (1995) and Welsch (1997).

In this paper we focus on presenting the experience made during the last three years regarding the development and usage of a change request management tool (CRM tool for short). Systematic change request management has been encountered as an important factor to assess product and process quality as well as monitoring quality assurance and maintenance costs. Therefore UTA/F has developed in 1993 a tool supporting change request management activities. This tool was built as a database application using a relational database management system. It offers standard forms at the user interface to enter change requests and to get information about the database content. After having used it for two years the following pitfalls have been encountered:

- Since the system did not support the distributed usage of the CRM database only a small part of the organization (located in Mannheim, Germany) had access to it. Hence the system could not directly be used by engineering organizations of other locations or at plant construction sites.
- As a consequence paper and electronic versions of change requests (CRs) existed in parallel. This lead to several misunderstandings as well as to a CR database content that was never up to date. Hence statistics regarding CR processing did not correspond to reality.
- Since procedure and workflow were hard-coded the tool was rather inflexible. Improvements and adaptations could not be implemented easily. People involved in the CR workflow could not be informed of taking over a CR for further processing via mail.
- The acceptance of the system was low, because it was not available on every workstation and the processing of CRs could not be controlled and monitored by the non-local submitters of CRs.

Based on these findings and due to the meanwhile distributed R&D organization ABB UTA/F launched a project that aimed at introducing a systematic and tool supported CR management that overcomes the problems mentioned before. In the following, we present the main experience and findings of this project. First we give an overview on the most central requirements and describe the overall architecture of the tool together with the CR workflow. Then we present some aspects of the user interface and of tool administration. Finally, we summarize our experience gained so far. In the last section we briefly describe some activities to enhance the current CRM tool that we plan for the future.

2 Essential Requirements

We expected the CRM tool to support the management of detected problems of all UTA/F products. This includes products developed by UTA/F or by a subcontractor as well as involved third party products. Problems may be detected by the R&D organization itself or by any other organization dealing with UTA/F products (test, training, engineering, sales, service, etc.). Therefore the most significant requirements on the CRM tool were:

- worldwide availability within ABB at all UTA/F R&D organizations and at all organizations dealing with UTA/F products
- workflow-controlled processing of CRs
- monitoring the current working state and the history of each CR
- generating CR statistics and reports
- selecting CRs according to online definable criteria
- archiving of fixed CRs
- easy and save worldwide tool administration
- flexibility regarding layout of workflow and user interface
- flexible allocation of persons to the roles defined in the CR workflow
- updating change requests after modification of administration data

After having evaluated some commercial tools supporting CR management we had to realize that no single tool did satisfy our expectations. Either they did not fulfill all of our essential functional requirements or they were not available on ABB's strategic platform. Looking for an individual solution we came into contact with another ABB organization that had already developed a similar application based on Lotus Notes. Due to its experience and because Lotus Notes is a strategic ABB platform we decided to develop an own Lotus Notes based CRM tool. Restricted by the features of Lotus Notes 3.x and by the poor performance of 486er PCs we could not implement all requirements in the first release. Due to the enhancements of Lotus Notes coming with versions 4.0 to 4.5 and being in the meanwhile equipped with powerful Pentium PCs, we were able to gradually implement all our requirements in the succeeding releases of our CRM tool.

3 The Overall Architecture

The central components of the CRM tool are two Lotus Notes databases: the *workflow database* and the *archive database* (see figure 2). The workflow database contains all CRs, the administration data, the request identification numbers and the user's guide. Two types of CRs are distinguished:

- *Public change requests* are CRs on already released products. They are visible to all CRM tool users at any location.
- *Private change requests* are CRs on products being under development. Developers can only create them.

The archive database is used to store closed public CRs in order to relieve the workflow database. Archived CRs are visible to all CRM tool users but can not be changed.

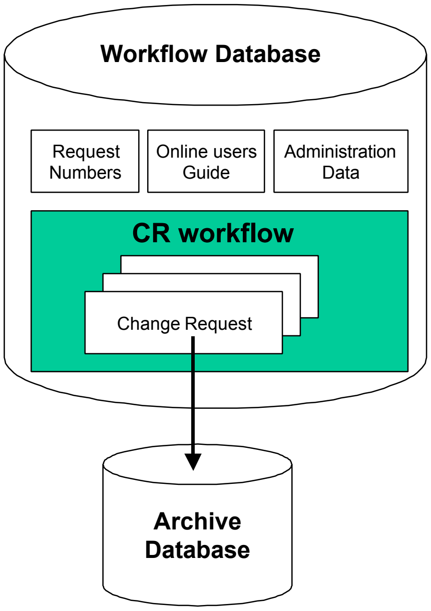


Figure 2: Architecture of the CRM tool

4 The CR Workflow

In the following we explain the processing of public CRs. Private CRs are processed in a much simpler manner.

Public CRs are processed in a controlled workflow. The workflow consists of ten different workflow states as depicted in figure 3. A specific role is assigned to each workflow state. A person assigned to the role of the current workflow state is called *work agent* of the CR. Only a work agent can process the CR and forward it to a succeeding workflow state.

Anyone detecting a problem on a product creates a change request using the CRM tool and submits it to the development organization responsible for the product. Then the CR gets the initial state *issued*. Its state changes to *in analysis* as soon as the problem analysis is initiated by the development organization. It remains in this state until the decision is made how to handle the problem. A CR may be *returned* to its submitter, if more information is required for analysis or if the CR is obsolete and should be *deleted* (the submitter only can delete a CR). In case the CR is accepted a developer is responsible to eliminate the problem. During this period the CR keeps the status *in work*. During the succeeding final test the CR is marked by the state *in final test*. The state *fixed* indicates that the final test has been successfully finished. After a non-successful final test the CR is reset into the state *in work*.
In case a non-critical CR has been accepted by the development organization but no resources (manpower, budget) are currently available it may be put into the state *postponed* for later implementation. A not accepted change request is put into the status *rejected*, e.g. if the reported problem is unfounded or if a workaround for the problem exists. Rejected and fixed change requests are transferred into the archive database after a predefined expiration time.

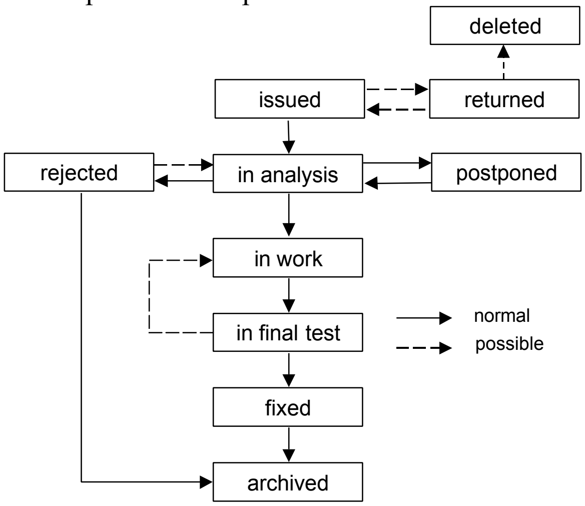


Figure 3: The CR workflow

Within the CR workflow there are some situations where people involved in the CR management have to be notified. Examples are:

- When forwarding a CR to a succeeding workflow state the person getting the new work agent is automatically informed by a notification.

- The issuer of a CR automatically gets a notification if the CR leaves the workflow i.e. the CR is postponed, fixed or rejected.
- The development project manager is automatically notified, if the decision authority does not start the problem analysis within a given time after issue.

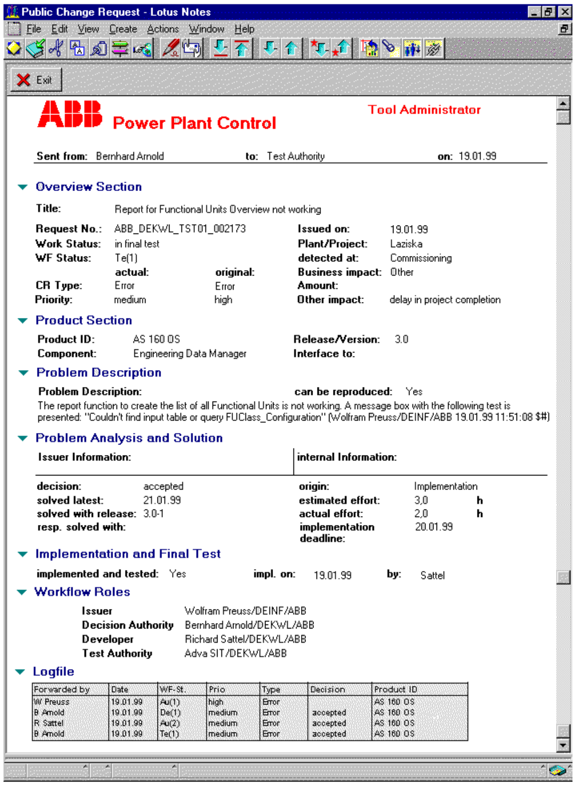
The CRM tool supports these notifications by automatically generating and sending emails to the corresponding persons.

5 User Interface

The user interface of the CRM tool is build by means of forms composed of standard Lotus Notes user interface widgets like text fields, buttons, combo boxes etc.

The layout of the user interface is identical for public, private and archived CRs. The user interface is divided into a number of sections (see figure 4):

- *Overview*: it contains e.g. CR No., title, CR type and priority.
- *Product Identification*: it is identified by product name, release and product component.
- *Problem Description*: this field allows a detailed description of the detected problem. Additional pages (e.g. screen dumps) may be attached.
- *Problem analysis and result*: this section documents the result of R&D's problem analysis as well as decisions that were reached.
- *Workaround and actions taken*: a workaround, if available is documented as well as the actions taken by the individual roles involved in the workflow process.
- *Implementation and test result*: the result of the implementation and of the final test by an independent test group is used for the identification of the succeeding workflow state.
- *Workflow roles*: The persons casting the individual roles within the workflow are indicated. The CRM tool automatically fills in the issuer and the product specific roles like test authority, development project manager and product manager
- *Logfile*: it presents the complete history of the CR. Each step in the workflow is documented with date, forwarding person, etc.


Public Change Request - Lotus Notes
File Edit View Create Actions Window Help
Exit
ABB
Power Plant Control
Tool Administrator

Sent from: Bernhard Arnold **to:** Test Authority **on:** 19.01.99

Overview Section

Title: Report for Functional Units Overview not working

Request No.:	ABB_DEKWL_TST01_002173		**Issued on:**	19.01.99
Work Status:	in final test		**Plant/Project:**	Laziska
WF Status:	Te(1)		**detected at:**	Commissioning
	actual:	**original:**	**Business impact:**	Other
CR Type:	Error	Error	**Amount:**	
Priority:	medium	high	**Other impact:**	delay in project completion

Product Section

Product ID: AS 160 OS **Release/Version:** 3.0
Component: Engineering Data Manager **Interface to:**

Problem Description

Problem Description: **can be reproduced:** Yes
The report function to create the list of all Functional Units is not working. A message box with the following test is presented: "Couldn't find input table or query FUClass_Configuration" (Wolfram Preuss/DEINF/ABB 19.01.99 11:51:08 $#)

Problem Analysis and Solution

Issuer Information: | **internal Information:**

decision:	accepted	**origin:**	Implementation	
solved latest:	21.01.99	**estimated effort:**	3,0	**h**
solved with release:	3.0-1	**actual effort:**	2,0	**h**
resp. solved with:		**implementation deadline:**	20.01.99	

Implementation and Final Test

implemented and tested: Yes **impl. on:** 19.01.99 **by:** Sattel

Workflow Roles

Issuer	Wolfram Preuss/DEINF/ABB
Decision Authority	Bernhard Arnold/DEKWL/ABB
Developer	Richard Sattel/DEKWL/ABB
Test Authority	Adva SIT/DEKWL/ABB

Logfile

Forwarded by	Date	WF-St.	Prio	Type	Decision	Product ID
W Preuss	19.01.99	Au(1)	high	Error		AS 160 OS
B Arnold	19.01.99	De(1)	medium	Error	accepted	AS 160 OS
R Sattel	19.01.99	Au(2)	medium	Error	accepted	AS 160 OS
B Arnold	19.01.99	Te(1)	medium	Error	accepted	AS 160 OS

Figure 4: User Interface

Three field types are distinguished in the user interface: mandatory, optional and computed fields. The type of each field is indicated by the color of the field title. Depending on the value of mandatory fields some optional fields may become mandatory too (e.g. if the value of field *decision* = accepted the optional field *solved with release* becomes mandatory).

Because Lotus Notes does not offer features to create and format statistics we were forced to develop two MS Excel based applications to generate CR statistics and reports. A predefined set of selection criteria can be used to limit statistics or reports to a subset of all existing CRs (e.g. a report may be restricted on CRs of one specific product).
The statistics application generates time related and product related charts and tables with the workflow state as parameter. Standard diagram types are presented that can be changed by the user however.
The report application generates very compact reports only containing the most important data for e.g. analysis and decision meetings.

6 Administration of the CRM Tool

One major requirement was adaptability as well as good support of tool administration. In the following we list some features of the CRM tool regarding these aspects.

Adaptability of workflow and user interface
The workflow is defined by workflow states. The work agent roles and the possible successor workflow states are assigned to each workflow state. Hence, the workflow can easily be adapted by changing these assignments if necessary.
The user interface consists of fields. Each field is defined by a field name, its position in the layout and its attributes. The layout of the user interface can be adapted to new requirements caused e.g. by workflow modifications by changing the affected fields and their attributes.

Update of administration data
Changes of administration data (e.g. keywords, workflow definition, field attributes etc.) normally affect existing CRs. To automatically update the workflow and archive database appropriate macros are available ensuring the consistency of the databases.

Distributed Tool administration
Any ABB company worldwide can be connected to the change request management process by installation of the CRM tool on their local Lotus Notes server. Distributed tool administration facilitates the worldwide operation and maintenance of the CRM tool. It is organized through a two level tool administration concept: The global tool administrator normally residing at the main location of the R&D organization manages tool development and maintenance, tool configuration and worldwide tool installation. He also maintains the global administration data. Each local tool administrator manages the local administration data regarding the ABB company he is responsible for.

7 Experience and Lessons Learnt

In this section we present our experience and the lessons we have learned. We summarize the experience regarding tool development as well as usage of the tool in our company.

But first we list some figures giving a better impression on the tool usage.

- Currently the CRM tool is installed on eight Lotus Notes servers in five countries.
- The database contains 2900 CRs (1750 public CRs and 1150 private CRs).
- The size of the database is about 54 MB (18 Kbytes/CR in average)
- The database is replicated between the servers every four hours.
- Currently about 800 users are connected to the CRM tool, 220 users have issuer rights.

7.1 Tool Development

We have chosen Lotus Notes as development platform, because we expected to have many advantages in development, maintenance and administration of a distributed CRM tool. On the other side we knew that there were some constraints of Lotus Notes that we had to deal with.

Restricted by constraints of Lotus Notes 3 and by the poor performance of the target machines, we first developed a CRM tool only implementing some of our essential requirements. In order to get an application with acceptable performance, we decided to implement for each workflow state a corresponding input form with specifically defined access rights on individual fields. This design results in a number of hard-coded forms with poor flexibility. Adaptations regarding workflow or form layout as well as error correction were very costly.

After being equipped with fast Pentium PCs and having Lotus Notus 4.x available we could redesign our application to improve its flexibility and maintainability. This was mainly reached by using only one input form, by centralizing the administration data and by online computing access rights and specific attribute values of input fields. Furthermore we could implement step by step all requirements without too much loss of performance.

There are two major disadvantages of our implemented solution coming from Lotus Notes constraints:

- All data must be part of each CR (viewed as a Lotus Notes document) in order to select and present CRs in views categorized and sorted by arbitrary field value. Splitting up the data and using shared data by referencing to other Lotus Notes documents can not be used. This results in heavy weight CR documents.
- After changing product or workflow specific administration data all affected CRs must be updated. Additional development and test effort was necessary to develop adequate update macros.

In summary we don't regret having chosen Lotus Notes as development platform especially because ABB regards Lotus Notes as one of its strategic platforms. Therefore Lotus Notes is installed at each ABB company and its administration has been established worldwide. This facilitates new installations of the CRM tool. The replication of the CR databases is ensured by the existing Lotus Notes infrastructure resulting in a simple procedure to upgrade all locations when a new release of the CRM tool is available: a new release of the tool is installed at the main location by the tool administrator. After one replication cycle it is available at all other locations.

7.2 Tool Usage

In order to systematically introduce the new CRM tool several courses were organized aiming at presenting the central concepts of CR management as well as the main features of the new tool.

In the beginning there was some skepticism on the usefulness of the concepts and the tool support. Particularly people argued against the visibility of public CRs. Since public CRs are visible worldwide, the usage of the CRM tool leads to an overall transparency concerning product related CRs. This situation was pretty new for most people and they needed time to accustom to this situation. In the meanwhile this is accepted and people now regard the worldwide visibility of public CRs as a main advantage of the new CR management process.

At the beginning private CRs were not used frequently. Although private CRs are only visible within a development team and their processing is not controlled by workflow, developers did not see benefits in issuing private CRs. In the meanwhile private CRs are used more and more, mainly to describe current problems or to describe new ideas and future product features. Hence, private CRs are currently also used to collect requirements for subsequent product releases.

Another aspect we like to mention here regards CR management and its results as a reasonable basis for overall product and project management decisions. Due to systematically collecting and processing CRs we now have data available that can be used to assess product quality as well as some aspects of the development process quality. For example we have realized that the time elapsed between issuing a CR and its analysis has come down from several weeks to about two weeks in average. Currently we do not systematically evaluate all the data, but we plan to do more data analysis in future.

Besides the usefulness of the concepts of CR management the degree of usability of the CRM tool was an important point concerning its acceptance by the users. Since different user groups, e.g. product managers, department managers or developers are using the tool it has to be customizable to the specific needs of these groups. The feature of Lotus Notes to define private views on a database is used to customize the user interface of the tool. Hence, a user can work with predefined selection facilities of the tool or can define private views presenting exactly those information he is interested in (e.g. product specific CRs, workflow state specific CRs, CRs sorted by work agent etc.). Because Lotus Notes is a strategic platform within ABB most CRM tool users had experience in creating private views and using Lotus Notes applications. This facilities the introduction and the acceptance of our tool.

8 Conclusions and Outlook

Before we present some ideas and features we plan to realize in future versions of the CRM tool we would like to summarize our discussion by comparing the pros and cons of our approach for systematically managing CRs.

We regard the following aspects as the main benefits of our solution:

- Due to the underlying features of Lotus Notes we were easily able to set up a CRM tool that runs in a highly distributed environment. This supports distributed product development as well as distributed product responsibility.
- As a consequence, a CR now can be issued at any location where a problem is detected.
- CRs are forwarded fast and secure to the location responsible for handling the CRs.
- Because CR processing is under control of a defined workflow, we are able to monitor CR processing. This makes it easy to detect those CRs that stay unacceptable long in a certain workflow state.
- Because the user interface only allows a predefined set of terms to characterize a CR, all CRs are describe in a uniform manner (facilitating selection and grouping of CRs).
- Often new requirements could be implemented quickly because both the user interface and the CR workflow were designed

to be adaptable.

Of course, there are weak aspects too. We see the following ones:

- The degree of adaptability of the user interface that we have implemented leads to a slower performance of the tool when it starts up.
- Because Lotus Notes does not offer typical database mechanisms like locking, we had to define administrational workarounds to prohibit parallel write access on CRs.
- Lotus Notes does not completely offer the standard Windows interface widgets. Hence, the user interface does not have the standard Windows look and feel.

The experience obtained in the project described in this paper was prevailing positive. We were successful in both, developing a useable workflow based CRM tool and in introducing it in a distributed environment. Nevertheless, there is a lot to do to get all potential benefits from systematically collecting and processing CRs. In future we plan to work on the following topics:

- Integrating our report and statistics application implemented by means of MS Excel in the Lotus Notes based CRM tool (e.g. by an OLE interface).
- Defining and implementing an interface to the configuration and version management tool used by the developers. This is a prerequisite to combine CR processing data and corresponding version management data.
- As mentioned before we plan to systematically evaluate and assess the data concerning CR processing. This needs a deeper definition of the metrics we want to apply.
- Last but not least we like to make the CRM tool accessible by internet.

References

Bröhl, A.-P., W. Dröschel (1995)*: Das V-Modell-Der Standard in der Software-Entwicklung mit Praxisleitfaden*, Oldenbourg Verlag.

C. Welsch, H. Lichter (1997): *Software Process Improvement at ABB – Commonn Issues and Lessons Learnt,* Proceedings of Software Quality Management SQM 97, Bath UK.

Lichter, H, C. Welsch, M. Zeller (1995): *Software Process Improvement at ABB Kraftwerks-leittechnik GmbH*, In P.Elzer, R.Richter (eds.) Proceedings of MSP'95 Experiences with the Management of Software Projects, Karlsruhe, IFAC Conference Proceedings, Elsevier.