A Case Study of Evaluating Configuration Management Practices with Goal-Oriented Measurement[1]

Gianpaolo Cugola[†], Alfonso Fuggetta[†°], Pierfrancesco Fusaro[‡], Christiane Gresse von Wangenheim[‡]
, Luigi Lavazza[†°], Serena Manca[§], Maria Rosaria Pagone[§], Günther Ruhe[‡], Roberto Soro[§]

† Politecnico di Milano - Dipartimento di Elettronica e Informazione[2]
‡ Fraunhofer Institute for Experimental Software Engineering[3]
° CEFRIEL[4]
§ SIA (Società Interbancaria per l'Automazione)[5]

Abstract

The paper describes the application of goal-oriented measurement for evaluating configuration management practices at Società Interbancaria per l'Automazione (SIA). SIA is in charge of running, developing, and maintaining the National Interbank Network of Italy. The results of a CMM-based process assessment [15] indicated that Configuration Management (CM) practice was one of the most promising areas for improvement. A project was initiated aimed at establishing an improved CM process supported by state-of-the-art tools and incorporating sound practices [6,11]. It was decided to apply the new process to one of the most important products of SIA, which deals with the development of a new generation of networking software. Goal-oriented measurement following the Goal/Question/Metric (GQM) approach was applied to monitor the establishment of the CM process. The paper describes the establishment and execution of the measurement program and reports about related product and process modeling. Different techniques for qualitative and quantitative analysis of experimental data were performed. Selected results and experiences are reported.

Keywords: Configuration Management, Goal-oriented Measurement, GQM, Process Assessment and Improvement, Maintenance, Data Analysis.

1. Introduction

SIA is a medium-sized enterprise whose mission is to provide interbank services to all Italian banks and financial institutions. SIA's products and services include network services, banking services, and financial market operations.

SIA is carrying out an ambitious process improvement program, that employs several modifications of the development environment from both the technical and organizational viewpoint. In particular, SIA obtained the support of the European Union for the project MIDAS [10]. The focus of the project is on the establishment of an effective Configuration Management (CM) process. This includes the definition of CM procedures and policies, the selection and customization of automated tools supporting CM activities, and the experimentation of the new CM process in a pilot project.

This paper concentrates on the evaluation of CM practices by explicit modeling of underlying processes and performing goal-oriented measurement following the Goal-Question-Metrics (GQM) paradigm [1]. Application of GQM in combination with a specific approach of modeling software processes including CM was studied in [2]. Software developed under CM by using the SCCS utility and a measurement-based assessment of quality and needed resources is described in [12].

---

[1] This work was partly supported by the ESSI Project # 21244 MIDAS (Measurable Improvement of Development, Deployment and Operation of Interbank Automation Software).

[2] Piazza L. Da Vinci 32, I-20133 Milan, Italy

[3] Sauerwiesen 6, D-67661 Kaiserslautern, Germany

[4] Via Fucini 2, I-20133 Milan, Italy

[5] Viale Certosa 218, I-20156 Milan, Italy

0-8186-8093-8/97 $10.00 © 1997 IEEE

This paper is organized as follows. A detailed description of the practical problem is given in Section 2. The formulation of explicit models of the underlying processes and products is described in Section 3. Application of GQM-based measurement is studied in Section 4. The related experimental results are reported in Section 5. Finally, summary and conclusions are given in Section 6.

2. Problem Statement and Industrial Environment

SIA's main products and services are:

- Network services:

  They are used by SIA and its customers to exchange and distribute information and financial data.
- Banking services:

  They are mainly related to interbank payment systems and fund transfers.
- Financial market operations:

  They are mainly related to support different financial markets such as government bonds, futures and options on government bonds. To produce the software needed to operate these services, SIA has adopted a mixed approach based on the internal development and on subcontracting independent software houses and IT providers.

The selected pilot project deals with one of the most important SIA products called NRO (New OSI Network). Various teams are involved in development of the different NRO components. For coordination of the different products and releases performed from the different units inside and outside SIA, common models of processes and products and uniformly adapted CM procedures and tools are needed. As part of SIA's overall improvement initiative, establishment of uniform CM procedures and tools was performed according to the following steps:

1. A model of the process and of the product was built. The model of the product was necessary to understand the nature of the items that should be put under control of CM. The model of the process was also needed because the CM process had to be tightly integrated within the existing development process (including technical and managerial activities) in order to result as seamless as possible.
2. CM procedures were implemented. This implied the selection of a CM tool and its customization according to the models developed during the preceding step.
3. CM procedures were deployed. Besides the rather trivial installation of the tool, this meant that training and support for the users had to be provided.
4. The pilot project was monitored and assessed. The management of SIA wanted to determine whether or not the established CM procedures actually provided the benefits that had been anticipated. Therefore, a measurement-based assessment of the technology impact was carried out.

In particular, it was expected to provide models that would support SIA management in the prediction of some performances of the established process (e.g., effort and duration of macro-component development, number and distribution of anomalies for each software macro-component) and in taking decisions (e.g., how to plan the releases of software in time, how many anomaly corrections to associate with each release, etc.).

3. Product and Process Modeling

3.1. The Product

NRO represents the new generation of SIA's communication software. It is composed of three subproducts: MTP, SIG and EAS, which interact through the interbank Italian network. Each subproduct runs on many platforms and is composed of several executables that provide many functionalities such as message switching service, transaction support service, file transfer service, and network management. Each subproduct is composed of several macro-components, i.e., executable programs that can be combined in different configurations and that provide the network services. Each macro-component is in its turn made of elementary components.

Before modeling the activities which compose the software process, an entity-relationship description of the typical software product developed by SIA was derived [10]. This was particularly important, since SIA products are collections of related products that can be customized and organized into different configurations. The software product diagram is shown in Figure 1.

The product is developed according to the following life cycle model:

- Development;
- Internal test composed by acceptance test (AT), and integration test (IT);
- External test (ET);
- Production;
- Archive.

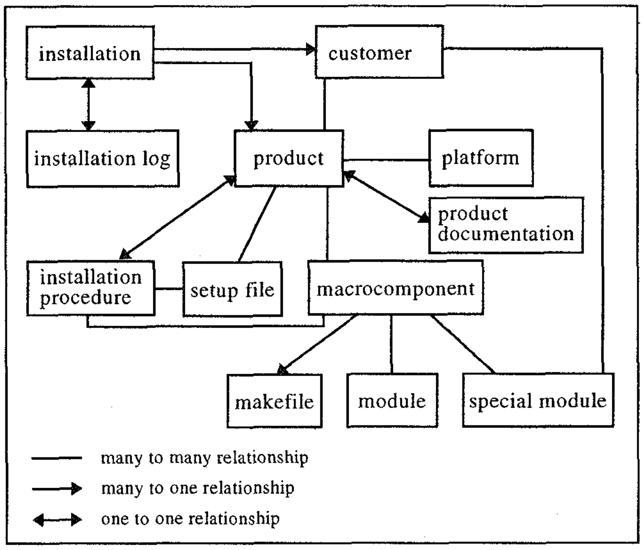

Figure 1. The Software product diagram

A software release can be archived in three different moments:

- when it enters internal test (it is called internal release);
- when it enters external test (it is called external release);
- when it enters the production phase, i.e., when it is delivered to users for deployment and usage (it is called production release).

It must be noticed that a release can be "stopped" at any test phase, thus being prevented from being released to final users. Such releases simply provide feedback to developers, that can use the results of tests to improve the following releases. This has implications on the measures: for instance, when measuring some features of production releases, the anomalies, development effort, number of changed modules, etc. must be computed taking into account the preceding internal and external releases that did not reach production, but indirectly contributed to the production release being considered.

3.2. The Process

The actual process modeling was carried out in a top-down fashion using the IDEF0 notation [9]. This modeling language is sufficiently intuitive and expressive, though its semantics is not formally defined. The model derived is a trade-off between the faithful representation of the actual development process and the definition of an "ideal" process, equipped with CM. The resulting model was similar to the current practice. Therefore, the effort to implement the most important features of the ideal process, especially as far as CM is concerned, was reasonable. A complete description of the SIA CM process can be found in [4,10].

The model of the process was used for several purposes:

- It allowed the people involved in the project (SIA management and consultants) to reach a common understanding of the process, the improvement requirements, and the practical constraints.
- In the formulation of goals, the process model was a constant reference, suggesting the product components and process activities to measure, highlighting variation factors, indicating the roles responsible for carrying out measures or validating them, and showing how the development process could have been enhanced with measurement activities.
- It was used as a reference in the interpretation of results (both collected data and feedback from developers).
- Finally, the model provided a sort of laboratory where to model process changes needed to improve the CM activities.

4. Application of GQM-Based Measurement

The measurement program defined and conducted at SIA followed the Goal-Question-Metric (GQM) approach [1,3]. GQM guides the definition of any kind of measurement goals and the subsequent refinement into questions and related metrics in a practical way. Main objective at SIA was an assessment of the software development process in dependence on the performance of configuration management. Besides the top-down refinement of the measurement, a mechanism is also defined for bottom-up interpretation of the collected data. Analysis and interpretation of the measurement is based on the characterization and understanding of the organizational context, environment and goals.

GQM has been successfully applied in several organizations [3,5]. A process model for GQM-based measurement was described in [7]. The basic steps of this model are shown in Figure 2. In what follows, the application of the different steps at SIA are described in more detail.

4.1. Characterization of the Environment

The definition of the measurement program started with a careful characterization of the organization context and the application project (NRO) to be measured. Some of the related results were presented in Section 2 and 3.

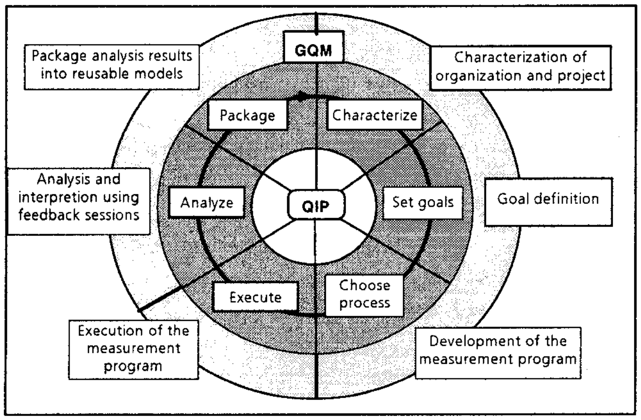

Figure 2. Phases of GQM-based measurement.

4.2. Definition of the Goals

The measurement goals were discussed during a brainstorming session attended by SIA management (including directors and mid-level managers), by GQM experts, and by the consultants that carried out the process assessment and the CM process definition. As a result of this plenary session, five goals focusing on effort, duration, and anomalies were set up. In order to limit the cost of the project, two of them were selected for execution within the measurement program.

The definition of the GQM goals was influenced by two considerations:

- SIA management was more interested in an absolute evaluation of the final situation than (as originally intended) in a comparison to the situation with the changed process. The main reasons for that was the difficulty in measuring the former situation due to the low maturity of the process (assessed to be at level 1 in the CMM scale). All data available were related to development procedures not well defined, unstable, and interpreted in different ways depending on the people involved and on the contingent situations. This clearly makes measurements unreliable and hard to compare to the corresponding metrics collected after the deployment of the new process.
- It was decided that the more interesting indicators of the process change effectiveness were the cost and duration of development and the quality of the result. It was also decided that effort, time, and problems should be tracked to their causes, including causes from CM.

The two final measurement goals are described according to the five-dimensions GQM template [1]:

Goal 1:	
Analyze	the maintenance process
for the purpose of	characterization
with respect to	effort and duration
from the points of view of	
	project managers and testers
in the context of	pilot project NRO

Goal 2:	
Analyze	the maintenance process
for the purpose of	characterization
with respect to	anomalies
from the points of view of	
	project managers and testers
in the context of	pilot project NRO

The selected goals deal with the maintenance of the product. This was a choice suggested by the contingent situation: the pilot project had already gone through the specification, design, implementation, and internal test phases, and was under external test, with frequent corrections and revisions. Note also that the assessment of the CM technology impact is not addressed directly, instead, the goals address the results of the performed development process with proper use of CM.

4.3. Development of the Measurement Program

Following the GQM paradigm [1], the selected goals were refined into questions related to the quality models for the measured properties, and to the factors influencing these quality models. In fact, in order to draw objective conclusions, an essential issue –often overlooked in other measurement approaches– is to take into account factors that influence the measures. All these questions were finally refined into metrics. The refinement was performed in two main steps: (i) knowledge acquisition from project team members and (ii) measurement planning, i.e., synchronization of the performance of the measurement program with the underlying development process.

During (i) knowledge acquisition, managers and testers from the NRO project were interviewed in order to make their knowledge concerning the measurement goal explicit. The information gathered by means of the interviews was later used to define a correct quality model including a set of variation factors (i.e., factors that are believed to affect the quality focus in the particular context) concerning the selected goals. This kind of information and all related hypotheses were coded into abstraction sheets [17] which give an aggregated view onto all essential information of the measurement program. Interviews increased the involvement of people in the measurement activity, showing the connection of metrics to developers' needs.

The factors that define the quality focuses of the goals derived from the interviews are:

- Effort per release of macro-component (distribution by life cycle phase and total).
- Duration per release of macro-component (distribution by life cycle phase and total duration).
- Anomalies generated by each release of macro-component.

Among the main factors that had impact on the above quality focuses were:

- Purpose of a release (i.e., anomalies fixing, new functionality, improvements).
- Size of a macro-component (i.e., LOC, new/reused LOC, modules, changed modules).
- Number of modified modules per macro-components;
- Number of modified LOC per module.
- Distribution of anomalies by cause, by detection phase, and by severity.

Some difficulties occurred during interviews when people involved in the interview were not actively involved in formulation of the measurement goal. A more careful and preliminary explanation of the measurement objectives and the overall approach would have probably made the subsequent activities easier.

During (ii) measurement planning, the measurement team documented the goals, refined them into questions and metrics, and defined the corresponding measurement procedures (i.e., how, by whom, and when the data had to be collected). Furthermore, they decided on quantitative and qualitative analysis to be used , and finally prepared the data collection media.

4.4. Execution of the Measurement Program

The measurement data was collected according to the procedures defined in the measurement plan. The measurement program for the assessment of the initial situation was restructured and reduced because of the unavailability of reliable data and in order to make the measurement process as seamless as possible.

The data collection activity regarding the situation after improved CM was supported by the CM tools, i.e., CCC Harvest on Unix and CCC LCM on main-frames. All data required by the measurement plan, except for the effort data, was a subset of the information stored in the CM repository. This data was automatically extracted through queries written specifically for this purpose.

4.5. Analysis and Interpretation

The results of the measurement program were discussed and interpreted by the project team from the data collected so far during organized meetings called feedback sessions [17]. Three consecutive sessions were performed. Results based on some formal quantitative and qualitative analysis were used as an essential input for performing interactive interpretation of measurement results during these sessions. Final results and interpretation are given in Section 5.

In order to adapt the measurement program to SIA needs and constraints, the measurement program was changed according to new insights (e.g., impossibility to collect rework effort data, data collected per release of macro-component instead of per subsystem, different aggregation of elementary data) gained during the feedback sessions.

4.6. Packaging

In order to reuse measurement results and experiences, the results of the measurement program have been packaged in such a way that future projects are able to reuse the results of the measurement program.

Many pieces of information that are now made widely and consistently available by the CM repository already existed before the introduction of CM, but were essentially the results of individual initiatives. As a consequence, they were heterogeneous, often available only within one group and incomplete (being tailored for the interests of the group). Before the introduction of CM the available data were also related to development procedures not well defined, unstable, and often adapted to personal preferences or needs. The lack of a systematic measurement program and, consequently, of historical data forced SIA management to take many important decisions only on the basis of their own experience and feelings.

5. Experimental Results

Data analysis provided valuable information concerning the quality and efficiency of the maintenance process in the NRO project. In particular, one principal result is the availability of reliable data concerning the measured quality aspects and their main influence factors. Based on this information, more detailed insights could be derived:

- The correction of an anomaly in a macro-component rarely involves more than two components and, in general, the number of affected modules is one or two per component. This is obviously a positive finding, since in the few cases where a modification spans more modules, this tends to require a bigger effort.

- Releases developed to correct anomalies do not introduce new anomalies (or a very small number of them), despite the number of LOC and modules changed.
- In some cases, the number of anomalies introduced into a release was greater than the number of changes. We found no clear explanation for this fact, but it can be observed that in 50% of such cases several (namely, more than 20) modules had been updated. Exploring the relationship between generated anomalies and modified modules, we found that
    - ◊ there are "clean" releases that did not introduce anomalies despite the number of changed modules (ranging from 2 to 18);
    - ◊ for the other releases, there is a clear correlation between the number of updated modules and the number of anomalies (2.5 to 5 anomalies every ten modules).
- Most releases (77%) grouped two or more changes of different kinds (e.g., adding new functionalities and correcting anomalies). In this type of releases, the number of modified modules varied from 1 to several tens. Considering the number of changes implemented in each release, it was found that the ratio '*number of modified modules / number of changes*' is greater than 4 in only 35% of the releases. The releases that incorporate just one change generally required changing an average of 9 modules. The conclusion that can be drawn is that the architecture of the system does not isolate the system's functionalities into separated modules very well, since on average a single change requires an update of several modules. However, grouping several changes into one release proves to be a good practice, since it allows the decrease of the average number of affected modules per change to 4.

A qualitative analysis was applied to supplement statistically based analysis. The objective was to generate rules describing and aggregating experimental results. These results were used as an essential input for performing interactive interpretation of measurement results during the last two feedback sessions performed at SIA.

We applied the Rough Set theory [13,14] to derive aggregated rules that have been used as foundation for integration of human based expert knowledge with the if-then-rules formally derived from experimental analysis data [8]. Rough Set based analysis (RSA) assumes ordinal or nominal scale of all the attributes. Discretization has to be done for all remaining cases. For this discretizations, experience and knowledge of SIA people involved in the measurement program was taken into account. In order to perform RSA, a software system called ProFIT developed at Poznan University of Technology [16] was used.

The rules obtained from RSA have been proven to be a suitable support for the analysis of measurement data and their interpretation in feedback session. Storage of aggregated and reliable rules in the experience base might be helpful for the planning of new projects and is part of the packaging step as mentioned in Section 4.6. In the context of performing goal-oriented measurement, the rules have been classified into different categories:

1. Rules which confirm the hypotheses: the rules state a formally correct confirmation of the hypotheses.
2. Rules which partially deviate to the hypotheses: the rules confirm the hypotheses after some minor modification.
3. Rules which contradict the hypotheses: the rules in this category contradict the formulated hypotheses.
4. Rules with no meaning: the rules in this category are formally correct but do not have any meaning to the goal definition.
5. Rules which are meaningful for the goal, but are not related to any hypothesis.

In Table 2, we summarize some selected rules which are based on the discretization described in Table 1: in particular, only rules belonging to categories 1 and 2 (i.e., confirming the hypotheses) are reported.

The different types of release are abbreviated as follows:

- New functionalities (NF)
- Enhancement (EN)
- Anomalies fixing (AN)
- Anomalies fixing and new functionalities (ANNF)
- Anomalies fixing and enhancements (ANEN)
- New functionalities and enhancements (NFEN)
- Anomalies fixing, enhancements and new functionalities (ANENNF).

For each rule the viewpoint team indicated in which category the rule belongs to. In Table 2, each individual rule is accompanied by its relative strength. While the absolute strength of a rule is defined as the number of releases supporting that rule., the relative strength is the ratio of its absolute strength and the number of objects belonging to the same concept as classified by that rule.

The quality of classification was evaluated for the case of anomaly prediction using leave-one-out-test. There were

- 56.5% of releases correctly classified;
- 17.4% of releases not classified;
- 26.1% of releases wrongly classified.

Metric	# interv.	Int1 (low)	Int2(medium)	Int3 (high)	Int4 (very high)
Number of new modules (N_new_modules)	4	0-10	11-31	32-93	94-299
Number of changed modules (N_changed_modules)	3	0-38	39-155	156-250	
Number of deleted modules (N_deleted_modules)	3	0-2	3-150	151-300	
Number of K lines of code (N_LOC)	3	19-46	46-190	190-1080	
Number of new lines of code (N_new_LOC)	3	0-2500	2501-9000	9000-12152	
Number of changed lines of code (N_changed_LOC)	4	0-100	101-1199	1200-2799	2800-5000
Number of deleted lines of code (N_deleted_LOC)	4	0-400	401-2200	2201-4000	4001-30000

Table 1. Discretization of attributes

Premise	Consequence	Relative Strength
(N_changed_Modules=high)	Anomalies=high	50.0%
(N_changed_LOC=low)	Anomalies=low	41.2%
(N_new_Modules=medium & N_new_LOC=medium)	Anomalies=low	40.0%
(Type of release=ANENNF) & (N_changed_Modules=low)	Anomalies=low	23.5%
(Anomaly_severity =high)	Duration=high	35.7%
(Anomalies=high)	Duration=high	28.6%
(N_changed_LOC=high)	Duration=high	28.6%
(N_new_LOC=medium)& (N_deleted_LOC=low) & (Anomalies=low)	Duration=low	28,6%
(Type of release=ANENNF) & (N_new_Modules=low) & (N_changed_LOC=medium)	Effort=very_high	37,5%

Table 2. Selected rules from RSA.

6. Summary and Conclusions

This was the first applications of goal-oriented measurement at SIA. The overall evaluation of this experiment is that the process worked fine. The total effort spent in performing the measurement program for the two selected goals was estimated to be about 1.5 person years This covers development of GQM-based quality models, formulation of measurement plans, setting up the measurement environment, data collection, data analysis and interpreting, and packaging of results. This effort was considered to be reasonable taking into account the dimension of the project and the organizational benefits obtained.

We have learned from this experiment that it is hard to assess the benefits obtained by the introduction of a well-defined and supported process in a development environment that was originally at CMM level 1. Another experience of the MIDAS project is that availability of a process and product models greatly facilitates the definition and execution of the measurement program.

The main results of the whole MIDAS project are:

- The understanding of the product has been improved. There is now a common view of the product, which is shared by managers, developers and operations management people.
- There is a correct perception of what is going on, based on objective parameters (e.g., number, state, properties, and history of anomalies), and on objective identification of items. Moreover, the people at SIA are progressively adopting a common language to express CM concepts.
- Further software development process improvement actions based on these understandings are formulated and initiated.

The measurement based evaluation of the CM process made explicit a series of detailed information which was previously incomplete, hidden or scattered throughout the process agents. In particular, the following issues have been clarified:

- Architecture of the product (subsystems, macro-components, components).
- Size of the product (number of components, number of modules, lines of code);

- Impact of the change (number of modified components, number of modified, deleted and new lines of code, number of modified, deleted and new modules);
- Availability and reliability of the product (release date, anomaly distribution by severity, by detection phase, and by cause);
- Release features;
- Responsibilities for the various process activities have been clarified.

The execution of goal-oriented measurement resulted in availability of essential, reliable data at a reasonable cost. It caused a real enthusiasm of SIA management towards this more problem oriented approach to the definition and exploitation of metrics. They fully understood the potential of process measurement to keep strategic aspects of SIA business (e.g., product and service quality) under control. Enhancements of the MIDAS measurement process have already been planned. In future measurement-based improvement programs, developers will be even more involved in directly accessing and analyzing data. One promising area of application is the optimization of sophisticated development tasks (e.g., to determine the correct amount of testing to be done on a given kind of anomaly).

Acknowledgments

The authors are indebted to the members of the SIA project team for their active cooperation and Susanne Hartkopf for performing Rough Set analysis. We also thank the referees for their useful comments.

7. References

[1] V. Basili, H.D. Rombach: The TAME Project: Towards Improvement-Oriented Software Environments, *IEEE Transactions on Software Engineering*, 14 (1988), p.758-773

[2] T.de Bunje et al.:Towards Measurable Process Models. *Software Process Technology, 5th European Workshop EWSPT'96, Nancy (France),* Springer 1996, 183-187

[3] L. Briand, C. Differding, H.D. Rombach: Practical Guidelines for Measurement-Based Process Improvement. *Software Improvement and Practice* 2(1997)

[4] G. Cugola, L. Lavazza, V. Nart, S. Manca, M.R. Pagone: An Experience in Setting-Up a Configuration Management Environment. *Software Technology And Engineering Practice - STEP'97 8th International Workshop (incorporating CASE'97)*, London, July 1997

[5] CEMP: Customized Establishment of Measurement Programs. ESSI Project 10358, Final Report, July 1996, available from http://www.iese.fhg.de /Services /Projects/Public-Projects/Cemp.html

[6] S. Dart: The Past, Present, and Future of Configuration Management. *Software Engineering Institute* Tech. Report CMU/SEI-92-TR-8 ESC-TR-92-8, July 92

[7] C. Gresse, B. Hoisl, and J. Wuest: A Process Model for GQM-based Measurement, Technical Report STTI-05-94-E, University of Kaiserslautern, 1995

[8] S. Hartkopf: Analysis in GQM-Based Measurement: Feedback Sessions with Rough Sets, Master Thesis, Univ. Kaiserslautern, Department of CS, April 1997

[9] Integration Definition for Function Modeling, *IDEF0.* Federal Information Processing Standards Publications, December 1993

[10] MIDAS: Measurable Improvement of Development, Deployment and Operation of Interbank Automation Software. ESSI-Project 21244, Final Report, September 1997

[11] A.K. Midha: Software Configuration Management for the 21st Century. *Bell Labs Technical Journal* 2(1997), 154-165

[12] R.J. Leach: Methods of Measuring Software Reuse for the Prediction of Maintenance Effort. *Journal of Software Maintenance: Research and Practice*, 8(1996), 309-320

[13] Z. Pawlak: Rough Sets: *Theoretical Aspects of Reasoning about Data.* Kluwer, 1991

[14] G. Ruhe: Rough Set based Data Analysis in Goal Oriented Software Measurement. *Third Symposium on Software Metrics, Berlin, March 1996*, IEEE Computer Society Press, p.10-19

[15] Software Engineering Institute: *The Capability Maturity Model.* Guidelines for Improving the Software Process, Addison-Wesley, 1995

[16] R. Susmaga and R. Slowinski: Rough Processing of Fuzzy Information Tables (ProFIT). Computing Science Inst., Poznan University of Technology, 1996

[17] F. van Latum, M. Oivo, B. Hoisl, G.Ruhe: No Improvement Without Feedback: Experiences From Goal-Oriented Measurement at Schlumberger. *Proc. Fifth European Workshop on Software Process Technology*, Nancy, France, Springer LNCS, 1149(1996), 167-182