The *METRICS NEWS* can be ordered directly from the Editorial Office (for address see below).

Editors:

Editorial Office: Otto-von-Guericke-University of Magdeburg, FIN/IVS, Postfach 4120, 39016 Magdeburg, Germany
Technical Editor: DI Erik Foltin
The journal is published in one volume per year consisting of two numbers.

C A L L F O R P A P E R S

Workshop des GI-Arbeitskreises "Softwaremetriken" der FG 2.1.9
vom 30.9. - 1.10.99 an der Universität R E G E N S B U R G

Der Einsatz von Metriken hat insbesondere in den letzten beiden Jahren auch im deutschsprachigen Raum deutlich zugenommen. Dieser Einsatz reicht von (weiteren) ISO 9000-Zertifizierungen, über CMM-Level-Bewertungen und die verstärkte Anwendung von Function-Point- oder COCOMO-Abschätzungen bis hin zur permanenten Installation eines Metriken-Programmes in einer Firmen einschließlich der dafür notwendigen Meßdatenhaltung. Bei der methodischen Vorgehensweise hat sich besonders die Goal-Question-Metrics-Methode bewährt.

Der diesjährige Workshop widmet sich daher vor allem (jedoch nicht ausschließlich) den Themenschwerpunkten

- Erfahrungsberichte zur Metrikeneinführung in der Praxis,
- Anwendungserfahrungen bei der Aufwandsschätzung, insbesondere mit der Function-Point-Methode,
- Lösungsformen und Erfahrungen in der Meßdatenhaltung,
- theoretische Grundlagen der metriken-basierten Software-Entwicklung und -Anwendung,
- Anwendung neuer Technologien für die Umsetzung und Installation von Metriken-Programmen,
- Erschließung weiterer Bereiche durch quantifizierte Meß- und Bewertungsformen (komponentenbasierte, multimediale, verteilte, partizipatorische Software-Entwicklungs- und -anwendungsformen).

Für die Präsentation sind ca. 20 Minuten vorgesehen, um jeweils ausreichend Zeit für Diskussionen zur Verfügung zu haben. Darüber hinaus sollen wiederum die bewährten Panel-Diskussionen Anwendung finden. Die Beiträge werden im Rahmen der Buchreihe "Information Engineering und IV-Controlling" beim Deutschen Universitätsverlag veröffentlicht.

Für die Zeit des Workshops besteht die Möglichkeit von Tool-Demonstrationen zum Gebiet der Software-Messung und-Bewertung.

Ganze Beiträge oder Abstracts schicken Sie bitte per Post oder per Email bis zum ***16. August 1999*** an eine der beiden Adressen

Prof. Dr. Franz Lehner
Universität Regensburg
Lehrstuhl für Wirtschaftsinformatik III
Universitätsstraße 31
D-93053 Regensburg
Email: franz.lehner@wiwi.uni-regensburg.de

Prof. Dr. Reiner Dumke
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Postfach 4120
D-39016 Magdeburg
Email: dumke@ivs.cs.uni-magdeburg.de

Sollte die Zahl der Präsentationen zu groß werden, so treffen die Organisatoren eine Auswahl.

Advanced Program

of the 9th International Workshop on Software Metrics

in Mont-Tremblant, Montreal, Canada, Sept. 8-10, 1999

Day 1st September 8

Validation and Evaluation of Measurement

8:30	WOLFF, Sybille BOURQUE, Pierre DUPUIS, Robert ABRAN, Alain	The Role of Measurement in Fundamental Principles of Software Engineering
9:00	ZUSE, Horst	Validation of Software Measures and Prediction Models
9:30	DION, Francis	Décisions et justifications dans un contexte industriel de génie logiciel
10:00	DUMKE, Reiner	Criterias for Software Measurement Evaluation
10:30	**Break**	

Measurement Techniques & Tools

11:00	BLACK, Sue WIGG, David	X-Ray: A Multi-Language, Industrial Strength Tool
11:30	BUGLIONE, Luigi ABRAN, Alain	LIME: A Three-Dimensional Software Performance Measurement Model for Project Management
12:00	**Lunch Time**	
13:30	FRENCH, Vern	Establishing Software Metric Thresholds
14:00	KOKOL, Peter	Universality – A Need for a New Software Complexity Metric
14:30	FOLTIN, Erik DUMKE, Reiner SCHMIETENDORF, Andreas	Conceptions and Experience of Metrics Data Bases
15:00	SCHMIETENDORF, Andreas DIMITROV, E. DUMKE, Reiner FOLTIN, Erik WIPPRECHT, M.	Conception and Experience of Metrics-Based Software Reuse in Practice
15:30	**Break**	

Measurement Programs

16:00	BURTON, Valérie, ALBERT, Linda	"Revenue Canada’s Extended Metrics Plan" or "Metrics with Muscle"
16:30	WHITE, Leonard	Evaluating Measurement Methods
17:00	MOLINIÉ, Luis	Outsourcing Contracts : An Economic Analysis
	Social Activity	

Day 2nd September 9

Functional Size Measurement Methods

8:30	SYMONS, Charles FAGG, Peter ABRAN, Alain MORRIS, Pam	'COSMIC - Aims, Design Principles and Progress'
9:00	OLIGNY, Serge ABRAN, Alain	Full Function Points and Compatibility Issue
9:30	LOKAN, Chris ABRAN, Alain	Multiple viewpoints in functional size measurement
10:00	MIRANDA, Eduardo	Establishing Software Size Using the Paired Comparisons Method
10:30	**Break**	

Full Function Points – Industry feedback

11:00	FOLTIN, Erik DUMKE, Reiner SCHMIETENDORF, Andreas	Conceptions and Experience of Metrics Data Bases
11:30	DESHARNAIS, J.-Marc ST-PIERRE, Denis	Full Function Points: Empirical Data
12:00	**Lunch Time**	

	Panel – Workshop – COSMIC – FFP – Release 2.0	
13:30	FETCKE, Thomas	A Generalized Structure for Function Point Analysis
14:00	ABRAN, Alain OLIGNY, Serge ST-PIERRE, Denis SYMONS, Charles DESHARNAIS, J.-Marc	Preview of Release 2.0 Feedback from Field Test
14:30	**Break**	
15:00	**Discussion**	**Identification of Strengths + weaknesses**

Day 3rd September 10

Functional Size Measurement – Methods – UML Notation

8:30	STUTZKE, Richard, D.	Using UML Elements to Estimate Feature Points
9:00	LABYAD, S. FRAPPIER, M. ST-DENIS, R.	Calculs des points de fonction à partir du diagramme des cas d'utilisation de la notation UML
9:30	LÉVESQUE, Ghislain ABRAN, Alain	UML Notation for Functional Size Measurement Method
10:00	**Break**	

Functional Size Measurement: Automation

10:30	HO, Tuong, Vinh ABRAN, Alain	A Framework for Automatic Function Point Counting from Source Code
11:00	DIAB, H. FRAPPIER, M. ST-DENIS, R.	Counting Function Points from B Specifications
11:30	PATON, Keith	Requirements for Automatic Function Point Counting – Test Cases and Approach
12:00	FISCHMAN, Lee	A Function Point Counting Rules Project at Galorath
12:30	**Lunch Time**	

	Panel – Workshop – UML – COSMIC – Automation
14:00	**Identification of – Outsourcing issues** **– promising research paths**

Day 4th September 11

Extended Workshop – COSMIC

For Finalization of Details of COSMIC-FFP – Release 2.0

IMPLEMENTATION OF A METRICS DATABASE FOR INDUSTRIAL USE

Andreas Schmietendorf, Deutsche Telekom AG Berlin (Germany);
Stanimir Stoyanov and Paisii Hylendarsky, University of Plovdiv (Bulgaria);
Alexandrina Mourdjeva, University of National and World Economy, Sofia (Bulgaria)

1 Introduction

The use of metrics in the development of industrial software is gaining importance. Metrics are particularly suited to qualitative and quantitative assessment of the software development process, of the resources used in development and of the software product itself. However, software metrics can only be used effectively if the requisite measurements are integrated into the software development process and if these measurement values are taken at regular intervals. This procedure produces extensive series of measurements and thus the need for efficient measurement data management, which must include the contexts to enable discourse on the measurements that are taken as well as providing extensive evaluation options. It should also be possible to store the results of validation of a measurement as a new experience within the database.

The metricDB-2 project was executed by Deutsche Telekom's Development Center in Berlin cooperating with the University of Plovdiv and the Otto von Guericke University in Magdeburg, acting in an advisory capacity. The project focused on developing a database-supported application for management of the metrics involved in object-oriented software development. This document refers to the current development level, version 0.8 of the metricDB-2 application.

2 Prerequisites for the effective use of metrics

The objective to develop a database management system for object-oriented metrics followed a whole range of preliminary activities. It was necessary to select metrics specially for object-oriented software development from the large number of metrics proposed in the academic discourse. This was carried out using the criteria described in brief below:

Effectiveness answers the question as to the degree to which a selected metric can meet company objectives, such as reducing error rates. Here it may be suitable to apply the Goal Question Metric Method according to Basili et al.

Feature coverage - It must be possible to apply the selected metrics to as many results and software artifacts that are produced during development as possible (e.g. models, program code, documentation) and to all phases of development.

Effort minimization - Executing continual measurements within the software life cycle requires the use of measurement tools which are able to make metrics largely automatically available from measured software artifacts.

Empiric valuation - As there is usually no experience to draw on when a software metric is first introduced, it is sensible to adopt the “initial settings” (threshold values) which are suggested in the measurement tools or in the relevant literature.

Data protection and security - It must not be possible to use the metrics to draw conclusions about person-related data.

Scale features (nominal, ordinal, interval, ratio) - of the metrics in use define the information content of the data used and the statistical analysis methods and mathematical operations which can be executed via them.

Based on these features, a catalogue for object-oriented metrics was selected and defined. Currently, mainly model- and source code-related metrics from the phases of object-oriented analysis, design and implementation as well as effort estimation according to object point are included.

Another prerequisite was to consider metrics within an object-oriented VM-OO procedural model with relation to defined measurement points. The VM-OO acts as a guide to software development and comprises the phases of concept, analysis, design and evolution (implementation, integration and testing), where these may run several times within cycles. The phases themselves are divided into segments and concrete activities. The initial stage was to define measurement points at the end of each phase in relation to the cycle that was currently running. This results in 4 measurement points per cycle for the VM-OO, whereby often 3 cycles are run so that a total of 12 measurement points should normally be available in a development project.

Besides creating stable general conditions with a binding procedural model, further standardization such as defining standard development technologies and the introduction of programming standards was necessary to enable individual projects to be compared. When using metrics, it is also important to standardize the methods and tools that are to be used for software development. For example, it would not have been recommendable to introduce metrics tools for model and source code metrics before the tools used in development had been declared as standards that were binding throughout the company.

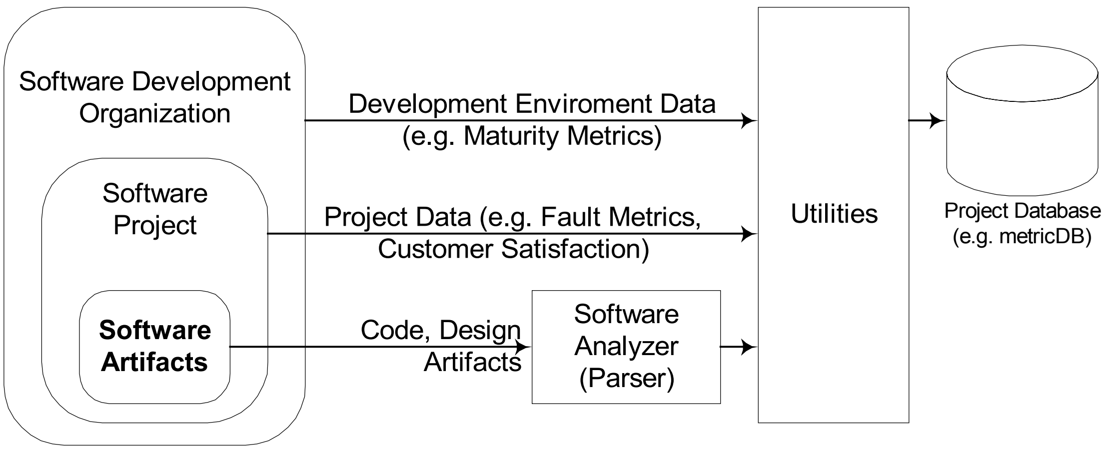


Fig. 1: Sources for a metrics database [4]

Fig. 1 shows the possible sources of software metrics according to [4]. The current version of the metrics database primarily considers software artifacts. Planning foresees extending this in the future to include metrics of software development organization, e.g. importing metrics for maturity valuation in accordance with the CMM[1] model.

3 Requirements of a metrics database

All software development projects should always have the special requirements of future users of the information system as their starting point. Additional requirements relate to the adaptability of the application to new situations, the use of different procedural models for software development and possibly also the use of different metrics sources.

As other database systems for classic metrics and experiences already exist at Deutsche Telekom, it was necessary to assume that interoperability or integration would be required in the future. The target of the metrics database is to cater for the needs of different users and, in particular, to make it easier to control the quality and cost of object-oriented software projects. In a workshop with potential customers (e.g. project managers) of the metrics database, requirements were identified and used as a starting point for data and function modeling. These requirements are summarized below:

- The effort involved in using and maintaining/administration of a metrics database must be kept to a minimum, which results in the need for extensive automation.
- It must be possible to map the procedure that is selected for a concrete software development project in the metrics database. It must also be possible to configure the created software artifacts (diagrams, documents, source texts) and to assign measurements that are taken to them.
- Serving different user types with project-specific rights. Planning currently involves application administrators (creating new projects), the project manager/developer (use of prefabricated evaluations) or academic staff who can subject the metrics to statistical analysis using external tools such as SPSS.
- Automatic problem detection in software development on the basis of exceeded, configurable threshold values in addition to offers of solution alternatives. It should be possible to store different threshold values (external, company- and project-specific experiences) in the system.
- Presentation of metrics flow and comparison with other projects by means of graphs and control diagrams. Histograms for graphical representation of, for example, effort related to the project phases.
- "Experience database" for project development and control, effort estimation, productivity/efficiency and (indirect) cost control.
- Automation of part of effort estimation (as in the present version, e.g. the object point method according to Sneed) in order to estimate effort and perform historical costing at different phases of the project.
- Checking qualitative modeling or implementation criteria by using validated metrics. Examples: maintainability, compliance with the object-oriented paradigm, stability of an object model in the face of change.

[1] CMM Capability Maturity Model of the SEI (Software Engineering Institute)

- The system must be able to incorporate new metrics and their interpretations into the database relatively easily. To this end, an internal adaptable metrics catalogue should be defined to which it must be possible to interactively map the results produced by measuring tools.
- The possibility of integrating evaluations which are not implemented on the basis of the standard functions offered by the application, such as an Excel or SPSS analysis.
- It should also be possible for users to transfer analyses to their own documents (e.g. Winword, Excel) via such mechanisms as the Windows clipboard.
- Deutsche Telekom employs several thousand software developers, who are thus potential customers of the metrics database. For this reason it was decided to implement an Internet-based client/server application to enable easy access to the information system with low administration effort.

4 Architecture and design of the application

4.1 Software architecture of the application

The main components of the application are the database server (MS SQL Server), a Web server (Internet Information Server), a Windows-based administration client and a Web client based on a standard browser (e.g. Netscape).

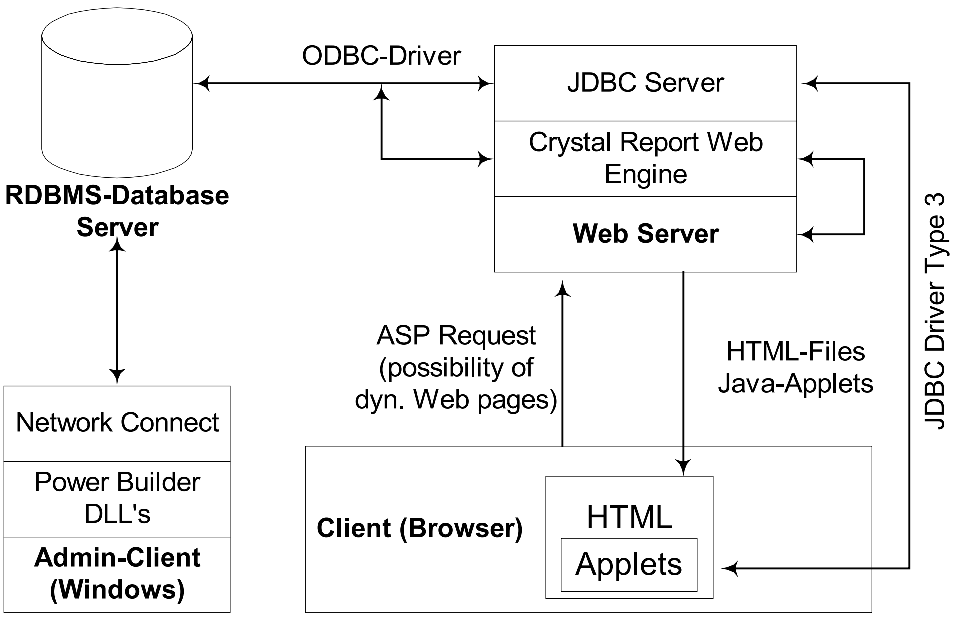


Fig. 2: Software architecture of the application

Fig. 2 shows the current software architecture of the metricDB application. The Web server contains the HTML files and the relevant Java applets from the application, which are downloaded to the Web client via HTTP. The database is accessed from the Java applet via the JDBC driver, which runs as middleware on the Web server. At the core of the application is the database, set up as a relational database management system. Despite the resulting paradigm inconsistency between the object-oriented application and database management based on the relational model, the following factors influenced the choice of system:

- Proven database management technology with extensive tool support and offering standard interfaces such as ODBC[2] and JDBC[3].
- Possibility of using the database contents under standard tools such as Excel and SPSS for statistical data analysis via the ODBC interface.
- Infrastructure for example for software distribution or appropriate centralized backup procedures is available in the company.
- To a high degree the administration client under Windows and the Web clients both execute update-intensive operations, which make the use of OLAP4 or data warehouse technologies doubtful.

Use of the type 3 JDBC driver permitted a 3-level client/server architecture to be implemented in the application on the Internet side. The administration client was linked directly to the database. Despite producing some disadvantages in terms of possible scalability, this is acceptable as it requires very few administrators in relation to Web-based users. The architecture we have implemented enables all application components to be executed on one system or, alternatively, the use of dedicated computer systems as database and Web servers. The number of administration and Web clients that can be used depends on the performance of the server systems and the load profiles caused by users. In a first step, approximately 50 Web-based users were allocated to 5 administrators, who access the database with a simultaneity factor of approx. 0.2. However, the architecture which was chosen permits far greater numbers of users.

4.2 Modeling the application

The application, in particular the database component, was modeled using Rational Rose 98. Fig. 3 shows the packages that are currently used and that contain the actual classes or entities. As most packages were described and implemented through parameterizable classes, this resulted in a highly generic data model, permitting various adjustments to be made within the application. For example, meta data is used to describe the structural elements (software artifacts) of a project, the hierarchic levels (mapping of the concrete procedural model) with for example, cycles, phases, segments, activities and the milestones in the temporal project flow (management view). On the one hand, this procedure safeguards the option of adapting to various procedural models in software development, on the other of considering various sources of metrics, related to the defined structural elements.

Inherent in a generic concept of this type is the disadvantage of increased administration effort for application operations. For this reason, the template technology was used several times, storing basic administration work, such as mapping a procedural model for object-oriented development, in the system.

---

[2] ODBC Open Database Connectivity
[3] JDBC Java Database Connectivity
[4] OLAP Online Analytical Processing

5 *Details of implementation*

5.1 Computing aggregate metrics

The computation of aggregate metrics refers at present to effort estimation according to object point. [11] The metrics which were imported via CAME tools and those which had to be input into the system manually (not measurable) were both used in computation. This is implemented technically via "store procedures" which are stored on the database server. Using these technologies offers performance advantages but has the disadvantage that it is dependent on the actual database system which is in use, here MS SQL Server.

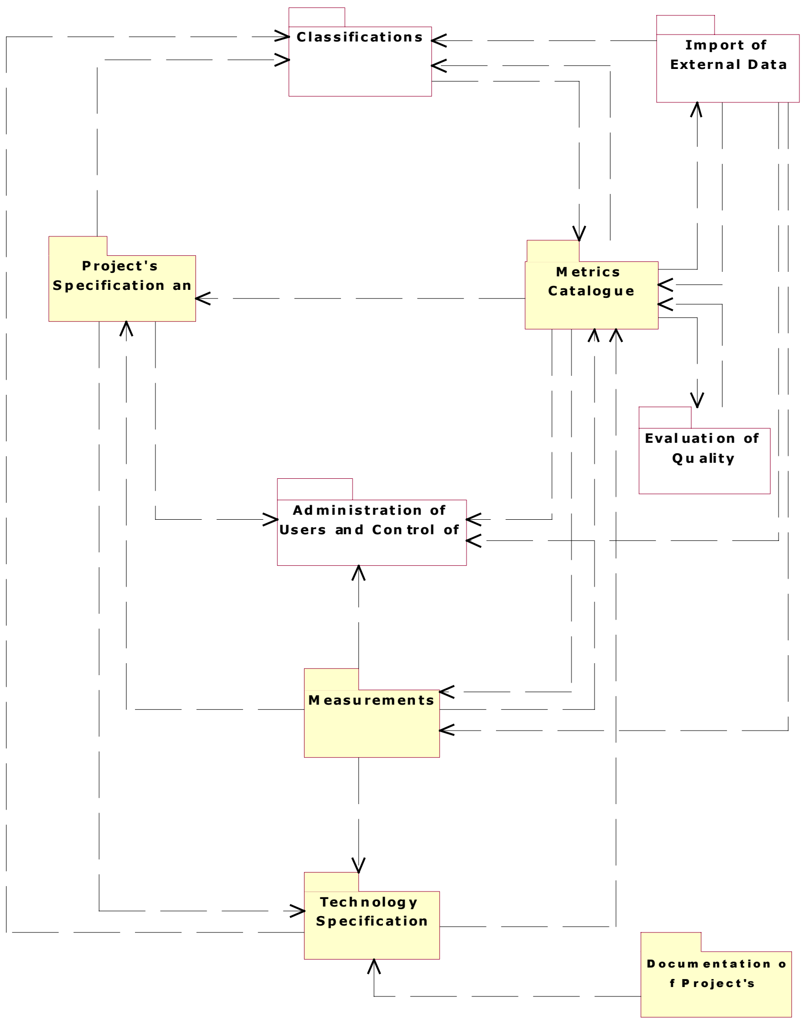


Fig. 3: Overview of the packages used in the metrics database

Within the store procedures the computation formulae for determining object points are mapped using variables so that the application administrator can assign the concrete metrics which are to be used from the metrics catalogue stored in the system. This enables the computation rule to be adapted to whatever metrics the system offers, thereby achieving independence of the concrete measurement tool in use.

For other aggregate metrics, it is possible to define new store procedures. Full disclosure of these program parts which are written in Standard SQL92 (approx. 80 effective LoC) makes this task relatively easy without the need for changes to the application itself.

5.2 Integrating the results of metrics tools

Templates in the application support the transfer of measurements performed using CAME tools. At present, the application offers templates for the data output formats of MetricsONE and RSM (Resource Standard Metrics). The functionality of the templates includes parsing the output file created by the CAME tools and writing the measurement values it reads to a temporary file. It is then possible to interactively assign (mapping) the imported measurement values to the metrics which have been mapped to the database via the defined metrics catalogue. In this way, the basic output format of the CAME tools can be retained while new versions are adapted to the metrics database by the application administrator.

In the case of MetricsONE, a comma-separated output file is created. This consists of a specification of the type of element (Package, Class, Operation, UseCase) which has been measured, the name of the metric and the actual measurement value. The sequence of datasets may vary, depending on the measurement tool settings, as it is possible, for example, not to display certain metrics; this is taken into account by the parser. The parser works on the basis of the defined keywords.

For the RSM output file, one template was developed for class structures, one for function structures and one for file structures. As the structure of the output file created by RSM (option for the metrics to be output) remains the same for each output, the template can be adapted to each output of this tool. Keywords refer to function, file and classes. In contrast to the previous tool, the actual measurement value is read out on the basis of the length format within the output file, which can also be adapted interactively.

It would only be possible to import data fully automatically if the definition of the elements administered in the metrics database were mapped to the output format of the metrics tool. The necessity of defining a standard interface for metrics tools became particularly apparent during processing. This standard should contain a generic metrics description, as well as define the grammar used inside the output file.

6 Functionality of the metrics database

6.1 Functions of the administration client

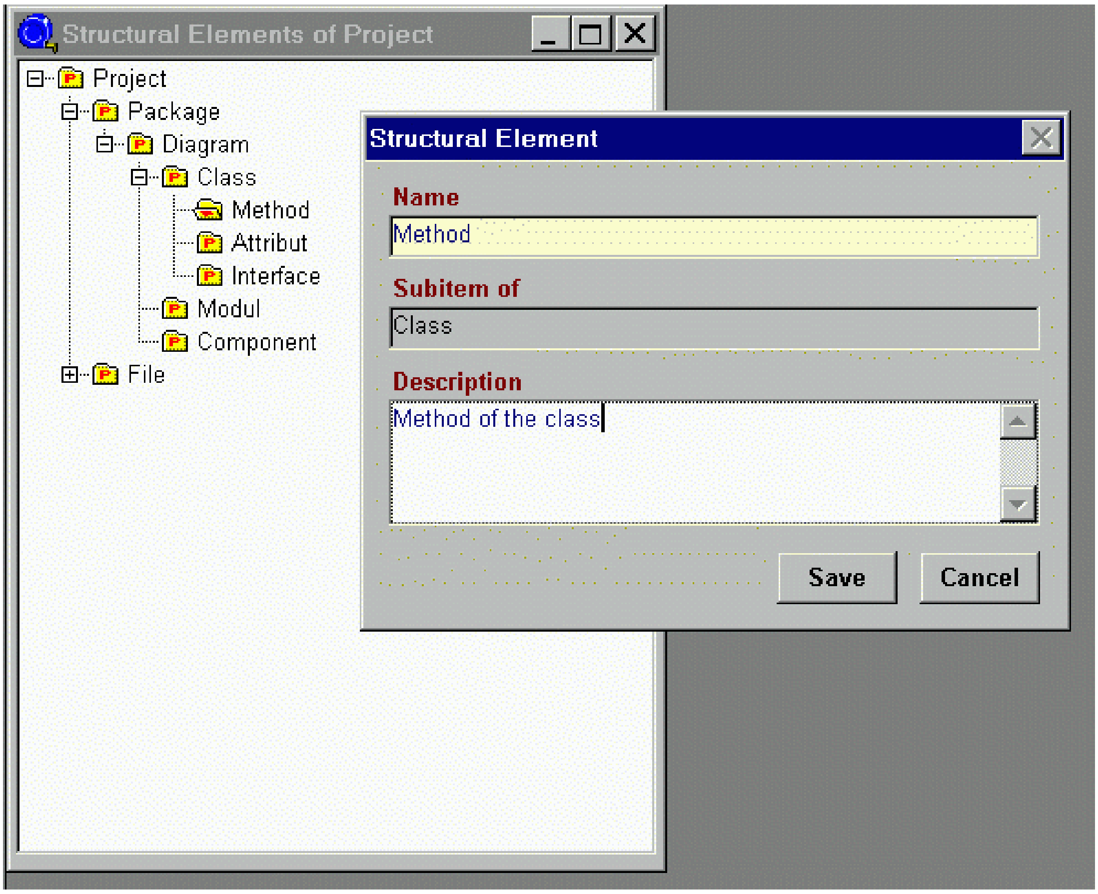


Fig. 4: Dialog for creating structural elements in a concrete project

Most of the functions described in this section are supported by templates in the metrics database, making administration easier. This means that, for example, once adjustment to a special procedural model has taken place, this template can be used for all projects which follow this procedure. Before the information system can be used for a concrete project, the application administrator must store the project structure in the database. In addition, the users who work with the application must be assigned rights for their specific projects; this is also supported by the templates.

The following describes the main functions used to set up a project:

Project Structure: Here the software artifacts used in the project are defined; they can be, for example, diagram types according to the UML notation (e.g. Packages, Classes, UseCase) or simply the source code files that are used.

Types of Methodologies: This function is used to map the concrete procedural model used (e.g. with VM-OO start, elaboration, construction, integration). A template is used to transfer this to configurations that have already been executed.

Global Stages: Management usually views projects independently of the concrete technology used. Typically, milestone plans are used, normally representing concrete tasks in sequential format. This function enables milestones to be stored in the system.

CAME tool integration: This function is used to import measurements that have been taken to the database and has already been mentioned from the viewpoint of implementation. Here, a definition of where the measurement values belong must take place. Relevant information includes the project name, the stage and cycle reached, as well as phase, date of measurement, the measurement tool used and selection of a specific template to import it into the database.

Metrics catalogue: Here, the metrics that are used are assigned to the administered project structure. New metrics can also be defined or aggregate metrics determined. The details that must be specified are the type of metric, the default tool which was used to import the metric and assignment to a structural element. Threshold values (the permitted boundaries of a metric) are also defined here. The metrics database currently contains 3 types of threshold values: A default limit which can be taken from external publications, one that has been declared binding within the company and one which can be adapted to meet the needs of specific projects. Any threshold values that are exceeded are clearly displayed at present on the Web client (red marking).

6.2 Functions of the Web client

The appearance of the application is shown in Fig. 4, the buttons on the left are used to select the requisite application function which is then made available within a Java applet.

An exception is selection of the report functionality, which creates tabular and graphics output using Crystal Reports. For user orientation, the status displays the function that is currently selected (in Fig. 4 Measurements) and the project currently being viewed (Current Project). The prerequisite for executing individual application functions is that the user has been assigned the right to do so by the application administrator.

Project Items: This function enables users to assign concrete entries to the project structures defined by the application administrator. This includes, for example, the project name, the names of the packages used, the files used and the concrete class names.

Project Implementation: The methodology stages are used here as a navigation structure, e.g. in the case of VM-OO (Start, Elaboration, Construction, Integration), and the method related to the concrete technology used (Phase, Segment,...) The start and end of the stages can be entered here, as well as cycles within a stage. When a stage is running, it is assigned a green check mark. Definition of a cycle after selecting a stage (number of cycle, start and end of the cycle and a description are possible).

Project Milestones: These correspond to the project milestones which are defined in the administrator module. They are independent of a concrete technology and correspond to those, for example, used in MS Project (mapping currently not available). Current and planned milestones for the entire project can be assigned here.

Metric Intervals: Outputs the metric intervals or threshold values used. The reports always use the "lowest level" of the defined intervals. If a project-specific threshold value has been defined, this is used, otherwise the one defined within the company or the one copied from external sources.

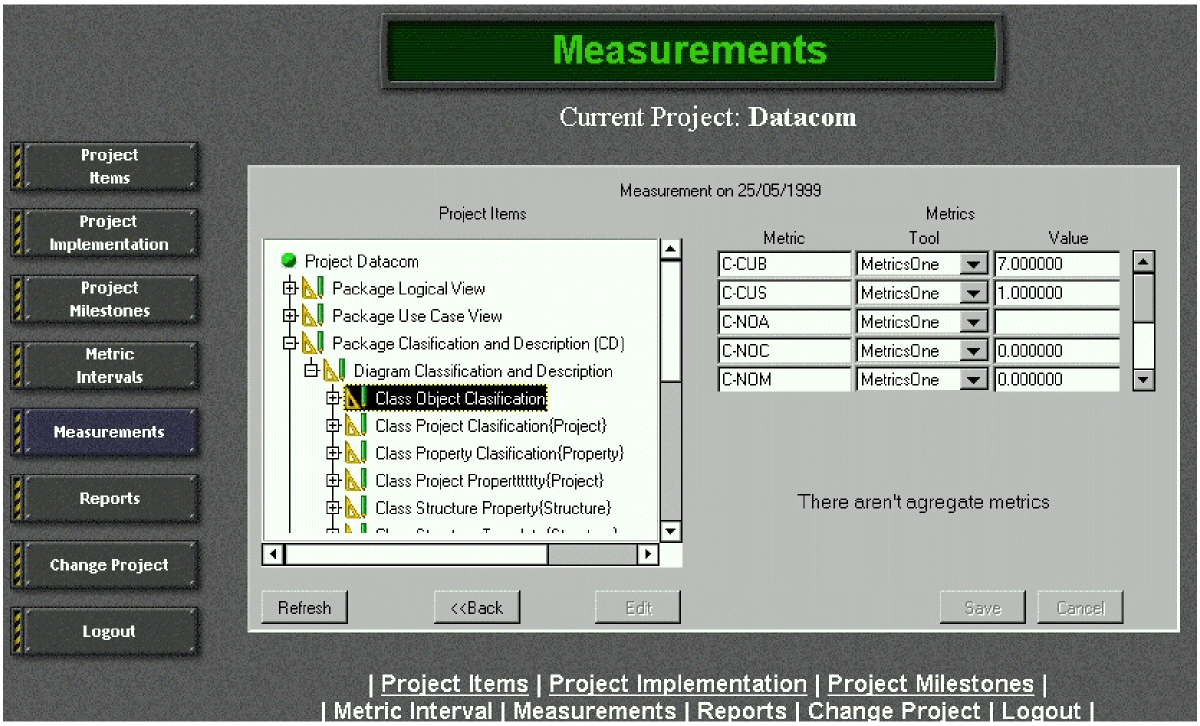


Fig. 4: User interface on the Web client

Measurements: The defined metrics catalogue is administered and the metrics are assigned to structural elements under the Admin module. Each measurement is allocated to a stage, a cycle and a phase. Thus there could be, for example, three measurements under one phase. The dialog shows which metrics it is possible to assign to a concrete measurement, how they were recorded (MetricsONE, RSM, manually – none,...) computation of the object points, whereby the previous results are deleted. The results of a calculation are also written to the database. At present, the OP computation is executed at project level, but it would make sense to include calculation for the package level, for example, in later versions. This is important when tasks in a concrete project overlap, for example in order to distribute implementation tasks over several users. Imports of measurements from result files of the CAME tools used is only possible within the administration client, as Java applets cannot access the system resources.

Reports: Support is currently provided for the following reports, which are created within an Active Server Page by the Crystal Reports used. They can be represented in either tabular or diagrammatic form. By using a filter over the period under review (start/end dates), the metrics which are to be output can be restricted. It is also possible to output the metrics and the aggregate metrics that are actually measured, and to display all metrics that are measured for one structural element (software artifact).

Statistics on elements of a project

These reports are used to output the metrics of a structural element over a defined period, in relation to global project stages, to the project cycles or to the project phases already run.

Summary of project

This type of report displays metrics in aggregate over global stages, cycles and phases of the project, with the result that relationships to individual structural elements of the project are no longer shown. The report also displays the possible intervals of a metric. If a measured value exceeds this, it is displayed in **red**, if it does not reach it, in **blue,** and in **black** if it is within the normal range.

Comparison of projects
Due to consistency rules which have to be complied with, it is only possible to compare the metrics of two projects if they have the same stage or phase structural elements. In this way it is possible to display the temporal development of metrics, such as the number of attributes defined by "public" over the phases of the project that have already run.

Additional reports
Additional reports can be used to watch the entire defined metrics catalogue (incl. defined threshold values), the status of the current project (phases and cycles of the project in relation to concrete timeframes) and the last measurement performed on a selected structural element.
The button **Project Change** is used to change to another project, and **Logoff** to exit the application.

7 Summary and Outlook

The version currently available represents a feasibility study, which cannot actually fulfil all the requirements made of a metrics database for industrial use as yet but which is still more than "just" a prototype. To summarize the results of the current version, it is a highly adaptable information system which is able to take into consideration the continually changing conditions in software development, such as new procedural models, new metrics tools and a successive increase in experiences, and thus meets the needs of investment protection. The easy-to-use Web interface makes evaluations available to a wide range of users, helping them to gain experience in the use of metrics and, implicitly, in metric validation.

The following contains a number of suggestions for suitable extensions to the metrics database which will be included in future versions:

Planning foresees extending the application by ordinal scaling metrics, for example in order to store metrics which are imported through CMM or bootstrap evaluation in the database. This will also make it possible to check the effects of a high degree of process maturity in software development against product-related features over a long period of time.

The text and graphic reports that are currently restricted to a comparison of two projects will be extended to cover more than two projects. Replacement of the currently used Crystal Reports by Java class libraries is being considered. A drill-down analysis is also to be offered; this provides information at a very abstract level (management view) on the status of a project and its phases and permits "drilling down" to elements which measurements have shown to be critical. In particular, a more powerful interpretation of threshold values that are exceeded (empirical evaluation through long-term analysis) is to be added to the available reports, enabling, for example, suggestions for improving critical situations to be given.
Planning also envisages integrating measurement results provided by the McCabe measurement tool, direct operation of the Rational Rose script-based interface, which will

make it possible to freely define metrics in relation to the results of object-oriented analysis and design, as well as to import process metrics in the shape of a bootstrap evaluation. Investigations are to show to what degree the milestone plans defined manually in the database can be imported directly into a management tool such as MS Project.

Some interesting enhancements involve the possible replacement of the current database vendor-dependent “store procedures” by an application server, enabling other database management systems to be used, an increase in the supported aggregate metrics and the already planned multilingual version in the German, English and French languages.

8 References

[1] **Backhaus:**
Multivariante Analysemethoden - Eine anwendungsorientierte Einführung.
Springer-Verlag, Berlin 1996

[2] **Basili, V.:**
Software modeling and measurement: The Goal/Question/Metric paradigm.
Technical Report CS-TR-2956, Dept. of Computer Science, University of Maryland, September 1992

[3] **Dimitrov, Evgeni:**
Ein Vorgehensmodell für die Objektorientierte Entwicklung.
Internal study, Deutsche Telekom, Berlin: 1998

[4] **Evanco W.M., Lacovara R.:**
A model-based framework for the integration of software metrics.
Journal of Systems and Software 26, 77-86, 1994

[5] **Foltin E., Dumke R.:**
Aspects of Software Metrics Database Design.
Software Process Improvement and Practice 4, 33-42, 1998

[6] **Gudlat D.:**
Konzeption und prototypische Implementation einer Datanbankanwendung für die Steuerung objektorientierter Softwareentwicklungsprojekte mit Hilfe von Softwaremeßdaten.
Diploma dissertation at the Humboldt University Berlin, 1999

[7] *Funktionale Beschreibung MetricDB*, Deutsche Telekom AG, Development Center Berlin

[8] *NumberSIX*, MetricsONE User’s Guide Version 1.0, Washington 1997, URL: http://www.numbersix.com

[9] *Ressource Standard Metrics Version 4.0 for Windows NT*, M Squared Technologies, URL: http://www.tqnet.com/m2tech/rsm.htm

[10] **Schmietendorf A.:**
Prototype of a Software Metrics Database for Industrial Use.
In Dumke/Abran: Software Metrics. DUV, 1999, 229-243

[11] **Sneed, Harry M.:**
Schätzung der Entwicklungskosten von objektorientierter Software.
Informations Spektrum 19: 133-140, Heidelberg: Springer Verlag, 1996

Dear Reader! We will start a series of articles about metrics and measurement tools. In the last years, experience has been gained in both industrial practice and in academic measurement laboratories. We encourage you to present your opinion or experience in metrics tool applications in our Metrics News. We begin with a short overview about some aspects of metrics tool applications at the SMLab of the University of Magdeburg.

METRICS TOOLS – AN OVERVIEW

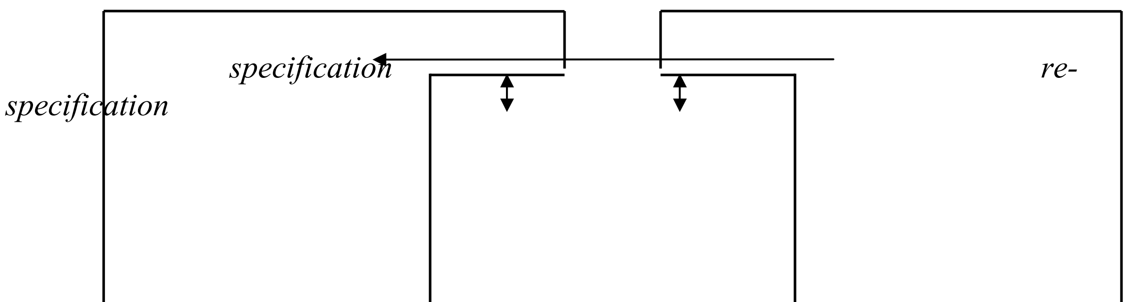
Reiner Dumke, University of Magdeburg

Abstract

This paper describes the role of the metrics tools as ***C****omputer* ***A****ssisted Software* ***M****easurement and* ***E****valuation (****CAME****) tools in the software life cycle. The most CAME tools are designed for code analysis and measurement. They are predestined to be applied to the implementation and maintenance development phases. But, more and more tools are developed for the earlier phases of software development to estimate the effort, complexity, and size of the software that will be created. This paper will provide an overview of the present situation on the area of the CAME tools and discuss their efficient use in the software maintenance.*

1 Introduction

CAME tools are tools for modelling and determinating the metrics of software development components referring to the process, the product and the resource. Presently, the CAME tool area also includes the tools for model-based software components analysis, metrics application, presentation of measurement results, statistical analysis and evaluation. In general, we can establish CAME tools for classification, for component measurement, for process or product measurement and evaluation, as well as for training in software measurement. The application of CAME tools is based on the given measurement framework (see [2], [3], [4], [7], [12], [13], [17], [18], [19], [20], [24], [25], [28] and [29]). The integration of CAME tools in the tool supports in the software engineering cycle is given in the Figure 1. On the other hand CAME tools can be classified according to the degree of integration in software development environments such as integrated forms, external coupling forms and stand-alone metrics tools.



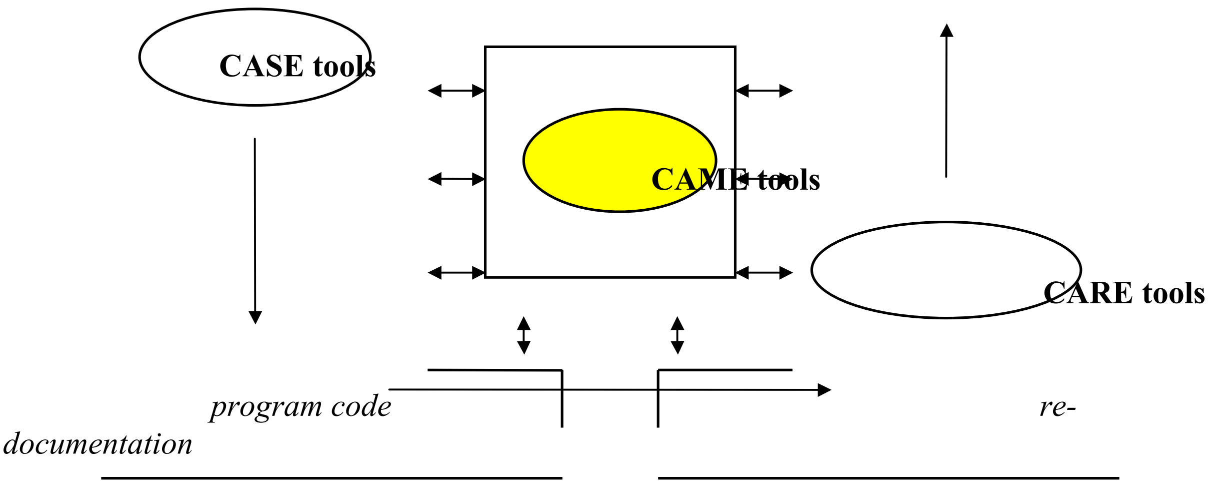


Fig. 1: The general CAME tool integration

The existing CAME tools in the Software Measurement Laboratory at the University of Magdeburg are given in the following table ([8], [11], [26], [27]).

Process Measurement and Evaluation:

- SynQuest *(Bootstrap)* ***Switzerland***
- NEXTRA *(gen. classification)* ***USA***
- Ami Tool *(GQM, CMM)* ***France***
- SPQR/20 *(Jones Experience, Function Points)* ***USA***
- Knowledge Plan, ***USA***
- SOFT-ORG *(general organizational model)* ***Germany***
- SQUID M-Base *(COQUAMO)* ***U.K.***

Product Measurement and Evaluation:

- Requirement Analysis and Specification:
  - PDM* *(HTML Text)* ***Magdeburg***
  - RMS *(Documentation)* ***Germany***
  - Function Point Workbench ***Australia***
  - FPTOOL (Demo) ***Germany***
  - CHECKPOINT (Demo) ***USA***
  - SOFT-CALC *(Function Point, Data Point, Object Point)* ***Germany***
  - SVS* *(McCall Model)* ***Magdeburg***
  - OOM* *(OOA/OOD Models of Coad/ Yourdon)* ***Magdeburg***
  - COSMOS (*Lotos) **Netherlands***
- Software Design:
  - MOOD (Demo, *C++)* ***Portugal***
  - DEMETER (Demo) ***USA***
  - SmallCritic *(Smalltalk)* ***Germany***
- Program Evaluation:
  - MCOMP* *(Modula Subset)* ***Magdeburg***
  - CodeCheck *(C, C++)* ***USA***
  - SOFT-AUDITOR *(Cobol)* ***Germany***
  - QUALIGRAPH (Demo) ***Hungary***
  - MPP* *(C++)* ***Magdeburg***
  - OOMetric (Demo) ***USA***
  - QUALMS *(Fortran, Pascal, C)* ***U.K.***
  - ProVista *(C)* ***Germany***
  - DATRIX *(C)* ***Canada***
  - LOGISCOPE *(C, C++)* ***USA***
  - COSMOS *(C)* ***Netherlands***
  - PMTool* *(C++)* ***Magdeburg***
  - PC-METRIC *(Fortran, C, Pascal)* ***USA***
  - PMT* *(Prolog)* ***Magdeburg***
  - MJAVA* *(Java)* ***Magdeburg***
- Software Test:
  - IDAS-TESTDAT *(C)* ***Germany***
  - LDRA Testbed *(C++)* ***U.K.***
  - STW-METRIC *(C++)* ***USA***
- Software Maintenance:
  - Smalltalk Measure* ***Magdeburg***

Resource Measurement and Evaluation:

- Productivity:
  - SPQR/20 ***USA***
  - and the other above
- Performance:
  - Foundation Manager ***USA***
  - SunNet Manager ***USA***

Measurement Presentation and Statistical Analysis:

- SOFT-MESS ***Germany***
- EXCEL ***USA***
- SPSS ***USA***

Software Measurement Training:

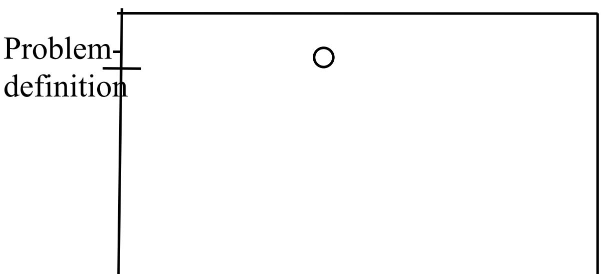
- METKIT ***U.K.***
- MIS ***Germany***

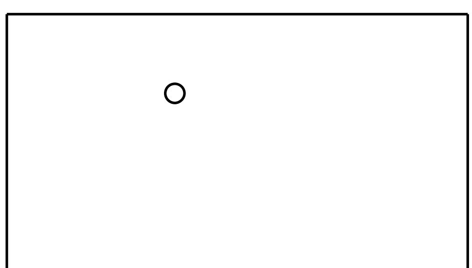
Tab. 1: CAME tools applied at the University of Magdeburg

The CAME tools should be embedded in a software measurement framework which includes the software process characteristics and their assessment and controlling. In general, we can distinguish two kinds of frameworks: the *informal* and the *formal*. ***Informal approaches*** of software measurement frameworks consist of the general components of textual descriptions/questions, rules and 'laws', experience notices, and standards. Examples of these measurement frameworks are ([1], [10], [12], [13], [17], [28]) the *ISO 9000-3 standard,* the *Capability Maturity Model* (CMM) (SEI in Pittsburgh), the *Goal Question Metric* paradigm (GQM) (University of Maryland), the *Software Quality Metrics* report (FAA Technical Center, New Jersey), the *TickIT approach* (UK), the *BOOTSTRAP quality standard* (ESI, European Esprit project), the *Software Measurement Guidebook* (NASA), the *Trillium standard* (Bell Canada), the *AQAP* and the *DOD STD 2167A* (USA military area), the European *SPICE project* etc. The situation in the informal approaches can be characterized by a 'break' in the measurement methodology between the measurement goals and the (selected) metrics. ***Formal approaches*** for the software measurement frameworks can be divided in *algebraic approaches, axiomatic approaches, functional approaches,* and *rule-based approaches* ([8], [12], [13], [25], [29]) and we can establish the situation no independence of the development paradigm, only a few practicable results or few empirical evaluations, and only a few empirical evaluation. The underlying rules in software (quality) measurement frameworks are for example in the NASA Guidebook [24]: establishing a *measurement program* (including the definition of the goals, the responsibilities and selecting the measures), core measures (especially the costs, errors, process characteristics, project dynamics, and project characteristics), operation of the measurement program (with the use of metrics tools, storage the measurement values etc.), analysis, application, and feedback (as goal of the (software process or product) improvement. Examples of the different measurement strategies and frameworks are the Bang metric for the Structured Analysis development method as model-based measurement and the CMM as evaluation.

2 The current situation of CAME tools

In the following, we present some examples of CAME tools with the different possibilities of model based presentation, metrics execution, component evaluation, and measurement education (see [6], [8], [15], [16], [21], and [29]). The following two diagrams characterize the current situation in short, concise, and very simplified form [6].





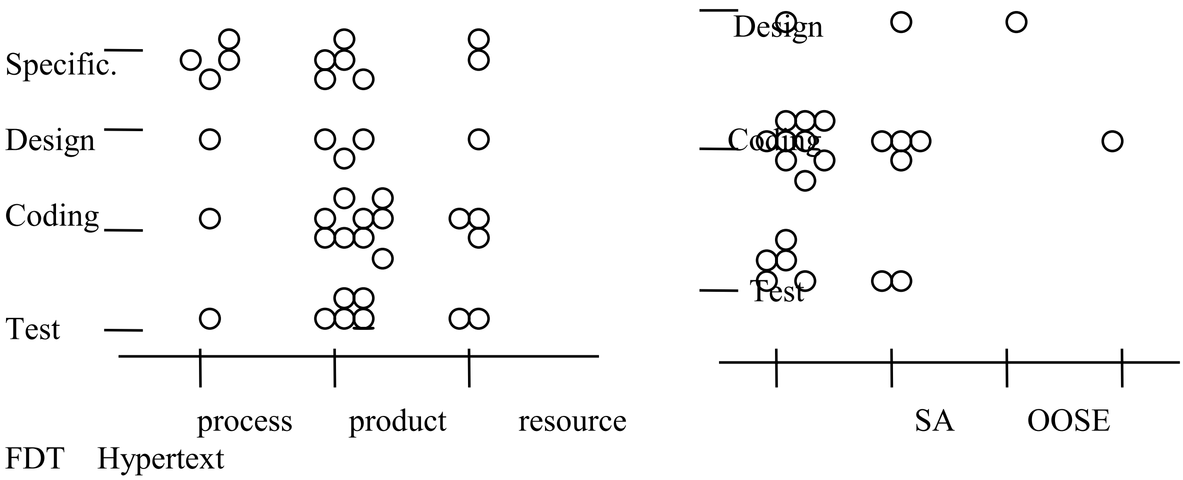


Fig. 2: CAME tool situation

The left diagram describes the CAME tool situation in the software development phases for the process, product and resource evaluation, wheras, the diagram on the right side describes the relations to the different programming/development paradigms (SA - Structured Analysis, OOSE - object-oriented, and FDT - formal description techniques). Figure 4 describes the situation of selected CAME tools corresponding to the different measurement aspects of flexibility and openness. The flexibility is necessary to manage the software development complexity in the company, and it allows to define new metrics in the tool or to change the empirical evaluation criteria. Openness is required to use any components of the CAME tool for the applied measurement framework, such as the modelling part, the metrics execution part or the statistical analysis and the presentation.

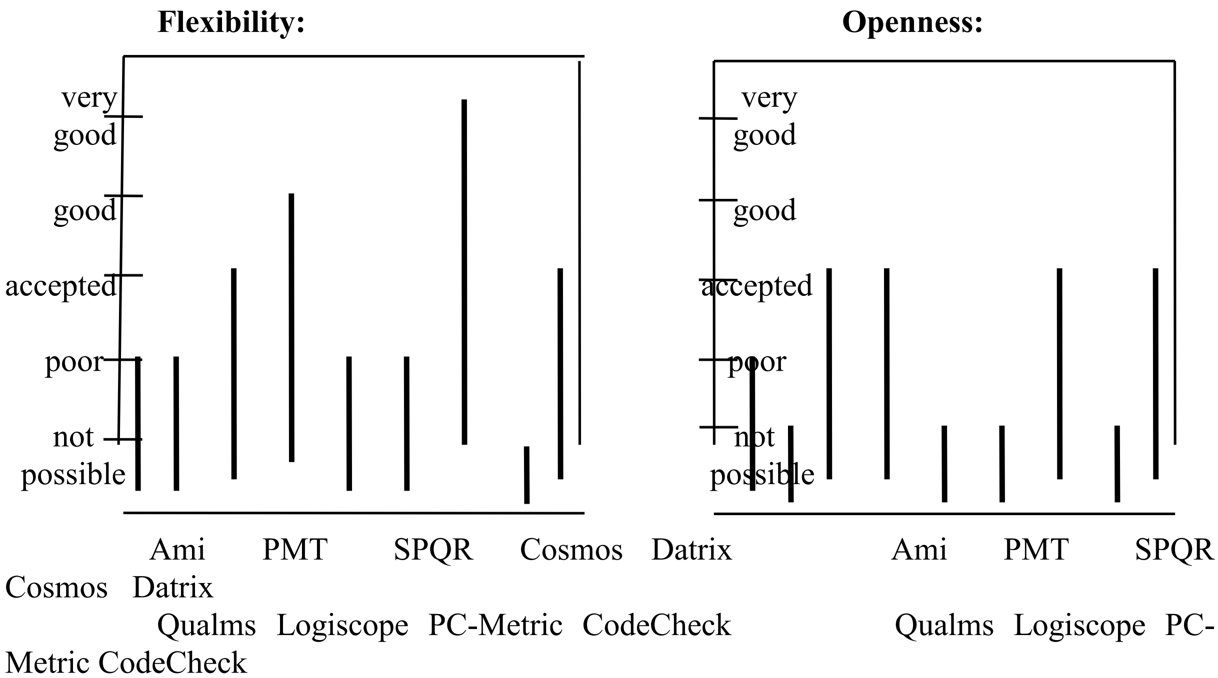


Fig. 3: ***Flexibility and openness of CAME tools***

Obviously, a lack of openess of the analyzed tools can be observed. Figure 5 shows how useful the evaluation of the selected CAME tools is.

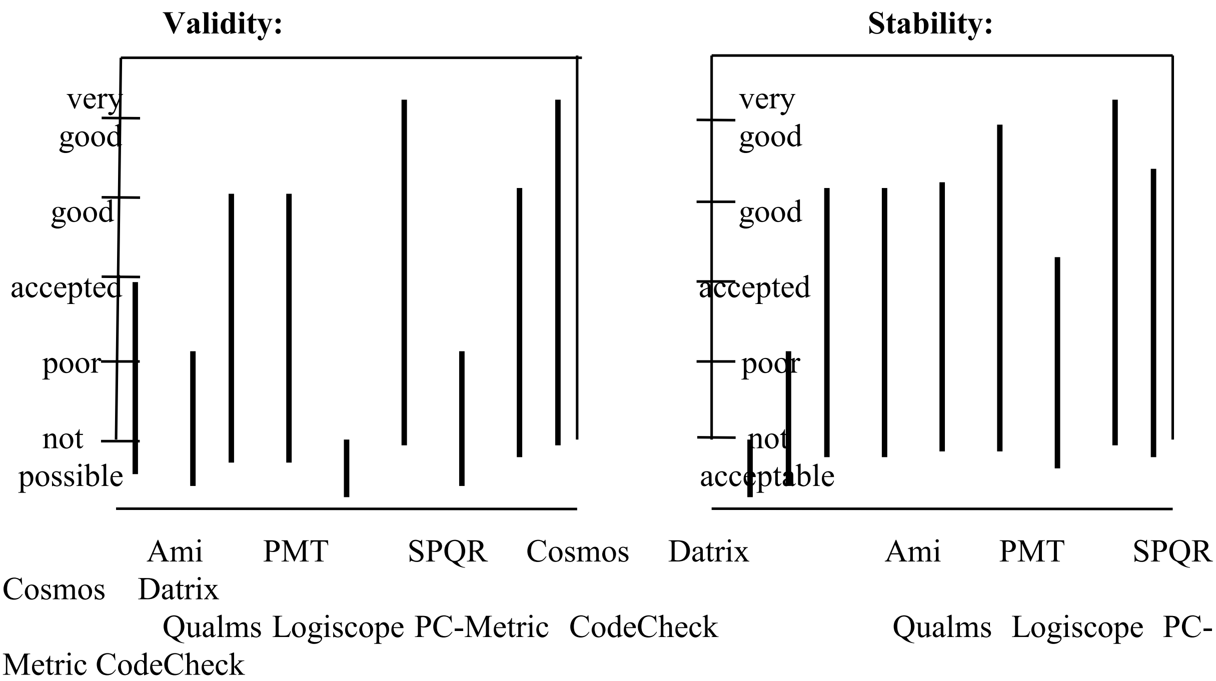
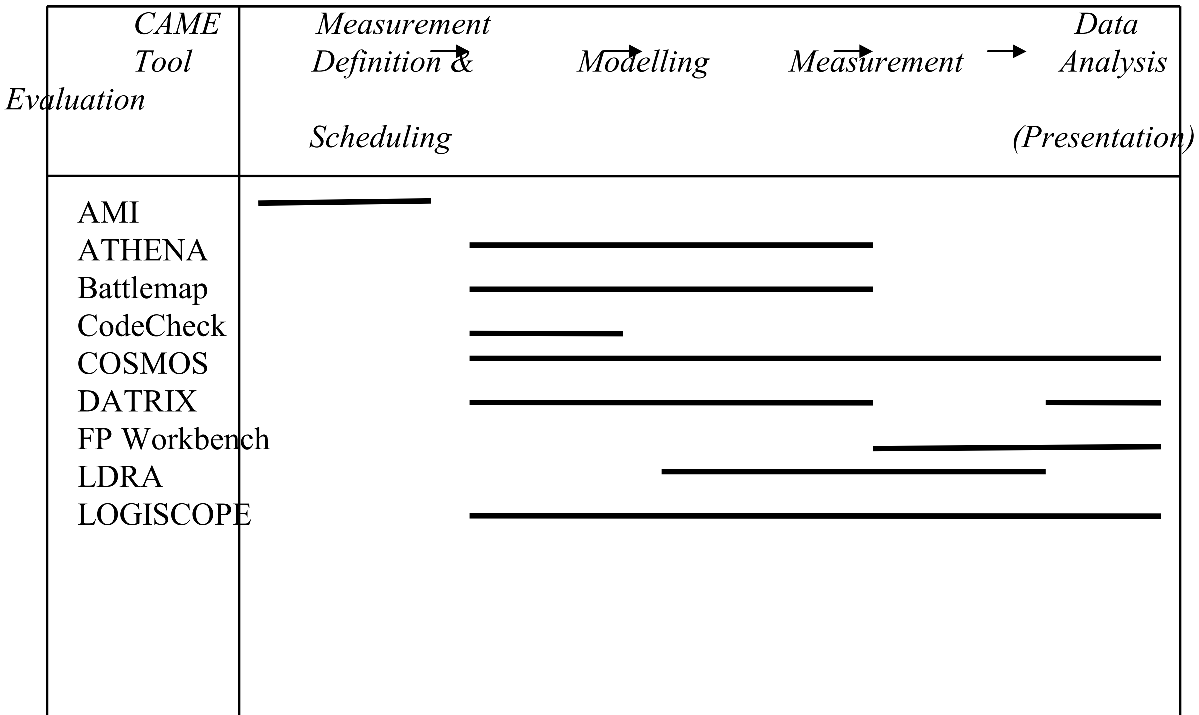
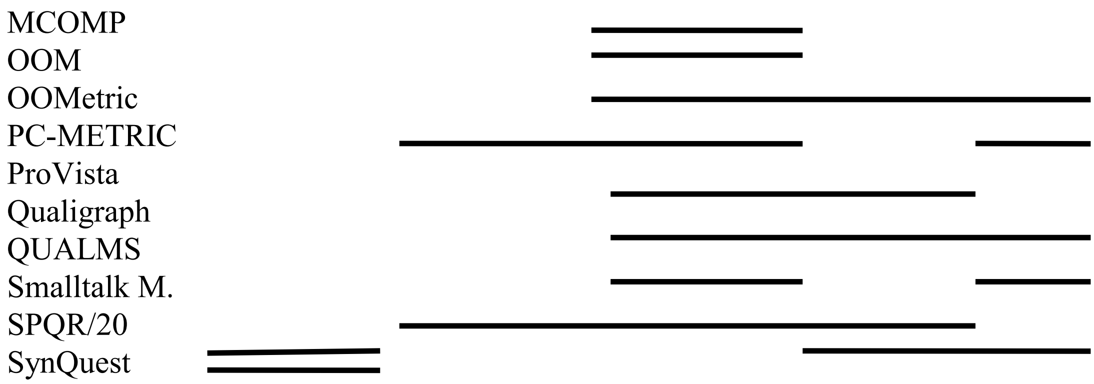


Fig. 4: Validity and stability of the CAME tools

The usefulness is divided in validity and stability of the tool application. The validity is the characterization of the correctness of the metrics values and the reconstructability of the results. The stability corresponds to the reliability of the tool at run-time. You can see in ***http://ivs.cs.uni-magdeburg.de/sw-eng/us/*** a short description of these tools in the World-Wide Web in our Measurement Laboratory Interface. We can establish the situation in the software measurement strategies as more and more consensus about the measurement areas (as process, product and resources) and the empirical goals for the evaluation and the assessment. But the goal must be a measurement-based controlling cycle. CAME tools are mostly directed to special areas of the measurement phases. The following table gives a brief idea of this situation.





Tab. 2: **CAME Tools in the measurement phases**

Some general CAME tool application problems and the necessity of standardization of the interfaces between the modelling, measurement, data analysis, and evaluation are described in the following:

- in *modelling*: the visualization of the software development components (especially standardization of flow graphs, call graphs, diagrams etc.) including modern visualization facilities;
- in *measurement:* modification of the IEEE standard to a standard measurement input and a standard measurement output and the extension with metrics definitions in a "workflow" manner;
- in *data analysis:* classification and standardization of the measurement output for the use of the existing statistical methods;
- in *evaluation:* the application of the experience in "classical" software development methodologies in new development paradigms.

The standardization must be considered the high dynamic in the software measurement process and the measurement objects itself.

Other criteria for an efficient use of CAME tools are given in the following points.

- The efficiency of CAME tool application depends on the "well-definedness" of the software process or product themselves.
- CAME tools require in general a good structuring of the measurement area or components: this can be also established as a (first) improvement aspect.
- The main problem in the application of CAME tools is to establish a really measurement-controlled cycle, like in the controlling engineering.

For the use of different CAME tools it is necessary to implement a measurement data base.

4 Conclusions

This paper has summarized the current situation and the problems of the efficiency of use of the CAME tools. One of the goals was also to present an overview of the present situation of the software measurement. In order to use CAME tools efficiently, some rules should be kept in mind:

- The present CAME tools are no suitable means of complex software evaluation. They are mostly based on existing assessment methodologies such as the Function Point method. The applied metrics must be algorithmic.
- The selection of a software metric tool should be influenced by the following considerations:
  - The tool should be designed specifically for the respective software/hardware platform.
  - The philosophy of the CAME tool should be applied consequently. The tool-specific conception of modelling, presentation and metrics evaluation should not be violated.

- Both hardware and software platforms are subject to a highly dynamic development process.

- Specific parameters of the software development environment should be known to ensure correct and complete input information for the CAME tool. A profound analysis of the empirical aspects such as effort and costs is an imperative precondition for the proper use of any selected CAME tool *(for the* ***right use*** *of the* ***right metrics tool).***

Further investigations are directed on the measurement and evaluation of the CAME tools themselves to improve the different parts of the software development in a quantitative manner. The current situation of the CAME tools prefers the start in the code evaluation in the implementation or in the maintenance phase for reengineering. But, it is only an assessment to involve the other aspects of the process and product measurement and improvement.

5 References

[1] **Ami** (application of metrics in industry) - *a quantitative approach to software management.* Handbook, CSSR, London, 1993

[2] **Arnold , R.S.:**
Software Reengineering. IEEE Computer Society Press, 1994

[3] **Arthur , L.J.:**
Improving Software Quality - An Insider's Guide to TQM. John Wiley & Sons, 1993

[4] **Basili, V.R.; Selby, R.W.; Hutchens, D.H.:**
Experimentation in Software Engineering. IEEE Transactions on Software Engineering, 12(1986)7, pp. 733-743

[5] **Boehm , B.W.:**
Software Risk Management. IEEE Computer Society Press, 1989

[6] **Dumke, R.:**
CAME Tools - Lessons Learned. Proc. of the Fourth International Symposium on Assessment of Software Tools, May 22-24, Toronto, 1996, pp. 113-114

[7] **Dumke, R.; Foltin, E.; Koeppe, R.; Winkler, A.:**
Measurement-Based Object-Oriented Software Development of the Software Project "Software Measurement Laboratory". Preprint Nr. 6, 1996, University of Magdeburg (40 p.)

[8] **Dumke, R.; Foltin, E.; Koeppe, R.; Winkler, A.:**
Softwarequalität durch Meßtools. Vieweg Publ., 1996

[9] **Dumke, R.; Pinkert,K.:**
Measurement and Evaluation of LINUX Components with the COSMOS Measurement Tool. SMLAB Report, 002/96

[10] **Dumke, R.; Winkler, A.:**
Object-Oriented Software Measurement in an OOSE Paradigm. Proc. of the Spring IFPUG'96, February 7-9, Rome, Italy, 1996

[11] **Dumke, R.; Winkler, A.; Zbrog, F.:**
Metrics in the Hypertext and Hypermedia Software Development. (German) Research Reports in Computer Science 1995-25, Technical University of Berlin, 1995, pp.121-127

[12] **Ebert, C.; Dumke, R.:**
Software-Metriken in der Praxis. Springer Publ., 1996

[13] **Fenton , N.:**

Software Metrics - a rigorous approach. Chapman & Hall, 1991

[14] **Fix, A.:**
Conception and Implementation of a Measurement Data Base for Distributed Use. Diploma Thesis, University of Magdeburg, July 1996

[15] **Grigoleit, H.:**
CAME Tools - An Overview. in: http://irb.cs.uni-magdeburg.de/se/metrics_eng.html

[16] **Heckendorff, R.:**
The Smalltalk Measure Browser. Study, University of Magdeburg, June 1996

[17] **Henderson-Sellers, B.:**
Object-Oriented Metrics - Measures of Complexity. Prentice Hall Inc., 1996

[18] **Jones , C.:**
Assessment and Control of Software Risks. Yourdon Press, 1994

[19] **Kan, S.H.:**
Metrics and Models in Software Quality Engineering. Addison-Wesley Publ., 1995

[20] **Kitchenham, B.:**
Software Metrics. Blackwell Publ., Cambridge, Mass., 1996

[21] **Kompf, G.:**
Conception and Implementation of a Prolog Measurement and Evaluation Tool.(German) Diploma Thesis, University of Magdeburg, July 1996

[22] **Lorenz, M.; Kidd, J.:**
Object-Oriented Software Metrics. Prentice Hall Inc., 1994

[23] **Lubahn, D.:**
The Conception and Implementation of an C++ Measurement Tool.(German) Diploma Thesis, University of Magdeburg, March 1996

[24] **NASA:**
Software Measurement Guidebook, Maryland 1995

[25] **Oman, P.; Pfleeger, S.L.:**
Applying Software Metrics. IEEE Computer Society Press, 1997

[26] **Prange, J.:**
Conception and Implementation of a HTML document metrics tool. (German) Study, University of Magdeburg, January 1996

[27] **Rudolph, T.:**
Implementation of a JAVA metrics tool. (German) Study, University of Magdeburg, August 1996

[28] **Software Productivity Consortium:**
The Software Measurement Guidebook. Thomson Computer Press, 1995

[29] **Zuse, H.:**
A Framework of Software Measurement. to be published

CALL FOR REVIEWERS

Guide to the Software Engineering Body of Knowledge Project - SWEBOK Second Review Cycle: June 21 to September 24, 1999 Third review Cycloe: October (see the SWEBOK web site for the exact schedule).

The IEEE Computer Society actively promotes software engineering as a profession and a legitimate engineering discipline notably through its involvement in the Joint ACM-IEEE Computer Society Software Engineering Coordinating Committee (see www.computer.org/tab/swecc/). This committee aims to foster and maintain software engineering as a professional computing discipline. The current chair of this committee is Leonard Tripp, 1999 President of the IEEE Computer Society. Gathering consensus by the profession on a core body of knowledge is a key milestone in all disciplines and has been identified as crucial for moving software engineering toward a professional status.

The purpose of the Guide to the Software Engineering Body of Knowledge (see www.swebok.org) is therefore to:

- characterize the contents of the Software Engineering Body of Knowledge;
- provide a topical access to the Software Engineering Body of Knowledge;
- promote a consistent view of software engineering worldwide;
- clarify the place of, and set the boundary of, software engineering with respect to other disciplines such as computer science, project management, computer engineering and mathematics;
- provide a foundation for curriculum development and for individual certification and licensing material.

In 1998, a Straw Man version of the Guide was written to define the project's strategy and rationale, to gather momentum in the profession and to jump start the Stone Man phase by proposing a draft list of Knowledge Areas of software engineering and a draft list of Related Disciplines.

Based on the results of this first phase, a Stone Man version is currently being developed with the corporate support of the ACM, Boeing, Comerica, the IEEE Computer Society, the National Institute of Standards and Technology, the National Research Council of Canada, Raytheon Company and SAP Labs (Canada). The project is managed by the Universite du Quebec a Montreal. All final and intermediate deliverables of this project are or will be available free at www.swebok.org.

The specific deliverables of the Stone Man version are a:

- Consensus on a list of Knowledge Areas;
- Consensus on a list of topics and relevant reference materials for each Knowledge Area;
- Consensus on a list of Related Disciplines;

To be successful, the Guide to the Software Engineering Body of Knowledge project requires the contribution of a large number of people. An underlying principle of this project is consensus-building within the international software engineering community, which of course implies a large number, and a wide spectrum, of contributors.
We are currently seeking Reviewers and Review Captains for the following Knowledge Areas:

Software Design
Software Engineering Infrastructure
Software Engineering Management
Software Quality Analysis
Software Requirements Analysis
Software Testing
Software Construction
Software Engineering Process
Software Evolution and Maintenance

Reviewers

Reviewers are responsible for:

- Reading the Knowledge Area Description and consulting the selected reference material
- Providing comments from one specified viewpoint

The criteria for selecting Reviewers are:

- Knowledge in the Area;
- Availability;
- Ability to give articulate, constructive comments;
- Representative of one of the viewpoints that has been identified:
  individual practitioners,
  trainers and educators, standards developers, regulators, etc.

Schedule

The plan is that the Reviewers will be called upon to contribute in the second review cycle in June and July, 1999. A third review cycle is currently scheduled for the October 1999 time frame. To sign up as a Reviewer, please complete and submit the electronic form available at www.swebok.org.

For further information, please visit www.swebok.org or contact
Robert Dupuis (robert.dupuis@uqam.ca) or
Pierre Bourque (pierre.bourque@uqam.ca).

All Reviewers will be recognized by having their name on the list of contributors.
Contact address:

ALAN ABRAN
Professor and Director of the Research Lab. in Software Engineering Management
Quebec-University of Montreal
Departement of Computer Science
C.P. 8888 Succursale Centre-Ville, Montreal, H3C 3P8, Canada
Tel.: *+1-514-987-3000, -89000,* ***Fax:*** *+1-514-987-8477*
Email: *abran.alain@uqam.ca*

Some of the Quality Initiatives in Germany:

- Arbeitskreis Software-Qualität Franken e.V.

  see: http://www.asqf.de/

- Gesellschaft für Softwarequalitätssicherung GmbH (SQS)

  see: http://www.sqs.de/

- Fraunhofer Institut Experimentelles Software Engineering (IESE)

see: http://www.iese.fhg.de/

- Deutsche Informatik Akademie Seminare:
  - Strategien zur Verbesserung des Softwareentwicklungsprozesses (Dr. C. Ebert)
  - Methodisches Testen und Analysieren von Software (Dr. P. Liggesmeyer)
  - Software-Projektsteuerung für die Praxis (Metriken für Projektmanagement, Qualitäts- und Prozeßverbesserung) (Dr. C. Ebert)

see: http://www.gi-ev.de/dia/

- Deutschsprachige Anwendergruppe für Softwaremetriken und Aufwandsschätzung (DASMA)

see: http://www.dasma.de/

- European TeleCASE Center (ETC) in Braunschweig

see: http://www.dlr.de/bs_d.html

- Das Software Meßlabor (SMLAB) der Universität Magdeburg

see: http://ivs.cs.uni-magdeburg.de/sw-eng/us/

Dumke, R.; Foltin, E.; Koeppe, R.; Winkler, A.: Softwarequalität durch Meßtools - Assessment, Messung und instrumentierte ISO 9000

Vieweg Publ., 1996 (223 p.) *ISBN 3-528-005527-8*

This book gives an overview about the software metrics tools for all phases of the software development process. The metrics tools are defined as CAME tools (Computer Assisted Software Measurement and Evaluation). The introduction describes the essential aspects of the software measurement. The description oft the CAME tools includes the cost estimation tools, Capability Maturity Model evaluation tools, metrics tool for software specification, design and code, and tools forthe software testing and maintenance (including network performance). Some tables helps to decision of choosing the useful tool integration for the software quality assurance process.

Zuse, H.: A Framework of Software Measurement

de Gruyter Publ., Berlin New York, 1997 (755 p.) *ISBN 3-11-015587-7*

This book describes a framework for software measurement from a theoretical, practical and educational view. The main idea is the application of the measurement theory on the area of software measurement.
The book is written in nine chapters and includes exercises for a teaching in software measurement. The chapters describe the software measurement aspect, the history of software measurement, the theoretical foundations from theoretical and practical view, especially the object-oriented software measures, the discussion about the properties and validation, and helpful remarks for a successful application of software measures.
The book includes a CD ROM that include a demo tool for software measurement education based on more than thousand references and metrics.

Dumke, R.; Abran, A.: Software Measurement – Current Trends in Research and Practice

DUV Publisher, Wiesbaden, 1999 (269 p.) *ISBN 3-8244-6876-X*

This new book includes key papers presented at the 8th International Workshop on Software Metrics in Magdeburg (Germany), September 1998. It is a collection of theoretical studies in the field of software measurement as well as experience reports on the application of software metrics in USA, Canadian, Netherlands, Belgian, France, England and German companies and universities. Some of these papers and reports describe new software measurement applications and paradigms for knowledge-based techniques, test service evaluation, factor analysis discussions and neural-fuzzy applications. Other address the multimedia systems and discuss the application of the Function Point approach for real-time systems, the evaluation of Y2K metrics, or they include experience reports about the implementation of measurement programs in industrial environments.

Solingen, R. van; Berghout, E.: The Goal/Question/Metric Method: a practical guide for quality improvement of software development

McGraw-Hill International (UK), 1999 (199 p.) *ISBN 0-07-709553-7*

The subject of software quality management is of the greatest importance to all organizations. Those responsible for the IT operation need to guarantee not only the highest quality, but also the highest quality within constraints of timeliness, functionality, complexity, and cost.
The Goal/Question/Metric method (GQM) is designed to support such a quality improvement approach by helping practitioners to achieve their explicitly stated objectives. GQM is the next logical step after ISO9000 certification or CMM level improvements. This book has been written to support people working an quality improvement in software development. Its use of GQM provides:

- motives to start goal-oriented measurement;
- detailed steps to take in applying GQM;
- examples of possible support for GQM;
- templates for the necessary deliverables for applying GQM;
- results from applications implemented by the Authors;
- suggestions for feedback material in which data is presented from projects carried out by the Authors.

The book is intended for:

• project managers • quality assurance personnel • software engineers • consultants • academics • and, lastly and importantly, any other people interested in working actively towards a particular measurable objective in software quality.

Support material for the book can be found on the Authors' website, http://www.mcgraw-hill.co.uk/vansolingen

Ezran, M.; Morisio, M.; Tully, C.: Practical Software Reuse: the essential guide

Freelife Publ., Paris, 1998 (185 p.)

Properly understood, and deployed in the right context, software reuse offers high value to businesses that develop software, through radical improvements to their software capability. Attempts to adopt reuse without a sound understanding of all the issues involved, however, can lead to expensive failure.
This essential guide is dedicated to first-time enquirers who are wondering if they should get started and how. It aims to present a vision and a greater understanding of the potential and reality of reuse, and to encourage much wider-take-up of reuse practices.
The book seeks throughout to emphasise the practical issues that influence success or failure in reuse, and to provide a concise and balanced coverage of the essentials of the subject rather than going into undue depth or detail on some topics at the expense of others. It is neither an academic textbook, nor a cookbook with ready-made recipes claiming to tell readers exactly "how to do it". It does aim, however, to be "an easy read" for business executives, software managers and software developers, whatever kind of software or applications are developed by their organisations, and whatever size those organisations may be.
A special feature of the book is the frequent use of experience notes, drawn from the real-life experience of organisations that have embarked on the reuse adventure.

The three authors have drawn on their extensive experience of reuse and of software process improvement.

Poels, G.: On the Formal Aspects of the Measurement of Object-Oriented Software Specification

University of Leuven (Belgium), 1999 (507 p.)

This work contributes to the state-of-the-art in software measurement by proposing a suite of measures for object-oriented software specifications related to a particular layer in a software system's architecture: the enterprise model. This model of business entities, business events, and business rules builds the nucleus of a software system. Modern software development strategies, such as the one recommended by the MERODE research group at the K.U.Leuven's Department of Applied Economics, start building a software system from the stable foundations laid by the enterprise model. As a consequence, measuring the enterprise model and its components offers great potential for the early assessment, prediction and control of the software engineering and management variables of interest. These include the cost of software development and maintenance, the quality of the software system, and the effectiveness and efficiency of the methods, techniques and tools that are used. However, this is not the only motivation for measurement. As a representation of the business within the scope of the software system, the enterprise model might tell us something about the business functioning itself. It is therefore a worthwhile exercise to conceptualise and quantify its attributes.

Pfleeger, S.L.: Software Engineering - Theory and Practice

Prentice Hall, New Jersey, 1998 (576 p.) *ISBN 0-13-624842-X*

A firm grounding in software engineering theory and practice is essential for understanding how to build good software and for evaluating the risks and opportunities that software presents in our lives. Software Engineering blends the two current software engineering worlds: that of the practitioner whose main focus is an building high-quality products to perform useful functions, and that of the researcher who strives to improve the quality of products and the productivity of those who build them. Dr. Pfleeger presents concepts at both the micro and macro levels, using numerous case studies and examples to illustrate clearly how large software development projects progress from need to idea to reality.

Noteworthy Features

- Applies concepts consistently throughout to two common examples - a typical information system and a real-time system.
- Embeds concepts such as reuse, risk management, and quality assurance in the software engineering activities that are affected by them instead of treating them as separate issues.
- Considers measurement issues as an integral part of software engineering strategy, rather than as a separate discipline.
- Features an associated Web page containing examples from current literature and links to other Web pages related to tools, methods, annotated bibliographies, newsletters, and more.
- Brief case studies are shown as sidebars in the book, with an expanded version available on the Web page.

- Expresses results at both macro and micro levels:
  - Macro level explains what the content of the chapter means for development teams.
  - Micro level discusses what the content implies for individual developers.
- Includes thought-provoking questions about legal and ethical issues in software engineering.
- Concludes each chapter with research and practice summaries.

IWSM'99:

9th International Workshop on Software Measurement,
September 8-10, 1999, Montreal – Mont-Tremblant, Canada
see: **http://www.lrgl.uqam.ca/iwsm99/**

CONQUEST'99:

Conference on Quality Engineering in Software Technology,
September 27-28, 1999, Nuremberg, Germany
see: **http://www.asqf.de/**

FESMA'99:

Second European Conference on Software Measurement,
October 4-7, 1999, Amsterdam, Netherlands
see: **http://www.ti.kviv.be/conf/fesma.htm**

IASTED'99:

International Conference Software Engineering,
October 6-8, 1999, Scottdale, Arizona, USA
see: **http://www.iasted.com/**

IFPUG'99, Fall:

International Function Point User Group Fall Conference,
October 20-22, 1999, New Orleans, USA
see: **http://www.ifpug.org/conferences/conf.html**

METRICS'99:

Sixth International Symposium on Software Metrics,
November 5-6, 1999, Boca Raton, Florida, USA
see: **http://www.iese.fhg.de/METRICS99/metrics99.htm**

metrics themes are also discussed in the yearly **OOIS, ECOOP** *and* **ESEC** *conferences*

Other Information Sources and Related Topics

- **http://rbse.jsc.nasa.gov/virt-lib/soft-eng.html**
  Software Engineering Virtual Library in Houston

- **http://www.mccabe.com/**
  McCabe & Associates. Commercial site offering products and services for software developers (i. e. Y2K, Testing or Quality Assurance)

- **http://www.sei.cmu.edu/**
  Software Engineering Institute of the U. S. Department of Defence at Carnegie Mellon University. Main objective of the Institute is to identify and promote successful software development practices.
  Exhaustive list of publications available for download.

- **http://dxsting.cern.ch/sting/sting.html**
  Software Technology INterest Group at CERN: their WEB-service is currently limited (due to "various reconfigurations") to a list of links to other information sources.

- **http://www.spr.com/index.htm**
  Software Productivity Research, Capers Jones. A commercial site offering products and services mainly for software estimation and planning.

- **http://fdd.gsfc.nasa.gov/seltext.html**
  The Software Engineering Laboratory at NASA/Goddard Space Flight Center. Some documents on software product and process improvements and findings from studies are available for download.

- **http://www.qucis.queensu.ca/Software-Engineering/**
  This site hosts the World-Wide Web archives for the USENET usegroup comp.software-eng. Some links to other information sources are also provided.

- **http://www.esi.es/**
  The European Software Institute,Spain

- **http://saturne.info.uqam.ca/Labo_Recherche/lrgl.html**
  Software Engineering Management Research Laboratory at the University of Quebec, Montreal. Site offers research reports for download. One key focus area is the analysis and extension of the Function Point method.

- **http://www.SoftwareMetrics.com/**
  Homepage of Longstreet Consulting. Offers products and services and some general information on Function Point Analysis.

- **http://www.utexas.edu/coe/sqi/**
  Software Quality Institute at the University of Texas at Austin. Offers comprehensive general information sources on software quality issues.

- **http://wwwtrese.cs.utwente.nl/~vdberg/thesis.htm**
  Klaas van den Berg: Software Measurement and Functional Programming (PhD thesis)

- **http://divcom.otago.ac.nz:800/com/infosci/smrl/home.htm**
  The Software Metrics Research Laboratory at the University of Otago (New Zealand).

- **http://ivs.cs.uni-magdeburg.de/sw-eng/us/**
  Homepage of the Software Measurement Laboratory at the University of Magdeburg.

- **http://www.cs.tu-berlin.de/~zuse/**
  Homepage of Dr. Horst Zuse

- **http://dec.bournemouth.ac.uk/ESERG/bibliography.html**
  Annotaded Bibliography on Object-Oriented Metrics

- **http://www.iso.ch/9000e/forum.html**
  The ISO 9000 Forum aims to facilitate communication between newcomers to Quality Management and those who, having already made the journey have experience to draw on and advice to share.

- **http://www.qa-inc.com/**
  Quality America, Inc's Home Page offers tools and services for quality improvement. Some articles for download are available.

- **http://www.quality.org/qc/**
  Exhaustive set of online quality resources, not limited to software quality issues

- **http://freedom.larc.nasa.gov/spqr/spqr.html**
  Software Productivity, Quality, and Reliability N-Team

News Groups

- **news:comp.software-eng**

- **news:comp.software.testing**

- **news:comp.software.measurement**

METRICS NEWS

VOLUME 4 *1999* *NUMBER 1*

CONTENTS

ISSN 1431-8008