Frank Bomarius Markku Oivo (Eds.)

Product Focused Software Process Improvement

Second International Conference, PROFES 2000
Oulu, Finland, June 20-22, 2000
Proceedings

Springer

[redacted]

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Frank Bomarius
Fraunhofer Institut, Experimental Software Engineering
Sauerwiesen 6, 67661 Kaiserslautern, Germany
E-mail: bomarius@iese.fhg.de

Markku Olivo
VTT Electronics
Kaitovayla 1, 90571 Oulu, Finland
E-mail: Markku.Oivo@vtt.fi

Cataloging-in-Publication Data

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Product focused software process improvement : second international conference ; proceedings / PROFES 2000, Oulu, Finland, June 20 - 22, 2000. Frank Bomarius ; Markku Oivo (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2000
(Lecture notes in computer science ; Vol. 1840)
ISBN 3-540-67688-0

CR Subject Classification (1998): D.2, K.6, K.4.2

ISSN 0302-9743
ISBN 3-540-67688-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a company in the BertelsmannSpringer publishing group.

Printed in Germany

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN 10718956 06/3142 5 4 3 2 1 0

:nmission, Fraunhofer
)ulu, and VTT Elec-
'ul to the authors for
reviewing and partic-
nittee, and numerous
'erence.

Frank Bomarius
Markku Oivo

Conference Organization

General Chair

Markku Oivo, VTT Electronics, Oulu (Finland)

Organizing Chair

Pasi Kuvaja, University of Oulu (Finland)

Program Co-chairs

Frank Bomarius, Fraunhofer Institut Experimentelles Software Engineering, Kaiserslautern (Germany)
Terry Rout, Software Quality Institute, Queensland (Australia)

Panel, Workshop & Tutorial Chair

Andreas Birk, Fraunhofer Institut Experimentelles Software Engineering, Kaiserslautern (Germany)

Industrial Chair

Seija Komi-Sirvio, VTT Electronics, Oulu (Finland)

Publicity Co-chairs

Petra Steffens, Fraunhofer Institut Experimentelles Software Engineering, Kaiserslautern (Germany)
Adriana Bicego, Etnoteam, Milano (Italy)

Program Committee

Adriana Bicego, Etnoteam (Italy)
Lionel Briand, Carleton University (Canada)
Richard Castanet, Université Bordeaux (France)
Reidar Conradi, NTNU (Norway)
Khaled El Emam, National Research Council (Canada)
Jacky Estublier, Centre National de la Recherche Scientifique (France)
Ilkka Haikala, Tampere University of Technology (Finland)

Bärbel Hörger, DaimlerChrysler (Germany)
Hajimu Iida, Nara Institute of Science & Technology (Japan)
Janne Järvinen, VTT Electronics (Finland)
Ross Jeffery, University of New South Wales (Australia)
Erik Johansson, Q-Labs (Sweden)
Kari Känsälä, Nokia Research Center (Finland)
Karlheinz Kautz, Copenhagen Business School (Denmark)
Marc Kellner, SEI, Carnegie Mellon University (USA)
Taghi M. Khoshgoftaar, Florida Atlantic University (USA)
Munish Khurana, Motorola (UK)
Graham King, Southampton Institute (UK)
Pasi Kuvaja, University of Oulu (Finland)
John Munson, University of Idaho (USA)
Paolo Nesi, University of Florence (Italy)
Risto Nevalainen, STTF (Finland)
Harri Reiman, Ericsson (Finland)
Günther Ruhe, Fraunhofer IESE (Germany)
Veikko Seppänen, VTT Electronics (Finland)
Forrest Shull, Fraunhofer Center Maryland (USA)
Reijo Sulonen, Helsinki University of Technology (Finland)
Ian Taylor, Process Research Consultants (Austria)
Rini van Solingen, Fraunhofer IESE (Germany)
Otto Vinter, Bruel & Kjaer, Denmark
Giuseppe Visaggio, University of Bari (Italy)
Liisa von Hellens, Griffith University (Australia)
Yingxu Wang, IVF (Sweden)
Claes Wohlin, Lund University (Sweden)

In addition, the following persons have helped in reviewing the papers:
Fabrizio Fioravanti, University of Florence (Italy)
Pierfrancesco Bellini, University of Florence (Italy)
Jens-Otto Larsen, NTNU (Norway)
Torgeir Dingsoyr, NTNU (Norway)
Jennifer Gasston, Griffith University (Australia)
Jo Orr, Griffith University (Australia)
Brent Cahill, Griffith University (Australia)

Last but not least many thanks to Patrick Leibbrand, Fraunhofer IESE (Germany) for copyediting this volume.

[redacted]

Table of Contents

[redacted]

Empirical Software Engineering

Industrial Experiences

Methods and Tools

Introducing the Data Role in Models for Database Assessment

Juan Garbajosa[1], Pedro P. Alarcon[1], Hector Garcia[1], Maria Alandes[1], and Mario Piattini[2]

[1] Technical University of Madrid. E.U. Informatica. Dpt. OEI. Ctra. ValenciaKm.7, E-28031 Madrid, Spain.
Tel. +34-913367879, Fax:+34-913367520
{jgs,hgarcia,pcavero}@eui.upm.es
[2] University of Castilla – La Mancha (UCLM). E.U. Informatica.
Ronda de Calatrava 5, E-13004 Ciudad Real, Spain.
Fax: +34-926295354
mpiattin@inf-cr.uclm.es

Abstract. Applications lay more and more on databases as long as technology provides new functionality. Databases present a characteristic that is not common to other software system components: they have contents. Data issue has been traditionally left on the end-user side. However, bad quality data may damage the application performance seriously. Even when this is simple to state a lot of problems remain behind. Assessment models do not always contemplate database development process characteristics. As we show, work products are not often those expected in a database development process. Furthermore studies on data quality are rather recent and are a not-yet-consolidated discipline. A strategy is presented to introduce some issues related to data quality in the base practices considering the proposed processes for a specific assessment model, SPICE. As we explain, these base practices and work products should be considered as extensions to the conventional assessment model, as it presents problems to support database development processes.

1 Introduction

Databases have specific characteristics that make them neatly different from other kind of software components that systems are made of and that are considered as more conventional. These differences are basically reflected in Database systems development products and processes.

An intrinsic database feature is persistent contents. These contents change along with the database life. Even with a database product of great quality and an excellent implementation, the system may perform poorly as long as the quality of the data is

F. Bomarius and M. Oivo (Eds.): PROFES 2000, LNCS 1840, pp. 48-58, 2000.

bad. Furthermore, data come up late in the database lifecycle. Actually data play a main role that comes up once the database becomes operational.

Some methodologies are starting to consider data quality/goodness issues during the development. Several recent publications [1,2,3,7,8,9,10] show the importance of the issue, even when different points of view exist. It is our strong believing that not to consider data during the development phase involve unexpected results in the performance of the system during the operation phase.

An in-house developed methodology, MEDITA [4], includes data quality/goodness issues as an activity to perform during the database development. MEDITA was originally developed for data warehouse development and then extended for databases in general, then called EMEDITA, though with special focus on relational databases.

As important as introducing practices is to assess them. For this we have studied the SPICE-ISO/IEC 15504 [5] process dimension from the database development point of view. In [6] we described extensively the findings of mapping a Generic database design process onto the SPICE-ISO/IEC 15504 process model. In it we just outlined how the goodness of data, described in terms of data quality, could influence the product quality and, therefore why practices oriented to consider this issue should be implemented and assessed.

Within this paper we explain in the depth what is our approach on how the data goodness check practices must be assessed, which base practices should be introduced and to which products they should be related.

As described in [6], we have found that there are three topics that deserve specific attention:

Work products
Base practices
A third topic, of a different class, is data quality/goodness

Work products and Base practices specific for database design find a reasonable placement within the SPICE-ISO/IEC 15504 model, as we shall discuss below. Data quality/goodness does not do so easily. We include it as long as it is our conviction, that we need to assess how data quality/goodness check practices are applied in the development phase if we want to produce reliable systems with databases. These practices make sense starting from the design phase. The design stage allows us to have a proper knowledge of data structure and properties. From the moment these issues are known data quality should be considered. The reason is that data quality issues have an influence on database constraints, it may lead to define specific procedures; even it may lead to modify the design considering that good quality data might be unavailable at given scenarios.

We understand that process dimension is the start point, as long as it is the base for the capability dimension. We have focussed our work in relational databases. However it can be easily applied to other models. Object models present some characteristics that should be analyzed before extending our conclusions to them. In order to limit the definition of base practices we have constrained our work to the DB design phase.

Within this section, Introduction, we present the paper guidelines. Then we explain the basic concepts of the database design process model mapping in *A Generic*

Database Design Process and SPICE. Next section, *Data Quality/Goodness* describes a number of base practices. *Data Quality Assessment in SPICE* proposes where the previous practices can be included within SPICE. Finally we present a number of conclusions.

2 A Generic Database Design Process and SPICE

As figure 1 shows, EMEDITA is comprised of a set of phases, which contain stages which, in their turn, contain tasks and finally, basic activities, subtasks. Subtasks are those taken into consideration to develop the mapping with SPICE base practices. The findings of mapping a Generic database design process onto the SPICE-ISO/IEC 15504-process model were described in [6]. Main results are summarized below, and some of the basics of the mapping are presented in here.

SPICE is defined on a general-purpose basis. We studied up to what extent SPICE-ISO/IEC 15504 offers guidance for database design process assessment and improvement. Specific characteristics of DBs become a problem when treated like other conventional software components.

In order to judge the goodness of SPICE for DB developments, a mapping between a generic DB design methodology, represented by EMEDITA, and SPICE was carried out.

The first step was to define a group of tasks that, eventually, could be mapped onto SPICE base practices. As we already mentioned in the Introduction, the process dimension was took into consideration. The mapping was done maintaining the maximum coherence degree between both models and trying to adjust as much as possible their contents. Mapping showed that DB design tasks are not fully contemplated in SPICE. We could say that DB design tasks would not feel at ease in SPICE, as listed in figure 2. Once processes were mapped, the second step was to find a group of base practices where they could fit, as long as it could be possible.

The problem of comparing conventional software concepts with DB concepts raised up in the base practices mapping. The architectural design concept was considered to be similar to the idea of he conceptual schema definition in DB, as well as the concept of detailed design was compared with the rest of subtasks performed to achieve the Physical design. The mapping was carried out taking this criteria into consideration. Figure 3 shows the result.

Finally the work products of SPICE base practices were analyzed from the database point of view. They represented the most difficult part of the mapping as some of SPICE work products matched those of the DB design methodology, but their needed characteristics in a DB context were rather different from the SPICE specifications. Some other DB work products were missing in SPICE.

The main conclusions were that, as far as process dimension is concerned, we believe that SPICE presents a good basis to support DB design process. On the other hand SPICE lacks some issues that can be introduced, but that are not straightforwarded, to be presented as a model that should be followed in the assessment of DB methodologies. On the other hand SPICE structure has supported well the extensions that we suggested as interesting to be introduced.

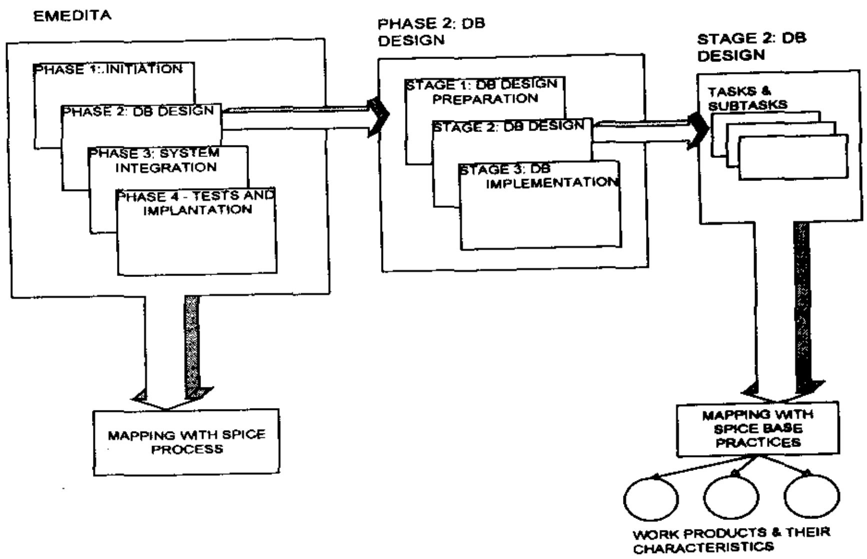

Fig. 1. EMEDITA structure and mapping philosophy

3 Data Quality/Goodness Practices

One of the conclusions extracted from the study [6] was the lack of processes and base practices addressing Data Quality (DQ) issues. In order to provide support and guidelines in the assessment of DQ, based on the EMEDITA approach, processes and base practices have been identified.

Data quality (DQ) is the measure of the agreement between the data views presented by an information system and that same data in the real world. The real concern with DQ is to ensure, not that the DQ is perfect, but that the DQ in our information system is accurate enough, and consistent enough for the organization to survive and make reasonable decisions [7]. At this point it must be also considered the context where data is included [8]. Data that can be appropriate for one use may not possess sufficient quality for another use [9]. The impact of poor DQ includes customer dissatisfaction, increased operational cost, less effective decision-making, and a reduced ability to make and execute strategy [10].

With data we face a problem. Data changes with time, while the database scheme remains basically unchanged. For example, if we have persons identified with an ID number (IDNUM) and with a name (NAME), it may happen that at the moment of loading the database IDNUM and NAME values are consistent, or they are not consistent, or even IDNUM has values not possible such as alphabetical characters, or NAME has wrong values even when being alphabetical. And a database operation is always a potential source of data not with the good quality. It is necessary to deal with these issues, early in the life cycle, at the design phase. Establishing data quality check procedures and practices may avoid future and unexpected performance problems. It is true that data use is one of the sources of information for data quality and that may change with the application life, but it does not seem a good idea just to forget the thing simply because it might change in the future.

In order to prevent problems that arise from the bad quality of the data, it is convenient to introduce a new group of practices. These practices will be defined within the context of SPICE.

The criteria followed to define the process and the suitable base practices is a generally accepted criteria, as explained in [1], [2], [3]: There are four different DQ dimensions (some authors, as [1] and [2], speak of categories or attributes) that define the quality of the data:

Intrinsic Dimension
Accessibility Dimension
Contextual Dimension
Representational Dimension

Each of these dimensions has a number of associated attributes. Once each dimension has been measured, the value associated to its attributes is used as an assessment of the DQ. In fact, if we know the measure of Current Quality, Required Quality and Expected Quality of each data set an improvement study may be carried out. That is to say, we can know Data Quality Improvement Cost and Value Added of this improvement, as described in [1], and its application on DB systems, [4].

If we define the practices needed to carry out a DQ study in terms of the dimensions and factors described above and we apply this practices to the DB design and data models we can avoid those risks related to bad data quality in an early phase of the development.

Following the SPICE scheme, the Data Quality process has been defined in the following way:

3.1 Data Quality Process

Purpose: The purpose of the data quality process is to ensure the quality of the data within the current context of the data. As a result of a successful implementation of the process:

Data will be consistent with the real world.
Transactional systems will be coherent, will have a reliable behavior and capacity for historical projection.
Historical files will contain data about data (metadata) and context data.

expected performance
nation for data quality
em a good idea just to

ility of the data, it is
ctices will be defined

le base practices is a
are four different DQ
attributes) that define

Once each dimension
d as an assessment of
Required Quality and
e carried out. That is
Value Added of this
ems, [4].
udy in terms of the
ices to the DB design
ility in an early phase

been defined in the

e quality of the data
ul implementation of

ehavior and capacity

text data.

EMEDITA Processes	DB	DW	15504	15504 Processes	Type	
Project Initiation	X	X				
Project planning & goals definition	X	X	CUS.1	Acquisition process	Basic	
Project goals identification	X	X	CUS.1.1	Acquisition preparation process	Component	
Involved areas identification	X	X	CUS.1.1	Acquisition preparation process	Component	
Participants identification & organisation	X	X	ORG.3	Human resource management process	Extended	¥
Project general planning	X	X	MAN.2	Project management process	New	¥
Project start	X	X	CUS.1.1	Acquisition preparation process	Component	
Project scope definition	X	X	CUS.1.1	Acquisition preparation process	Component	¥
Involved areas study	X	X	CUS.1.1	Acquisition preparation process	Component	
Involved areas data needs	X	X	CUS.1.1	Acquisition preparation process	Component	
Project environment definition	X	X	CUS.2	Supply process	Basic	
Initial planning preparation	X	X	MAN.2	Project management process	New	¥
Cost-benefit analysis	X	X	CUS.1.1	Acquisition preparation process	Component	¥
Revisit contents	X	X	CUS.1.2	Supplier selection process	Component	
Initial planning refinement	X	X	CUS.1.2	Supplier selection process	Component	
Standards adoption	X	X	CUS.1.2	Supplier selection process	Component	
Data base design	X					‡
Data warehouse design		X				‡
Data base design preparation	X		CUS.1	Development process	Basic	‡
Data warehouse design preparation		X	CUS.1	Development process	Basic	‡
Data source study	X	X	ENG.1.1	System requirements analysis & design process	Component	
Event analysis		X	ENG.1.1	System requirements analysis & design process	Component	‡
Storage capability estimation	X	X	ENG.1.1	System requirements analysis & design process	Component	
Data base design	X					‡
Data warehouse design		X				‡
System architecture definition		X	ENG.1.1	System requirements analysis & design process	Component	‡
Logical model design	X	X	ENG.1.3	Software design process	Component	
Physical model design	X	X	ENG.1.3	Software design process	Component	¥
Data source analysis	X	X	ENG.1.5	Software requirements analysis process	Component	
Interface design	X	X	ENG.1.5	Software requirements analysis process	Component	
Architecture components analysis		X	ENG.1.1	System requirements analysis & design process	Component	‡
Revisit component specification		X	CUS.2	Supply process	Basic	‡
Standard packages evaluation	X	X	CUS.1.1	Acquisition preparation process	Component	¥
Data quality strategies definition	X	X	MAN.3	Quality management process	New	¥
Inherited components integration study	X	X	ENG.1.1	System requirements analysis & design process	Component	
System initial documentation	X	X	SUP.1	Documentation process	Extended	
Architecture components implementation		X	ENG.1.4	Software construction process	Component	‡
System integration study	X	X				
System integration planning	X	X				
Participants identification & organisation	X	X	ORG.3	Human resource management process	Extended	¥
General planning	X	X	MAN.2	Project management process	New	¥
Assembly cost estimation	X	X	ENG.1.5	Software integration process	Component	
Physical model refinement	X	X	ENG.1.5	Software integration process	Component	
Assembly environment preparation	X	X	ENG.1.5	Software integration process	Component	
Test plan development	X	X	ENG.1.3	Software design process	Component	
Data loading	X	X	ENG.1.5	Software integration process	Component	
Final documentation generation	X	X	SUP.1	Documentation process	Extended	
System integration	X	X	ENG.1.5	Software integration process	Component	
Tests	X	X	ENG.1.6	Software testing process	Component	
Data load test	X	X	ENG.1.6	Software testing process	Component	
System test	X	X	ENG.1.6	Software testing process	Component	
Tests verification	X	X	ENG.1.6	Software testing process	Component	
System implantation & acceptance test	X	X				
System implantation planning	X	X				
User training plan developement	X	X	ORG.3	Human resource management process	Extended	
Implantation environment stablishment	X	X	ENG.1.7	System integration & testing process	Component	
Final data loading	X	X	ENG.1.7	System integration & testing process	Component	
System implantation	X	X	ENG.1.7	System integration & testing process	Component	¥
Hardware & software systems adaptation	X	X	ENG.1.7	System integration & testing process	Component	¥
Existing data conversion	X	X	ENG.1.7	System integration & testing process	Component	¥
Current system stop & backup	X	X	ENG.1.7	System integration & testing process	Component	¥
Conversion rutines execution	X	X	ENG.1.7	System integration & testing process	Component	¥
Acceptance testing	X	X	ENG.1.7	System integration & testing process	Component	
Acceptance test preparation	X	X	ENG.1.7	System integration & testing process	Component	
Acceptance test execution	X	X	ENG.1.7	System integration & testing process	Component	
Last chages incorporation	X	X	ENG.1.7	System integration & testing process	Component	¥
‡ Process not common to DB & DW				¥ It can be mapped to different processes		

Fig. 2. EMEDITA vs SPICE

3.2 Base Practices

DQ.BP1 Study Intrinsic Dimension. Perform a study of accuracy, objectivity, believability and reputation of the data to be stored, as the database conceptual model (e.g. Entity relationship E/R) specifies.

Referring to the example of the IDNUM and NAME set in the former section, this will lead us to set a number of user constraints and/or to define a number of processes that guaranteed that data have proper values. It should be understood that it is not so simple as setting a procedure that checks that IDNUM is numeric. It would be also necessary to check that IDNUM and NAME are consistent, and this requires further research (before and after the database load). To see the importance of this, if IDNUM is found to have a quality range below what could be considered as acceptable (or that it might happen in the future) we could/should not use it as a key for a relation.

DQ.BP2 Study Accessibility Dimension. Perform a study of accessibility and security of the data, as specified in the database conceptual model (e.g. E/R).

DQ.BP3 Study Contextual Dimension. Perform a study of the relevancy, added value, timeliness, completeness and the amount of data to be stored and managed.

DQ.BP4 Study Representational Dimension. Perform a study of the interpretability, ease of understanding, concise representation and consistency of the data.

DQ.BP5 Carry out data quality assessment and improvement. Develop a data quality assessment and data base conceptual model (e.g. E/R) improvement. Establish a rating system that shall allow the calculation of Current Quality, Required Quality and Expected Quality and database improvement cost and Added Value in order to evaluate whether the improvement of the current database conceptual model (e.g. E/R) is needed.

4 Data Quality Assessment in SPICE

In this section we suggest our approach to include DQ practices within the SPICE framework. One of the issues that came up when defining EMEDITA, was to decide at what point of the DB life cycle DQ actions must be undertaken. No literature can be found on this matter, but we understand that in order to approach the problem from the earliest possible stage, it is recommendable to start once the entities and their relations in the DB are defined. That is when the core conceptual schema exists. Definition of data functional dependencies may also be convenient at this stage.

One of the advantages of examining the problem at this point is saving costs in the development of the DB, since it is previous to implementation and the decision of removing or adding new user constraints and/or relationships and/or entities has no side effects on the DB design.

SPICE does not provide specific support for DB design as explained in [6], and where we defined our mapping scheme. This mapping scheme guides us to locate where the entities and relation definition are performed. These tasks are placed within the SPICE process ENG 1.3 (Software design process), more exactly in the

accuracy, objectivity,
base conceptual model

he former section, this
a number of processes
:rstood that it is not so
ieric. It would be also
id this requires further
importance of this, if
uld be considered as
uld not use it as a key

y of accessibility and
lel (e.g. E/R).
f the relevancy, added
red and managed.
n a study of the
and consistency of the

ment. Develop a data
mprovement. Establish
ality, Required Quality
dded Value in order to
conceptual model (e.g.

tices within the SPICE
IEDITA, was to decide
aken. No literature can
roach the problem from
the entities and their
ceptual schema exists.
ient at this stage.
t is saving costs in the
on and the decision of
and/or entities has no

s explained in [6], and
me guides us to locate
These tasks are placed
s), more exactly in the

SPICE base practices Develop Software architectural design and Detailed design. This is to be represented in figure 3.

We believe that it is at this point where the Data quality process should be placed. Actions taken in the data quality process might affect design decisions though. Therefore feedback is required. Figure 4 shows where DQ process can be placed.

In any case, DQ should be carried out not only at the design stage but also during the maintenance of the DB. The real difficulty with data quality is change. Data definition is static, but the real world keeps data (the contents) changing. Therefore a feedback is necessary as discussed in [7]. At this point the activities that must be carried out are basically:

Compare data in the DB with the data in the real world
Correct any inconsistencies and insert the corrections in the DB

Nevertheless, designers should anticipate future problems derived from the lack of data quality and to prevent them during the design phase. It is also true that some issues will not appear before DB becomes operational. It is likely that some of the problems concerning DQ that are encountered at the design stage might also appear during the use of the DB. Therefore ensuring that some of the DQ base practices are carried out during the maintenance is just a supporting action.

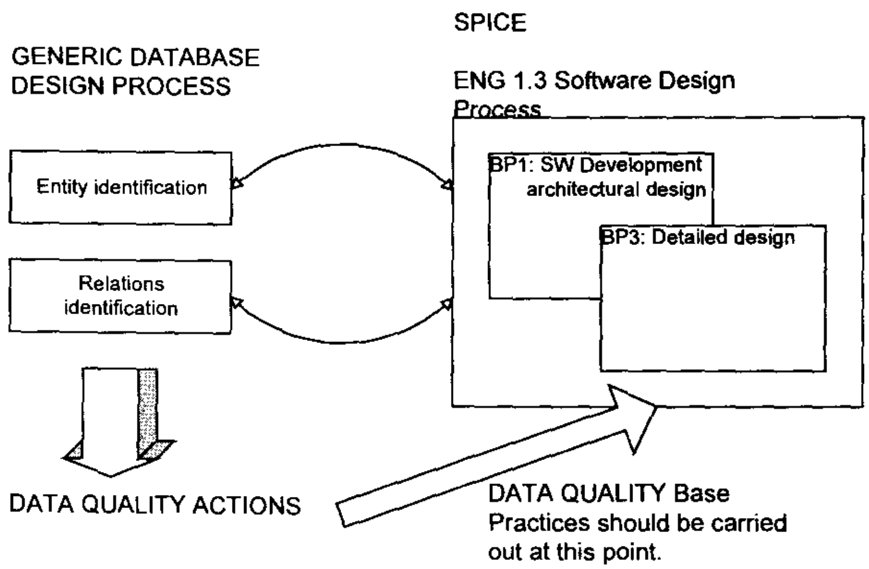

Fig. 3. Data Quality in SPICE

Generic Database design Subtasks	SPICE process	Generic Database design Base practices	SPICE base practices
Entity identification	ENG 1.3: Software design process	Entity determination and description	BP1: Develop SW architectural design
Relations identification		Relations determination and description	
Attributes identification		Determine entities and relations attributes	
Keys identification		Entity keys determination	
Conceptual design		E/R model, External schema design	
E/R revision		E/R modification	BP3: Detailed design
Transactions design		Transactions specification	
Normalize and design changes		Get initial tables, 1FN, 2FN, 3FN, FNBC, changes.	
E/R final diagram		Obtain E/R final diagram	
Obtain tables		Obtain tables with keys and attributes	
Integrity study		Determine attribute domains Constraints study, rules, triggers	
Specific logic design		Data and constraints definition according to the DBMS DB creation sentences definition	
Physical design		DB store files definition, Files location, Query frequency study, Index election	

Fig. 4. Figure 3- Mapping with SPICE processes and base practices

SPICE process ENG.2 System and software maintenance process, ought to be reviewed in order to incorporate the suitable actions described in section 3. We believe they are of sufficient importance to be explicitly stated instead of taking them for granted in SPICE general statements.

5 Conclusions

This document aims at establishing the necessary foundations to start considering data quality matters for the Database design process assessment in the frame of SPICE-ISO/IEC 15504. As we have explained, it is not possible to forget maintenance, however.

DQ actions turn up in two different phases of the DB design process: the first DQ actions are considered, is once the group of entities and the relations among them of the DB are defined. Then, the data inferred from these entities and relations could undergo the necessary actions that will ensure their goodness. The defined entities and relations are then refined according to the results of the Data Quality assurance.

One of the main drawbacks of defining DQ processes is the lack of available support, methods and standards, due to the little advance in the DQ science. Therefore the proposed process and associated base practices shall be considered as guidelines for further and broader studies.

The process where we can introduce practices during the database design is within SPICE is process ENG 1.3 (Software design process), more exactly in the SPICE base practices Develop Software architectural design and Detailed design.

Although performing these actions at the design stage saves a lot of work and costs, DBs might never be free from poor data quality. Therefore the same actions shall continue during the maintenance of the DB. Therefore, we have also to consider SPICE process ENG.2 System and software maintenance process, and introduce there the practices as well. It is our believing that SPICE framework ought to be reviewed in order to incorporate the suitable actions described in section 3.

Acknowledgements

Authors are indebted to Ivan Bernardo and Carolina Fernandez for their useful comments. The work presented within this paper has been performed in the context of the project PROCBD and under a contract with CRONOS IBERICA S.A. This project is partially supported by the Ministry of Industry of Spain Ref. T87/1998, Programme ATYCA.

References

1. Ballou, Kumar and Tayi. Enhancing Data: Quality in Data Warehousing Environments. Communications of the ACM. January 1999, Vol. 42, No. 1.
2. Total Data Quality Management. Richard Y. Wang. Communications of the ACM. February 1998, Vol. 41, no. 2.
3. Beyond accuracy: What data quality means to data consumers. Richard Y. Wang, D. M. Strong. J. Manage Information Systems. 12, 4. 1996.

4. Methodology for the Development of Data Warehouse Projects (in Spanish). Oscar Carmona, Hector Garcia. Minor Thesis. Technical University of Madrid. E.U. Informatica. 1999.
5. ISO/IEC JTC1 TR 15504:1999 Information Technology - Software Process Assessment. Parts 1-9.
6. Database Design in the Context of the SPICE Process Dimension. Juan Garbajosa, Hector Garcia, Maria Alandes and Mario Piattini. SPICE 2000 Conference. Limerick, Ireland, May 2000.
7. Data Quality and Systems Theory. Ken Orr. Commun. of the ACM. Vol 41. N 2. Feb 1998.
8. Data Quality in context. D. Strong, Y. Lee, and R. Wang. Commun. of the ACM. Vol 40, N. 5 May 1997.
9. Examing Data Quality. G. K. Tayi and D.P: Ballou. Comm. Of the Acm. Vol 41 N2. Feb. 1998.
10. The impact of Poor Data Quality on the Typical Enterprise. T.C. Redman Comm. Of the Acm. Vol 41 N2. Feb. 1998.