Software Quality Management IX

Pathways to Software Quality

Editors:
R Dawson
Loughborough University
G King
Southampton Institute
M Ross
Southampton Institute & BCS Quality SIG
G Staples
BCS Quality SIG Chairman

THE BRITISH COMPUTER SOCIETY

Conference endorsed by

COUNCIL OF EUROPEAN
PROFESSIONAL INFORMATICS SOCIETIES

ISBN 1-902505-40-9

First published 2001
Printed in Great Britain
Published by the British Computer Society

Ninth International Conference on Software Quality Management

SQM2001

CONFERENCE CHAIRMEN

R Dawson
G Staples

CONFERENCE DIRECTORS

M Ross
G King

INTERNATIONAL ADVISORY COMMITTEE

H Basson
P Burgess
M Curtis
T Daughtrey
J Dolado Cosin
D Edgar-Nevill
E Georgiadou
E Gray
T Hall
B Hirsh
D Kitson
P Linecar
M Love
Z Mahmood
Ming Jing
B Myers
T P Rout
A Symons
H Wickberg
S Zahran
J Brinkworth
D Chadwick
K Daily
R Dawson
A Dorling
I Fernandez
A Grady
R Green
S Hill
R Hunter
P Kuvaja
C Long
C Mackie
D Maisey
M Mullerburg
P Oriogun
V Sivess
J B Thompson
D Wilson

The front cover design is taken from " An Analysis of Options for Handling Legacy Systems" by Mark de Chazal and Ray Dawson.

PREFACE

This volume contains the edited proceedings of the ninth International Conference on Software Quality Management, SQM2001, held in April 2001 in Loughborough, organised by the Quality Special Interest Group of the British Computer Society. This conference has been combined with the Quality Special Interest Group's sixth annual INSPIRE conference.

The objective of this series of annual conferences is to promote international co-operation among those concerned with software quality and process improvement by creating a greater understanding of software quality issues and by sharing current research and industrial experience

The papers cover a broad spectrum of practical experience and research. The topic areas include process improvement, process maturity models, quality metrics, project management issues, approaches to systems development, risk analysis, e-commerce systems, professional issues, teaching and training issues

We would like to thank the many people who have brought this ninth international conference into being: the Conference Chairmen, the International Advisory Committee, particularly for all their hard work in reviewing both the abstracts and the final papers, and the committee members of the British Computer Society's Quality Special Interest Group.

The organisers would like to thank Loughborough University and Southampton Institute for their sponsorship.

Margaret Ross and Geoff Staples
Conference Directors

CONTENTS

Section 1: Quality in Systems Development

Section 2: Issues of Electronic Communications

Section 3: Systems Development Issues

Section 4: Teaching and Training Issues

Integration of Information in a Training Environment for Software Project Management

J. Tuya, P. Fernández, M.A. Prieto (Univ. of Oviedo),
J. Aguilar, I. Ramos, J. Riquelme, F. Ferrer, M. Toro (Univ. of Seville),
M. Ruiz-Carreira (Univ. of Cádiz),
D. Rodriguez-García, M. Satpathy, R. Harrison (Univ. of Reading),
A.Ruiz de Infante, J.J. Dolado (Univ. of the Basque Country) and
R. Matilla, M.A. Álvarez (SYSECA Cantábrico)
Spain

Abstract

Project management can still be considered as an art in which the use of quantitative information tends to promote a more scientific approach to management. We present the structure and main components of an environment for training project managers. The goal of the system is to have a uniform structure so that new techniques can be incorporated into the structure smoothly. The system collects and records both actual and simulated project data and implements different techniques such as machine learning, project tracking, dynamic modelling, etc. The basic assumption of this work is that management decision should be supported by integrating different sources of information.

1 An Environment for Training

Project management is one of the activities in software engineering that is still in need of a solid technical basis. Although it is not clear that this will ever be achieved, every step in the areas of estimation, tracking, data interpretation, etc. will make a project closer to the goal of "reliable, on time and within budget". Training systems for presenting different scenarios to the future software managers is a way of overcoming the lack of data related to project management.

In actual management settings, the manager has to take decisions in relation to an approximate picture of the project environment. The fact that much of the information about past projects is unknown or uncertain makes it more difficult to ascertain the final project parameters. Moreover, the available data comes from

25. Cimitile, A, De Lucia, A, Di Lucca, A, & Fasolino, A. R, "Identifying Objects in Legacy Systems", *Proceedings of the 5th Workshop on Program Comprehension (WPC97)*, 1997
26. Seacord, R. C, Wallnau, K, Robert, J, Comella-Dorda, S, & Hissam, S. A. "Custom vs. Off-the-Shelf Architecture," *Proceedings of 3rd International Enterprise Distributed Object Computing Conference*, University of Mannheim, Germany, September 1999
27. Johnson, M. "A Beginner's Guide to Enterprise JavaBeans," *Java World 3*, 10. <URL: http://www.javaworld.com/javaworld/jw-10-1998/jw-10-beans.html> October 1998
28. Weeiderman, N. H, Bergey, J. K, Smith, D. B, & Tilley, S. R. *Approaches to Legacy System Evolution*, Technical Report, CMU/SEI-97-TR-014, Pittsburgh College, Carnegie Mellon University, December 1997
29. Campbell, Ian, *The Intranet: Slashing the Cost of Business*, International Data Corporation, Framingham, MA. 1996

MANTICA: An Object-Relational Database Metric Tool

Francisco J. Granados, Coral Calero, Mario Piattini
Grupo ALARCOS
E.S. Informática-UCLM
Ciudad Real (Spain)
e-mail: {ccalero, mpiattin}@inf-cr.uclm.es

Abstract

The final products of software engineering need to be of good quality to make those products competitive. The way we know if our products have a high quality is to use metrics.

If the acquisition and presentation of metrics is automated, then the effort of applying metrics to our software will be minimised. To reach this automation we need metric tools.

This paper presents MANTICA, a metric tool oriented to the field that has not yet been covered by the metrics technology, databases. The future of databases is oriented to the object-relational ones and for this the reason MANTICA is a tool that covers the acquisition and presentation of object-relational database metrics.

1. Introduction

In software engineering as in other engineering areas, measuring is essential. Many metrics have been developed over past decades to improve the quality of final products. These metrics have been oriented to measure programs but not data. However, nowadays databases are the core of the Information Systems and to design metrics for them is essential. For that reason we have developed specific metrics for measuring conceptual and logic database models (Calero et al., 1999, Piattini et al., 2001, Genero et al., 2001).

In the field of databases, object-relational technology joins relational and object properties. Relational technology is the most commonly used in the market and the object technology gives representational power. So, the union of these two characteristics makes a great future option (Stonebraker and Brown, 1999).

2. Tools

We have seen the relevance of object-relational databases and the support that metrics can bring to evaluate the quality of these databases. It would be a great help if we achieve the automation all the metrics computation, because if we reach it we minimize the effort and the evaluation task costs, and we avoid mistakes that can be committed by the human factors in manual data calculation. To reach all of these advantages we need a metric tool.

The advantages of using a tool for automating the acquisition, presentation and analysis of metric values are (Lavaza, 2000):

- Possibility of getting metric values without effort.
- Minimize mistakes in computating metrics, obtaining more precision.
- Focus on the analysis of measurement not on the acquisition step.

There are many studies which evaluate and compare metric tools (Software Measurement Laboratory, 2000; Giles and Daich, 1995; Daich and Giles, 1995; Daich et al., 1995 Erickson and Steadman, 1995; Kingsbury and Dawood, 1995; Giles and Barney; 1995), but none of the tools analysed refers to measurement ofdatabase schemas The existing tools have a lack of database metric tools because all of them are program oriented.

Giles and Daich (1995) said that there are three main tasks that metric tools should undertake:

1. Get data: Manually, semiautomatically, automatically and programable.
2. Measurement analysis: Data storage, data recovery and statistical analysis.
3. Data Presentation: Tables, Graphics, the possibility to export files to other applications.

Based on the main tasks discussed above, Giles and Daich (1995) have classified two tool categories.

1. General metric tools: They have been designed to work with metrics.
2. Specific metric tools: They have been designed to do other tasks and, as an add-on, compute some specific metrics.

The usefulness of a metric tool has been proved and if we focus that tool on the future of databases, the object relational databases, we find the perfect cocktail.

3. What MANTICA is

MANTICA is a software metric tool that is focused on object-relational databases. As we have seen before, there are many metric tools, but all of them are oriented to measure programs. The field of databases does not have specific tools for automate metrics, so we have tried to fill that gap with MANTICA.

MANTICA has the following design characteristics:

- The software runs on a Windows 9x environment.
- The programming language is MS Visual Basic 6.0.
- The Database used by MANTICA has a MS Access 97 format.
- MANTICA is able to import Oracle 8 and SQL ANSI schemas. To import Oracle 8 schemas MANTICA uses ODBC for connecting to these schemas.

4. MANTICA Metrics

The software is oriented to the object-relational databases measurement and to reach that objective MANTICA uses specific object-relational metrics, relational metrics and object oriented metrics.

The metrics that MANTICA applies to any object-relational database schema are:

Relational (Piattini et al., 2001)
We present the metrics proposed by Piattini et al. (2001) for relational databases.
RD (Referential Degree): Defined as the number of foreign keys in a table or schema.
DRT (Depth Referential Tree DRT): The length of the maximum referential path of the relational schema.
COS (Cohesion of the schema): The aggregate of the square of the number of tables in each un-related subgraph in the subgraphs that represent the database.

Object Oriented (Chidamber and Kemerer, 1991)
Here is presented one of the more representative metric series, the Chidamber and Kemerer (1991) one:
CBO (Coupling Between Objects): Defined as the number of classes that are linked to our class.
DIT (Depth of Inheritance Tree): In which level in the inheritance tree is the class.
NOC (Number Of Children): Defined as the inheriting class number.
RFC (Response For a Class): The number of classes that are called by another one.
WMC (Weighted Methods per Class): Defined as all class methods size.
LCOM (Lack of Cohesion in Methods): The number of common attributes that are used by different methods.

Object-Relational (Calero et al., 1999)
The metrics presented now have been proposed by Calero et al. (1999) for object-relational databases.
SS (Schema Size): Defined as the aggregate of table size.
TS (Table Size): Simple columns size plus complex columns size.
TSSC (Table Size Simple Columns): We consider all simple columns as having a size of 1, so this metric is defined as the number of simple columns in the table.
TSCC (Table Size Complex Columns): All complex columns size.
CCS (Complex Columns Size): The size of the hierarchy, in which the column is defined.
SHC (Size Hierarchy Class): Defined as all classes, that formed the hierarchy, size.
SC (Size Class): Defined as follows

$$SC = \frac{SAS + CAS + SMC}{NHC + NCU}$$

where SAS is the number of simple attribute, CAS is defined as all complex attribute size, SMC is the Methods size, NHC is the number of hierarchies in which the class is and NCU is defined as the number of table columns that the class is defined in.

5. Design and Implementation

The system has been developed on three separated modules:

- Data acquisition and Metric Computing Module (responsible for the database schema importation and the metrics computation)
- Storage Module (responsible of the storage of all the information for the first module)
- Data Presentation Module (responsible for showing all the information to the user).

These modules are related with a central database, called MetaDatabase. This MetaDatabase must store all the information about database schemas and their metrics.

Due to the fact that the database schema has an structure very similar to a graph (classes related with tables or with other classes, etc..), there is a main class in MANTICA, called Tgrafo, that has the mission of being an interface among all the application actions: Data computing, Results Presentation, etc..

So, the Tgrafo class is formed by classes that represent the structures that we can find in an object-relational database schema: Tables, Classes and Results (Metrics). All the metrics information is stored in the Results class. So if we want to measure the schema we use the Results class in Tgrafo, and if we want to measure other entities like tables, classes or methods, we use the Results class in Tables class, Classes class or Methods class.

The application has a basic design with the structure shown in figure 1:

With the appropriate queries on these tables we can get the information required for building the schema internal structure from which the tool can obtain the metrics values.

SQL : 1999
To import from a text file with an SQL99 definition is very important to know which information is relevant. For building the schema's internal structure, information about tables, classes and tables is needed. All this information is related with three SQL sentences: "Create Table", "Create Type" and "Create Method". So, with these sentences we can get the information needed.

In each case an automaton is developed to recognize the sentence syntax. For example, the "CREATE TYPE" syntax is:

```
CREATE TYPE Type_Name AS (
        Attribute_Name Data_Type,
        ...) NOT FINAL INSTANTIABLE
        METHOD Method_Name(parameters) RETURNS Data_Type,
...;
```

and is recognized with the automaton shown in figure 3:

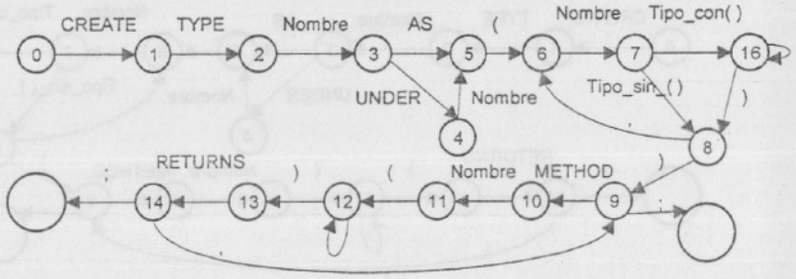

Figure 3. Automaton for the CREATE TYPE sentence

There are two definitions that need to be explained, simple attribute and complex attribute. We made the difference between them through the data type. That is, a simple attribute is called in SQL : 1999 predefined type (CHARACTER, VARCHAR, DATE, etc..). A complex attribute type does not belong to this relation.

6. MANTICA User Interface

MANTICA is so easy to work with, because it has a visual graphic user interface based on Windows OS. Once the application is running, the schema manager window is shown, and then you can select one of the schemas and to see its properties or to calculate its metrics values. Figure 4 shows the main screen of MANTICA.

With the appropriate queries on these tables we can get the information required for building the schema internal structure from which the tool can obtain the metrics values.

SQL : 1999

To import from a text file with an SQL99 definition is very important to know which information is relevant. For building the schema's internal structure, information about tables, classes and tables is needed. All this information is related with three SQL sentences: "Create Table", "Create Type" and "Create Method". So, with these sentences we can get the information needed.

In each case an automaton is developed to recognize the sentence syntax. For example, the "CREATE TYPE" syntax is:

```
CREATE TYPE Type_Name AS (
        Attribute_Name Data_Type,
        ...) NOT FINAL INSTANTIABLE
        METHOD Method_Name(parameters) RETURNS Data_Type,
    ...;
```

and is recognized with the automaton shown in figure 3:

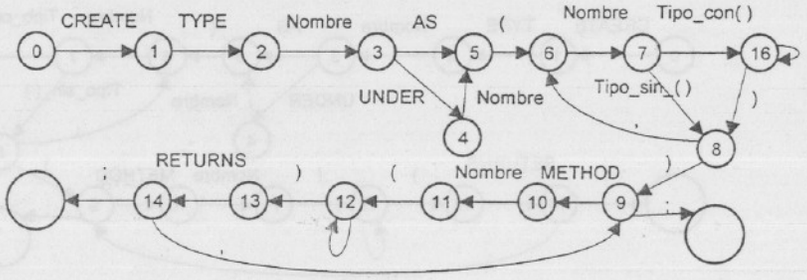

Figure 3. Automaton for the CREATE TYPE sentence

There are two definitions that need to be explained, simple attribute and complex attribute. We made the difference between them through the data type. That is, a simple attribute is called in SQL : 1999 predefined type (CHARACTER, VARCHAR, DATE, etc..). A complex attribute type does not belong to this relation.

6. MANTICA User Interface

MANTICA is so easy to work with, because it has a visual graphic user interface based on Windows OS. Once the application is running, the schema manager window is shown, and then you can select one of the schemas and to see its properties or to calculate its metrics values. Figure 4 shows the main screen of MANTICA.

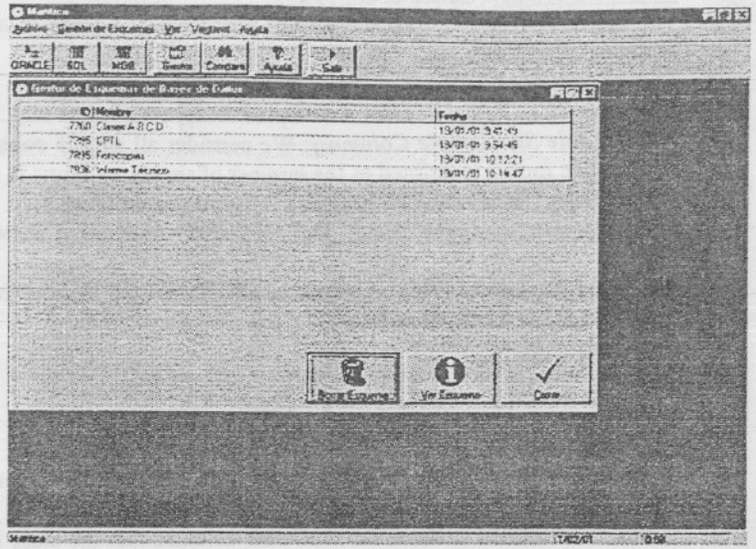

Figure 4. MANTICA main screen

All the information related with an object-relational database schema is shown on the screen presented in figure 5.

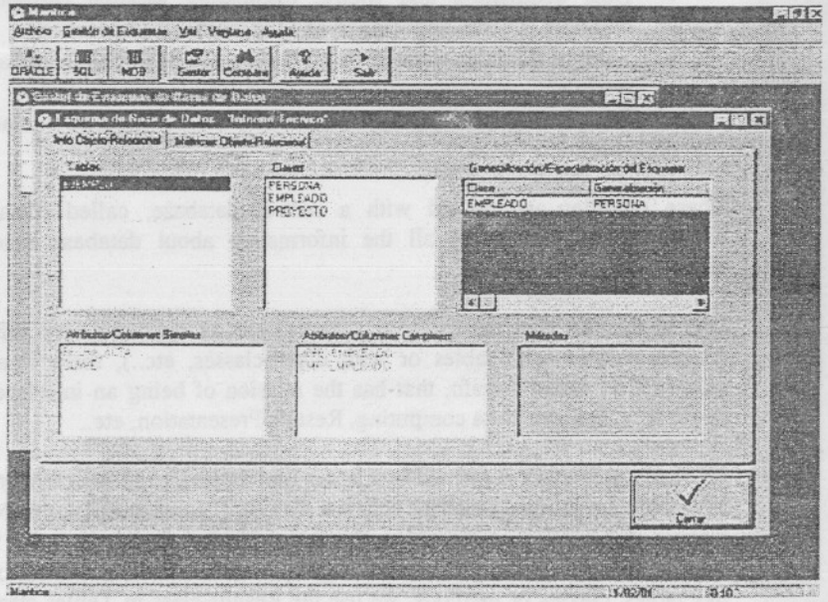

Figure 5. Object-relational information screen

In figure 6, the screen with the selected metrics that have been applied to the schema we are working with is shown. To obtain the table, class or method metrics desired you only have to select th corresponding item and only these metrics values will be shown

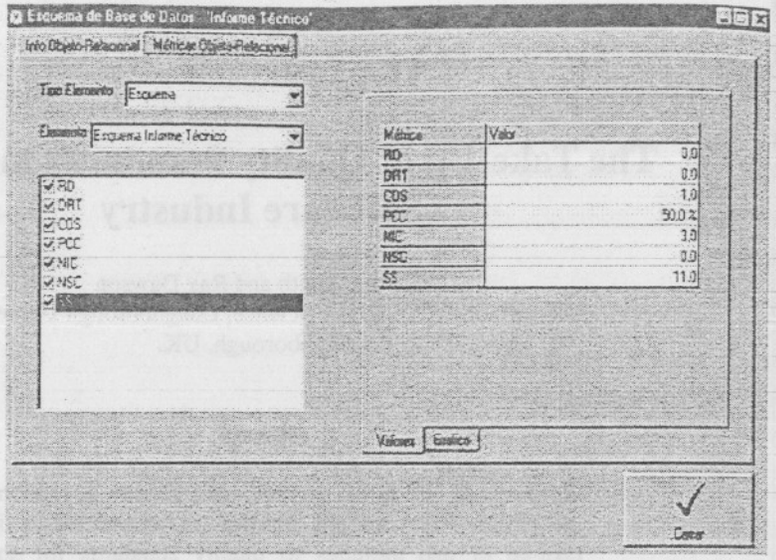

Figure 6. Metrics screen

Also it is important to note that the metrics values can appear graphically (figure 7) or through its value (figure 8).

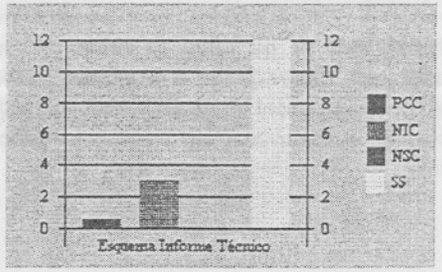

Figure 7. Graphical information about metrics values

Metric	Value
PCC	50,0 %
NIC	3,0
NSC	0,0
SS	12,0

Figure 8. Text information about metrics values

To import new database schemas it is necessary to choose SQL or ORACLE import and then if the selected option is SQL the schema database file name is given and if the option is ORACLE, the login and the password user, the connection string and the schema name are needed (figure 9).

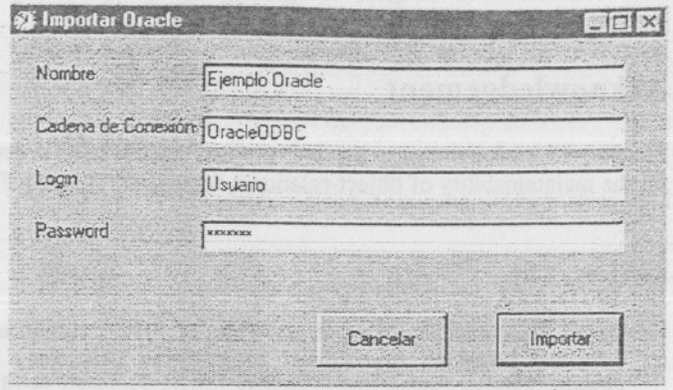

Figure 9. Oracle importation screen

MANTICA also has a module in order to compare schemas (figure 10).

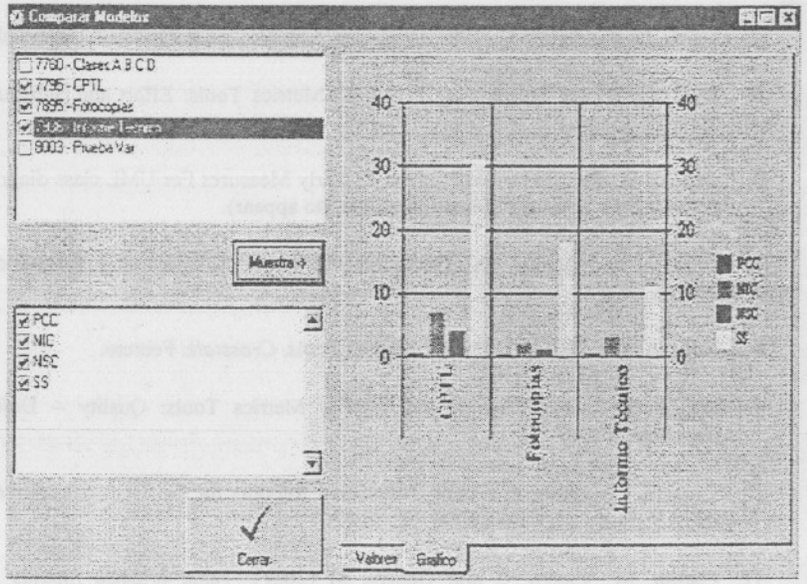

Figure 10. Schemas comparison screen

7. Conclusions and future work

Metrics are needed to obtain high quality in software products and to achieve all the metrics computation a metric tool is necessary. Nowadays there is a lack of database metric tools and for that reason MANTICA tries to fill that gap.

In future MANTICA will be extended to new metrics and database schemas. MANTICA has been designed in a generic way to make it easy to write new versions that improve the current one.

Acknowledgement

The present work belongs to the I+D project "MANTICA: A set of metrics definition for the maintainability of object-relational databases" (CICYT-FEDER - 1FD97-0168 TIC).

Reference List

1. Calero, C., Piattini, M., Ruiz, F. and Polo, M. (1999) Validation of metrics for Object-Relational Databases, *International Workshop on Quantitative Approaches in Object-Oriented Software Engineering ECOOP99*, Lisbon (Portugal), 14-18 June., pp 1-12.

2. Chidamber, S. and Kemerer, C. (1991) Towards a metric suite for object oriented design. *Proc OOPSLA'91. Sigplan Notices*. 26(11). pp.197-211.

3. Daich, G., Price, G. Dawood, M. (1995). Metrics Tools: Size. *Crosstalk*. Abril.

4. Daich, G. and Giles, A. (1995). Universal Metrics Tools. *Crosstalk*. Septiembre.

5. Erickson, D. and Steadmand, T. (1995). Metrics Tools: Effort and Schedule. *Crosstalk*. Marzo.

6. Genero, M., Piattini, M. and Calero, C. Early Measures For UML class diagrams. *L'Objet*. 6(4), Hermes Science Publications, 2001, (to appear).

7. Giles, A. and Barney, D. (1995). Metrics Tools: Software Cost Estimation. *Crosstalk*. Junio.

8. Giles, A. and Daich, G. (1995). Metrics Tools. *Crosstalk*. Febrero.

9. Kingsbury, J. and Dawood, M. (1995). Metrics Tools: Quality – Defect Tracking. *Crosstalk*. Mayo.

10. Lavazza, L. (2000). Providing Automated Support for the GQM Measurement Process. *IEEE Software* May/June 2000. pp. 56-62.

11. Piattini, M., Calero, C and Genero, M (2001). Table oriented metrics for relational databases. Accepted for publication in *Software Quality Journal*.

12. Software Measurement Laboratory (2000). http://ivs.cs.uni-magdeburg.de/sw-eng/us/

13. Stonebraker, M. and Brown, P. (1999) *Object-Relational DBMSs tracking the next great wave*, California, Morgan Kauffman Publishers.

The Take-Up of Quality Standards in the UK Software Industry

Graeme Smith and Ray Dawson
Department of Computer Science, Loughborough University,
Loughborough, UK.

Abstract

A survey of UK software companies chosen at random from the Yellow Pages and the Internet was conducted to find the take-up of and attitudes to quality standards for software development. The survey generated 8 responses from a possible 40. Although the survey was too small to draw definite conclusions, the indications are that, at least amongst smaller organisations, there is little interest in such standards. The respondents stated that they believed the standards produced nothing more than a bureaucratic paper trail that offered little or no value to either the developers or their customers. This response is examined in the light of research into organisational culture and preconceived ideas. The paper provides a useful starting point for further research in this area.

1.0 Introduction

Many are aware of the existence of models in the area of software project management. For example, the Waterfall Model, the Spiral Model and the Evolutionary Prototype. In recent years however, several new models for software development have been introduced, including TICKIT, BOOTSTRAP, SPICE [1] and the Capability Maturity Model [2].

The principle aim of this paper is to highlight some preliminary findings from an investigation into the current state of affairs in the sphere of software development process management. This paper will discuss the method chosen to investigate this (a questionnaire), and also focus on some results. These will highlight what, if any, take up of standards or models is being made and which particular standards