INCREMENTAL IMPROVEMENT OF THE SOFTWARE DEVELOPMENT PROCESS WITHIN THE DUTCH AGRICULTURAL RESEARCH DEPARTMENT

M.J. van der Velden
P.R.H. Hendriks
B. van der Wal

DLO Winand Staring Centre for Integrated Land, Soil and Water Research (SC-DLO), P.O. Box 125, 6700 AC Wageningen, The Netherlands, m.j.vandervelden@sc.dlo.nl

Software Engineering Research Centre (SERC), P.O. Box 424, 3500 AK Utrecht, The Netherlands, hendriks@serc.nl

BSO/Quality Management Competence Centre (BSO/QMCC), P.O. Box 8348, 3503 RH Utrecht, The Netherlands

ABSTRACT

In many research organisations software development is becoming one of the most important support activities. A lot of research organisations however, do not have structural means to control the quality of the software they produce. Therefore, the Dutch Winand Staring Centre for Integrated Land, Soil and Water Research started a project to improve its software engineering activities. The project can be characterized as a mixture of software engineering en organisational change. The results show that for sustainable improvement the development of the organisation is more important than achieving technical results.

Keywords: software process improvement, software engineering, organisational change

1. INTRODUCTION

In many research organisations software development is rapidly becoming one of the most important support activities. An increasing number of research projects depend on the use of software products which are produced by the researchers. Examples in the field of agricultural research are simulation programs and geographical

information systems. In contrast to the emphasis on the quality of other support activities such as statistical analysis and chemical experiments, most research organisations do not have structural means to control the quality of the software they produce.

To cope with the problems associated with the increasing importance of software, the Dutch Winand Staring Centre, Institute for Integrated Land, Soil and Water Research (SC-DLO), started a project to improve its software engineering activities. SC-DLO is part of the Dutch Agricultural Research Department. The project focused on both change management and software engineering. The project is done by SC-DLO together with the Dutch Software Engineering Research Centre - SERC and BSO/Quality Management Competence Centre (BSO/QMCC). BSO/QMCC is part of ORIGIN, a member of the Philips group of companies. This paper describes the structure of the project and analyses its achievements and problems.

2. SOFTWARE QUALITY AT SC-DLO

2.1 Trends

At the end of 1989 a group of researchers expressed their concern of the software engineering situation within SC-DLO (ICW, 1989). Their concern was based on recognition of the following trends:

- An increasing number of research projects depend on the use of simulation programs and geographical information systems.
- Apart from the information written down in reports and papers, expert knowledge of researchers becomes more and more available in software. This is true for knowledge based systems, but also for simulation programs. Software is therefore considered to be of strategic importance.
- In the past, new programs were developed for each new research project. But nowadays a growing number of these projects use existing programs (with some modifications). As a result, poor quality accumulates through each new version.
- A lot of customers do not only want a report, but they also want the programs that were used to produce the report. Therefore, programs must be much more 'user-friendly' and more robust than in the past.
- In the past, simulation programs were made using Fortran only. Today these programs are built using Fortran, relational database systems, graphical user interface systems, and numerical libraries. Additionally they are used on a variety of operating systems. It is not feasible for an agricultural scientist to follow the rapid development of software engineering technologies together with his/her own specialism.
- The organisations with the easiest to use and most flexible products will be able to maximise income of consultancy projects based on these programs.

From these observations one may conclude that the efficiency of research organisations depends (among other things) on the efficiency of the software development process. The quality of research output depends (among other things) on the quality of software products. Since an increasing number of programs is released to customers, they have an impact on the reputation of the organisations that produced them.

2.2 Software quality project

To cope with the trends mentioned in section 2.1 a project was started to increase the quality of software produced at SC-DLO. Since so many researchers are involved in software development, this software quality project was set-up with a strong organisational focus. A rather heavy projectorganisation with a steering committee, a projectgroup, a technical supportgroup, an interest group and an external reviewer was established. Also a great number of researchers were involved by means of pilot-projects. This software

quality project was started at the end of 1992 with an investigation of the actual software engineering situation. Customers were asked to fill in questionnaires about SC-DLO programs, source code was inspected and researchers were interviewed.

The investigation made clear that about 80 of the 300 employees produce software as part of there daily activities. There were no guidelines for software development, there was no formalised development process and most products had poor usability and poor design.

After the evaluation a number of guidelines were developed by members of the software quality project together with researchers, and a software quality manager was assigned (1993). The guidelines covered the topics documentation, file IO, configuration management en Fortran coding style. In 1994, 13 pilot projects used the guidelines to improve their software. Additionaly, guidelines were developed for project management and reviews. In 1995 a start was made with reviewing software products to measure whether or not they are developed according to the guidelines. A more detailed description of the project can be found in (Velden et al., 1995).

The evaluation of the activities so far lead to the following conclusions (Bogaart, 1995):

- Researchers needed a lot of support to actually implement the suggestions made in the guidelines;
- Guidelines related to 'normal' research activities (such as documentation) were more succesful, than guidelines for activities more related to software engineering (such as configuration management);
- There is substantial overlap in the software produced by researchers throughout the organisation.

These conclusions suggest that the activities so far were not enough to achieve a high level

2.3 CMM-assessment

The aim of most activities covered by the software quality project was the improvement of the software *product*. It became clear however, that in order to achieve sustainable success it was needed to improve the *process* of software development as well. It was also rather difficult to objectively measure the progress made by the software quality project and some researchers felt a need for benchmarking.

One of the most well known models for software process improvement is the Capability Maturity Model (CMM) developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (Paulk et al. 1993). This model can be used to 'measure' the capability of an organisation to control its software development process. Therefore, in the beginning of 1995, a CMM-project was started to evaluate the usefulness of CMM in the context of a research organisation and to get an idea about the maturity of SC-DLO *in* terms of CMM. The main activities during this evaluation were:

- Management meeting
- Filling in questionnaires by 10 SC-DLO employees
- Interviews with 10 SC-DLO employees
- Evaluation and determination of company profile in terms of CMM
- Recommendations
- Final report
- Presentation of final report to management

One of the results of the CMM-project was a company profile with a score on a five level maturity scale defined by CMM (level five is the most mature level). SC-DLO was rated to be at the beginning of level two. A more detailed company profile based on CMM is presented in (Velden et al., 1996).

Although the topics addressed by CMM are considered relevant to researchers, it became clear that organisational growth with respect to software development is very difficult to

achieve, because software development is too much interweaved with research activities. The main advise that resulted from the project was therefore to separate professional software development from research projects. That is, researchers are allowed to make prototypes themselves, but if their software will be used by customers (inside or outside the organisation) the prototypes should be upgraded by a specialised software development department.

3. LESSONS LEARNED

Start with product improvement
The software quality project started with activities to improve the software *products*. This turned out to be a good choice because time was needed to create awareness within the organisation for the need for *process* improvement. In an environment were people do not consider software development to be of importance, they are more easily motivated to improve their software products then to improve their software development process. If the project had started with process improvement activities, the researchers would probably not have been as co-operative as they are now.

Sustainable results can only be achieved by process improvement
Although product improvement is good to start with, the CMM-project made clear that without process improvement it will not be possible to get sustainable results. Without process improvement, product improvements will only be achieved incidentally.

Be aware of communication problems
With a project on software quality it is almost certain that there will be a lot of confusion about terminology. Software engineers do not speak the language of agricultural scientists (and vice versa). Software consultants are used to communicate with software managers, they are not always used to communicate with general managers or research managers. An open mind and a lot of patience is needed to overcome the problems.

It is difficult to get sustainable results from pilot-projects
Although it is quite common to use pilot-projects to introduce new technologies into an organisation, it can be concluded that sustainable results are not necessarily achieved. There is always the danger that pilot projects are to much isolated, and that the results are considered specific to the pilot-project. A lot of 'public relations' is needed when working with pilot projects.

Many software engineering topics are relevant in research organisations
At the beginning of the quality project a lot of researchers were sceptical about the usefulness of software engineering technologies as SC-DLO was not considered to be a software developing company. Based on the experiences at SC-DLO it can be concluded that most software engineering topics are equally important in research organisations as in software development companies.

The CMM model can be used within research organisations, but...
All researchers involved in the CMM-assessment agreed that the topics mentioned by CMM (by means of the questionnaires) were relevant to their daily software engineering activities. The main reasons not to apply CMM to research projects is because the model demands more knowledge of modern software engineering technology then can reasonably be asked from a researcher, and the implementation of the model demands a level of discipline that conflicts with the culture in many research organisations.

Your business may be unique, your software quality problems are not unique
In a sense, each organisation is unique. No two organisations have the same culture, produce the same products, use the same tools etcetera. Therefore, employees tend to think that their quality problems are unique too. Based on the experiences within SC-DLO one can conclude that, with respect

to software quality problems, there are more similarities than differences between organisations. There is no reason to ignore the many tools and techniques available in the field of information technology. With minor adjustments most techniques seem to work within research organisations as well. However, it is important to define a procedure to *implement* the techniques into your organisation. This procedure can vary considerable depending on the organisation.

Talk business, not quality

Many people involved with quality projects try to get commitment by talking about quality. What they should do is talk about the business. Topics that did not directly relate to research work (such as configuration management) were much harder to implement than topics with a more direct link to research work (such as documentation).

4. CONCLUSION AND FURTHER ACTIVITIES

Based on the experiences at SC-DLO it can be concluded that incremental improvement of the software engineering activities within research organisations is feasible. It is also clear that a lot of effort is needed to make researchers aware of the need for software quality improvement. Even then, since most researchers are not primarily interested in software development, a lot of effort is needed to actually improve the situation.

Although SC-DLO is currently working on the implementation of a separate software development department (as suggested by the CMM-project) it is clear that quality related activities will be needed in future both within this new group and within the research projects (were prototypes will be made). The software development group might use CMM to develop its internal organisation. The researchers will be trained in software development methods and the use of the guidelines.

A new important topic will be a metrics program. Product oriented metrics can be used to specify the quality of software products prior to their development. As such, they might be added to functional specifications of software products, providing a means to clearly define development projects between researchers and the software development department. Proces oriented metrics can be used for better understanding of the software development process. Hence, they are a basis for future enhancements of this process.

REFERENCES

Bogaart, E.G.J. (1995), *Evaluatie Pilotprojecten - Rapportage taak I van fase 3 van het project Software-kwaliteit van het DLO-Staring Centrum*, Intern Rapport SERC.

ICW (1989), *Op weg naar een samenhangend software-ontwikkelingsbeleid voor het Staring Centrum*, Kerngroep Automatisering, 30 mei 1989, ICW, Wageningen. (in Dutch)

Paulk, M.C., Curtis, B., Chrissis, M.B. & C.V. Weber (1993), *Capability Maturity Model for Software, Version 1.1*, Report No. CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh, USA.

Velden, M.J. van der, B. van der Wal, A. Symons en J. Vreke (1996), Experiences with the Capability Maturity Model in a research environment. In: *proceedings third international conference on sofware quality management*, Britisch Computer Society, Cambridge, April 1996. (*to be published*).

Velden, M.J. van der, P.R.H. Hendriks en A.J. Udink ten Cate (1995), Software process improvement in a research environment. In: *Software Quality Management* (M. Ross, C.A. Brebbia, G. Staples en J. Stapleton, Ed.),. Vol. 1, pp. 63-74, Computational Mechanics Publications, Southampton (UK).