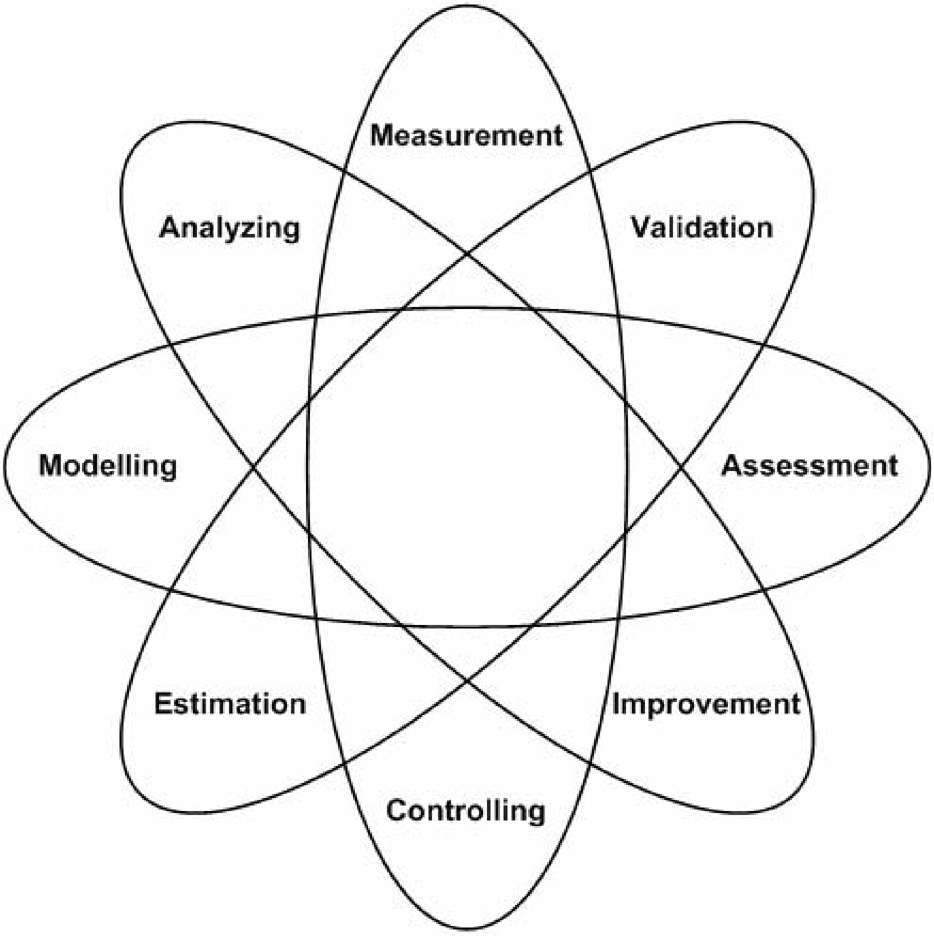
Volume 10, Number 1, August 2005

METRICS NEWS

Journal of the Software Metrics Community



Editors:

Alain Abran, Manfred Bundschuh, Reiner Dumke, Christof Ebert, Horst Zuse

Otto-von-Guericke-Universität Magdeburg

Université du Québec
École de technologie supérieure

SM L@b

COSMIC-FFP
Common Software Measurement International Consortium

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

Editors:

Alain Abran
Professor and Director of the Research Lab. in Software Engineering Management
École de Technologie Supérieure - ETS
1100 Notre-Dame Quest,
Montréal, Quebec, H3C 1K3, Canada
***Tel.:** +1-514-396-8632, **Fax:** +1-514-396-8684*
aabran@ele.etsmtl.ca

Manfred Bundschuh
Chair of the DASMA
Sander Höhe 5, 51465 Bergisch Gladbach, Germany
***Tel.:** +49-2202-35719*
bundschuhm@dasma.de
http://www.dasma.org

Reiner Dumke
Professor on Software Engineering
University of Magdeburg, FIN/IVS
Postfach 4120, D-39016 Magdeburg, Germany
***Tel.:** +49-391-67-18664, **Fax:** +49-391-67-12810*
dumke@ivs.cs.uni-magdeburg.de

Christof Ebert
Dr.-Ing. in Computer Science and Director SW Coordination
Alcatel HQ
54, Rue La Boetie, F-75008 Paris, France
***Tel.:** +33-675-091999, **Fax:** +33-1-4076-1475*
christof.ebert@alcatel.org

Horst Zuse
Dr.-Ing. habil. in Computer Science
Technical University of Berlin, FR 5-3,
Franklinstr. 28/29, D-10587 Berlin, Germany
***Tel.:** +49-30-314-73439, **Fax:** +49-30-314-21103*
zuse@tubvm.cs.tu-berlin.de

Editorial Office: Otto-von-Guericke-University of Magdeburg, FIN/IVS, Postfach 4120, 39016 Magdeburg, Germany

Technical Editor: DI Martin Kunz

The journal is published in one volume per year consisting of two numbers.

==
=

C A L L   F O R   P A R T I C I P A T I O N

==
=

15th INTERNATIONAL WORKSHOP ON SOFTWARE MEASUREMENT – IWSM 2005

Workshop : Sept 12-14, 2005

Tutorial Sept. 15

Preliminary Program V6

Co-sponsored by:
École de Technologie Supérieure - Université du Québec
Otto von Guericke Universitat (Magdeburg, Germany)

In cooperation with:
COSMIC – Common Software Measurement International Consortium
SPIN-Montréal

Location:

École de technologie supérieure
1100 Notre-Dame Street - West
Montréal (Québec) CANADA
www.gelog.etsmtl.ca/iwsm2005

Software measurement is one of the key technologies to control & to manage the software development process. Measurement is also the foundation of both sciences and engineering.

Conference co-chairs:

Alain Abran & Reiner Dumke

alain.abran@etsmtl.ca
Phone: +1-514-396-8632

dumke@ivs.cs.uni-magdeburg.de
Phone: +49-391-6718664

Fees:
Workshop: $300 for the full workshop or $150 per day
Tutorial: $200 for one day – or $125 for half day
(Prices including: lunches and related Canadian taxes)

Fees for authors = none (lunches not included – prices = $15 lunch including taxes)

September 12 (Monday) – Measurement Systems

8:30	ABRAN, A. DUMKE, R.	Welcome and Introduction
9:00	Kececi (USA)	Experiments-based measurement process
9:30	Braungarten (Germany)	Towards Meaningful Metrics Data Bases
10:00	**Break**	
10:30	Daneva (Netherlands)	Architecture Maturity and Requirements Maturity Do not Explain Each Other
11:00	Reitz (Germany)	Tool supported effort monitoring and estimations in EAI multi projects
11:30	**Lunch – salon des Professeurs** **Keynote: Offshoring - 6 years of industrial experience in distributed software development** by Andreas Schmietendorf (Germany)	
13:30	Kunz (Germany)	Measurement eLearning - A classification approach for eLearning-Systems
14:00	April (Canada)	Software Maintenance expert system (SM^{xpert}) - Measuring the use of the knowledge base
14:30	Break	
15:00	Paulus (Belgium)	On the Applicability of FPA and COCOMO II in a workflow and maintenance context
15:30	Dekkers (Netherlands)	Benchmarking is an essential control mechanism for management
16:00	Dery (Canada)	Investigation of the Effort Data Consistency in the ISBSG Repository
16:30	**Discussions panel & Closing**	

September 13 (Tuesday) – Functional Size Measurement Systems

9:00	Monge (Spain)	MTPF Function Points Measure Early Method
9:30	Gencel (Turkey)	A Case Study on Using Functional Size Measurement Methods for Real Time Systems
10:00	Break	
10:30	Dumke (Germany)	FSM Influences and Requirements in the CMMI-Based Software Processes
11:00	Glowacki (Poland)	Adapting Use Case Model for COSMIC-FFP Based Measurement
11:30	**Lunch**	
13:30	Abu Talib (Canada)	COSMIC-FFP & Functional Complexity (FC) Measures: A Study of their Scales, Units and Scale Types
14:00	Desharnais (Canada)	Measurement Convertibility - From Function Points to COSMIC-FFP
14:30	Break	
15:00	Nagano (Japan)	Improvement of analysis model by removing improper parts based on functional size measurement
15:30	Santillo (Italy)	Functional details visualization and classification in the COSMIC FSM framework
16:00	Edwards (USA)	Function Point Counting Patterns
16:30	Schmietendorf (Germany)	Complex Evaluation of an Industrial Software Development Project
17:00	**Discussions panel: International Certification on COSMIC-FFP ISO 19761**	

September 14 (Wednesday) – Measurement Foundations

9:00	Malhotra (India)	Analysis of Object-Oriented Metrics
9:30	Ormandjieva (Canada)	Measurement of Cohesion and Coupling in OO Analysis Model Based on Crosscutting Concerns
10:00	**Break**	
10:30	Blazey - Dumke (Germany)	Information Management for Industrial eLearning Projects
11:00	Al Qutaisch (Canada)	An Analysis of the Designs and the Definitions of the Halstead's Metrics
11:30	**Lunch – Keynote – Salon des Professeurs** **Software Measurement Body of Knowledge – Overview of Empirical Support** By Luigi Buglione (Italy)	
13:30	Rodriguez (Spain)	Using Simulation to Determine the Sensibility of Error Sources for Software Effort Estimation Models
14:00	Neumann (Germany)	Independent Dimensions of Software Complexity
14:30	Break	
15:00	Garbajosa (Spain)	The Measurement Service in Software Engineering Environments
15:30	Lopez (Belgium)	On the Impact of the Types Conversion in Java onto the Coupling Measurement
16:00	Dumke (Germany)	An Agent-based Measurement Infrastructure
16:30	Cheikhi (Canada)	Analysis of the Designs of Coupling Measures: A Case Study
17:00	**Concluding panel & Invitation to IWSM 2006 in Germany – Dumke & Abran**	
17:30	**Closure**	

September 15 (Thursday)

ISO 19761 – COSMIC-FFP Tutorials

Instructor: Ton Dekkers (Sogeti – Netherlands)

Participants will carry out measurement exercises with case studies

9:00 – 11:30	Instructor: **Ton Dekkers**	***Part 1: Introductory Tutorial***
11:30	**Lunch Time**	
12:30 – 14:30	Instructor: **Ton Dekkers**	***Part 2: Advanced Tutorial***
14:30	Break	
15:00 – 17:00	Instructor: **Ton Dekkers**	***Part 2: Advanced Tutorial – Continued***
17:00	**Closure**	

Tutorial by: Ton Dekkers (Sept. 15)

Measurement of Software Functional Size using Cosmic-FFP (ISO 19761)

Aim

The aim of the tutorial is to demonstrate the practical application of the COSMIC-FFP to derive a measurement of function size, and how this can contribute to the overall software development process.

Audience

1. FSM practitioners wishing to learn more about the application of the COSMIC-FFP method by means of a detail analysis of ISO case studies
2. Anyone wishing to learn more about the COSMIC-FFP methods and how it can contribute to the overall development process.

Tutorial Content

Functional Size Measurement Methods (FSMM) measure the size of the functionality delivered by software as expressed in the software Function User Requirement (FUR). Given a complete and accurate FUR, analysts experienced in the FSMM will report the same functional size.

The COSMIC-FFP FSMM defines a process for measuring size, and one of the steps in that process is called "Mapping".

In this tutorial, case studies from ISO-14143-4 are used to construct the COSMIC-FFP Measurement model. This will demonstrate the application of the COSMIC-FFP method in a practical situation.

DASMA Metrikon 2005 Entwurf Programm

	Spur A	Spur B
	15.11.05	
08:00:00	Registrierung	
09:00:00	Begrüßung durch Gastgeber und DASMA-Vorstand	
09:15:00	Hauptvortrag I **Software-Qualitätsmessung: Von der Theorie zur Empirie** *Prof. Dr. Peter Liggesmeyer*	
10:15:00	Pause – Registrierung – Gedankenaustausch – Ausstellung	
10:45:00	Sitzung A1 – Thema Usability Metriken – Stand und Perspektiven *Rüdiger Liskowsky; Christina Zahn; Ralf Überfuhr* Verwendung der Data Envelopment Analysis (DEA) im Kontext von ERP Implementierungsprojekten: Vergleich und Aufwandsschätzung *Stefan Koch*	Sitzung B1 – Thema Eine Untersuchung zum Metrikeinsatz in der Industrie *Tilmann Hampp* Empirische Betrachtungen zur Softwareentwicklung im Rahmen von Offshore-Kooperationen *Andreas Schmietendorf*
12:15:00	Mittagspause – Gedankenaustausch – Ausstellung	
14:00:00	Sitzung – Industrie 1 M-System NT - Ein flexibles, datenbank-basiertes Mess- und Analyse-System *Jürgen Münch; Axel Wickenkamp* Industriebeitrag 2 *NN*	Sitzung – Industrie 2 Industriebeitrag 3 *NN* Industriebeitrag 4 *NN*
15:00:00	Pause – Gedankenaustausch – Ausstellung	
15:30:00	Sitzung A3 – Thema Resolving the Mysteries of the Halstead Measures *Horst Zuse* Ein zielorientierter Ansatz für kontinuierliche,automatisierte Messzyklen *Christoph Lofi* DA-Preis-Vortrag *Preisträger des DASMA-DA-Preises 2005*	Sitzung B3 – Thema Warum Prüfen oft 50 mal länger dauert als Lesen und andere Überraschungen aus der Welt der Software-Reviews *Peter Rösler* Das wundersame Verhalten von Entwicklern beim Einsatz von Quellcode-Metriken *Frank Simon; Daniel Simon*
17:00:00	Pause – Gedankenaustausch – Ausstellung	
19:00:00	Abendprogramm	

16.11.05

08:30:00	Sitzung A4 – Thema Measuring the Effectiveness of Software Testing *Harry Sneed* Definition and evaluation of system requirements metrics based on CMMI *Silvia Linschi; Marek Leszak*	
10:00:00	Pause – Gedankenaustausch – Ausstellung	
10:30:00	Sitzung A5 – Thema Die Requirement Points Analyse – ein Ansatz zur Aufwandsschätzung im Requirements Engineering *Nicolas Fernando Porta* Messen von Projekt-Risiken *Johannes Bergsmann* Management von Projekten in neuen Domänen durch prozessorientierte Erfassung und Analyse von Aufwandsdaten *Fabio Bella; Jürgen Münch; Alexis Ocampo*	Sitzung B5 – Thema QScope - Metriken formulieren, berechnen und visualisieren *Daniel Germanus; Lukas Mrokon* Goal-oriented Data Visualization with Software Project Control Centers *Jens Heidrich; Jürgen Münch* An ISO 15939-Based Infrastructure Supporting the IT Software Measurement *R. Dumke; R. Braungarten; M. Kunz; H. Hegewald*
12:00:00	Mittagspause – Gedankenaustausch – Ausstellung	
13:30:00	Sitzung A6 – Thema Nicht falsch, sondern das Falsche geschätzt! Warum viele Function Point Installationen scheiterten? *Robert Hürten* Software-Metriken als Schlüssel zur Aufwandsabschätzung - Anforderungsmanagement in der Automotive Software-Entwicklung *Dirk Abendroth; Wolfram Bohne* Ermittlung von Synergiepotenzialen auf Grundlage der Function Point Analyse *Harald Kreher; Marcus Mauser*	
15:00:00	Pause – Gedankenaustausch – Ausstellung	
15:30:00	Hauptvortrag II **Produktivität messen und verbessern** *Dr. Christof Ebert*	
16:30:00	Abschlusssitzung	

Bericht zum 6. Workshop „Performance Engineering in der System- und Softwareentwicklung“ (PE 2005)

Reiner Dumke (Sprecher der GI-FG 2.1.10)
dumke@ivs.cs.uni-magdeburg.de

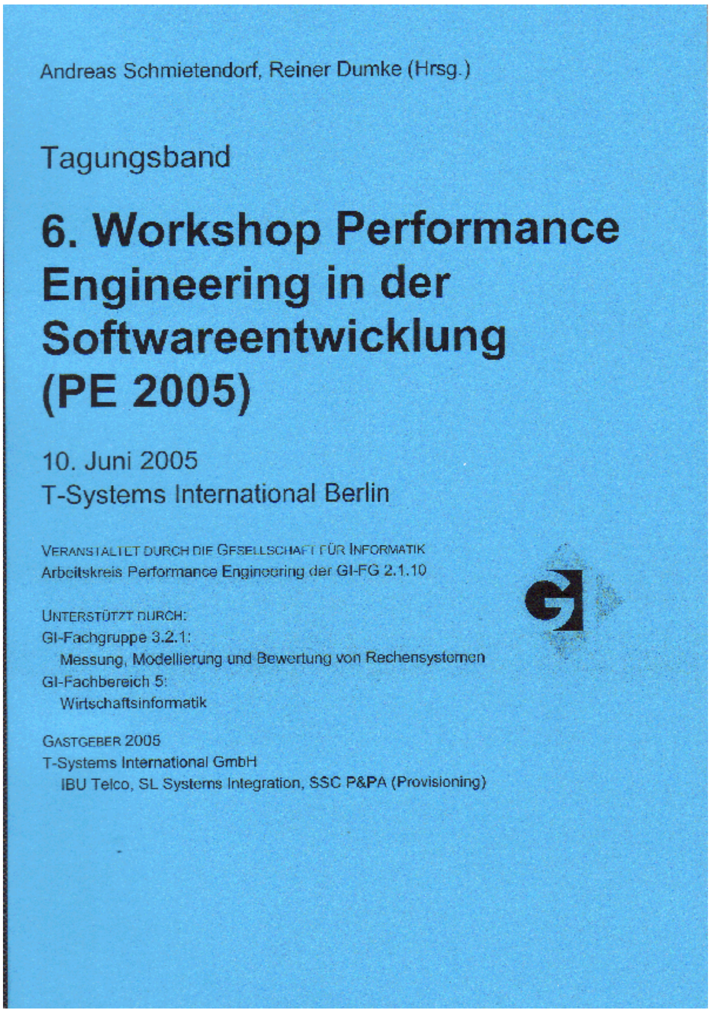
Andreas Schmietendorf (Sprecher des GI-PEAK)
andreas.schmietendorf@t-systems.com

Der im Jahr 2000 gegründete Arbeitskreis zum Performance Engineering (kurz GI-PEAK) führte am 10. Juni 2005 bereits zum 6. Mal seinen jährlich stattfindenden Workshop durch. Gastgeber in diesem Jahr war die T-Systems International in Berlin. An dieser Stelle sei noch einmal Herrn Dr. Evgeni Dimitrov von der T-Systems für die hervorragende Organisation herzlich gedankt.

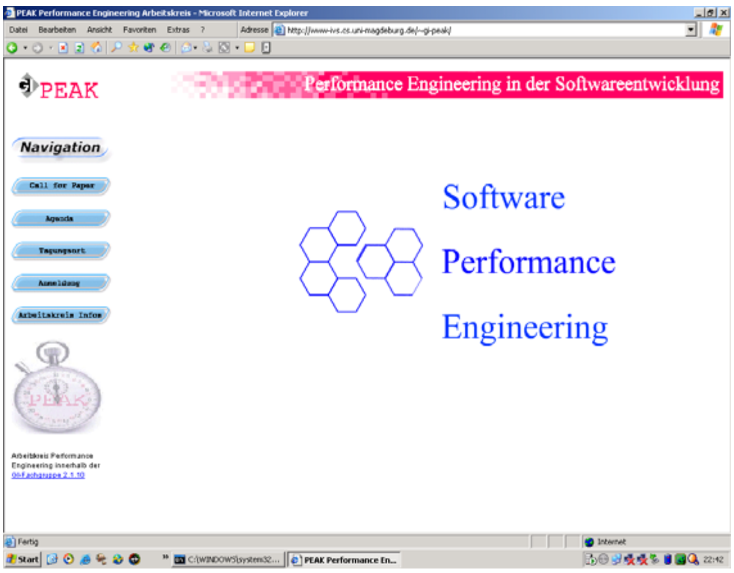
Beiträge des Workshops

Die Beiträge des Workshops bildeten einen Canon zu Themen aus dem industriellen aber auch akademischen Umfeld, wobei der Schwerpunkt - wie bereits in den vergangenen Jahren - aus der Industrie kommt. Im Folgenden findet sich die Übersicht zu den entsprechenden Vorträgen bzw. Postern.

- *Lev Olkhovich, Evgeny Rachinsaky, Andreas Henning*: Using Partly-Emulated Execution Environment for Measuring QoS-related Parameters of Business Processes
- *Fritz Zbrog, Reiner Dumke*: Neue Konzepte und Ansätze für die Performance-Messung mittels Java 5.0
- *Uwe Ryssel, Mario Neugebauer, Gunnar Stein, Klaus Kabitzsch*: Ein Low-Cost-Protokollanalysator zur Messung von Transaktionszeiten
- *Rainer Gerlich, Ralf Gerlich, Thomas Boll, Klaus Ludwig, Philippe Chevalley, Neil Langmead*: Software Diversity by Automation
- *Torsten Hörmann*: Content-basiertes Quality-of-Service (QoS) Management in IP-basierten Netzwerken
- *Lars Gentsch*: BOSPORUS – Ein generisches Framework zur Prozessdaten-verarbeitung
- *Andreas Schmietendorf, Dmytro Rud*: Performanceaspekte choreographierter Anwendungen auf Basis lose gekoppelter Web Services
- *Jan Lolling*: Performance in der Entwicklung und zur Laufzeit durch MDA und einem objektorientierten Datenbankdesign



Alle Beiträge sind im Tagungsband [Schmietendorf/Dumke 2005] zum Workshop enthalten, der über den Sprecher des Arbeitskreises zu einem Unkostenbeitrag von 15,- Euro noch bezogen werden kann. Weitere Informationen (auch zum PE-Workshop 2006) unter **`http://ivs.cs.uni-amgdeburg.de/~gi-peak/`**



A Proposal for a Metrics Data Base Maturity Model

René Braungarten

Software Engineering Group
Institute for Distributed Systems, University of Magdeburg
P.O. Box 4120, D-39106 Magdeburg, Germany

braungar@ivs.cs.uni-magdeburg.de

Abstract. *The importance of measuring artifacts emerging during the software development process is beyond controversy, these days. To expand those data towards empirical series of measurement and thus to benefit from it in the longrun, structured and persistent acquisition is almost compulsory. Renowned measurement guidance describes the prominence of measurements and supposes the usage of appropriate metrics data bases. But, analogous to common project management or process frameworks they leave organizations alone in providing guidelines or standards for the collection process, too. So we propose Metrics Data Base Maturity Model (MDB^{MM}) bearing the potential to appraise an organization's measurement collection process' maturity on five level ordinal scale and to expose leverage points for its improvement. Withal, the conceived model strongly avails itself of an investigation in factors influencing and MDB's maturity thereby not abstracting away from accepted frameworks like CMMI, V-Modell® XT, or Measurement-CMM.*

1 Introduction

In the area of software development measurement and assessment of artifacts (for instance specification documents, modeling, or source code) have become fundamental especially for industrial organizations striving for optimization of development efficiency and cost effectiveness for economic reasons. However, for meaningful quantification and appraisal of processes, products, and ressources occurring in the course of software development, organizations strongly depend on qualified measurement guidance that provide answers for several essential questions like:

What shall be measured? Since measurement is usually an activity causing additional effort and cost in an environment of limited financial and staffing resources, organizations should only focus on measuring meaningful indicators that map their operational and/or strategic goals for themselves and their projects on specific measurement aspects. For this purpose the Goal Question Metric (GQM) approach [1] can help in discovering adequate measurement goals and related metrics.

How shall measurement be performed? Issues like complexity and missing traceability of poorly arranged or adhoc measurement programs, resulted in the development of the information-driven measurement process Practical Software Measurement (PSM) [2] that incorporates technical and business goals of organizations. Then, with PSM serving as its base document, the international standard ISO/IEC 15939 [3] defines measurement process for software development and systems engineering, whose main target is to "… identify the activities and tasks that are necessary to successfully identify, define, select, apply and improve software measurement within an overall project or organizational measurement structure." Withal, the process is not only applicable to all software-related engineering and management disciplines but also flexible, tailorable, and adaptable to the needs of different users.

How well can the measurement be performed? After understanding how an exemplary measurement process should look like, the inspection of an organization's measurement capabilities could help in determining the present state and could circumvent starting points for improvement. Regrettably, no official standard addresses this issue. Hall and Fenton [4] presented list of 15 meaningful success factors for measurement programs that give premises for success but not roads to get there. However, there have been different approaches to outline measurement programs and assess measurement capabilities with different appreciation [5], [6], [7], [8].

Subsuming the introduced guidance, one has to realize that in fact they describe the prominence of measurements and suppose the usage of appropriate MDB in different fields of application. But, analogous to common project management or process frameworks they leave organizations alone in providing guidelines or standards for the collection process, too. Substantiated by study in practice [9], that lack leads to multitude of more or less poorly formed MDB in organizations. So we propose Metrics Data Base Maturity Model (MDB^{MM}) bearing the potential to appraise an organization's measurement collection process' maturity and to expose leverage points for its improvement. That model is intended to address the question: *How meaningful is the software measurement collection process?*

For this purpose, after mentioning related work in section two, we strongly avail ourselves of an investigation in factors influencing and MDB's maturity that is detailed in section three, thereby not abstracting away from accepted frameworks like CMMI [10], V-Modell® XT [11], or Measurement-CMM. As result, the MDB^{MM} introduced in section four enables us to appraise an organization's measurement collection process' maturity with the aid of maturity scale similar to that of CMMI-SE/SW simultaneously suggesting ways for improvement. Ultimately, concluding remarks and future research prospective close the paper in section five.

2 Related Work

In the whole, there are two main related works focusing on the topic of maturity modeling of rather business data aspects than the acquisition process of software measurements or metrics, indeed.

On the one hand there is the Enterprise Data Management Maturity Model of DataFlux [12] that aims at assessing an organization's data management maturity on a four level ordinal scale (unaware, reactive, proactive, and predictive) to give it better understanding of risks of undervalued data management practices but also to reveal benefits and costs. And on the other hand there is the Information Management Maturity Model of METAGroup [13] that attends the fact of measuring information maturity on five level ordinal scale (aware, reactive, proactive, managed, and optimized) to enable organizations ”… to put in place appropriate programs, policies, architecture, and infrastructure to manage and apply information better.”

3 Factors Affecting MDB Maturity

In the course of our investigations, we could reveal a number of influencing factors for an MDB’s maturity that can be grouped into seven categories we want to describe subsequently: Guidance, people, infrastructures, data management issues, applicable operations, business usefulness, and the supposable effort, ultimately.

3.1 Guidance

In contrast to most service oriented organizations involving repetitive operations, in the IT area temporary endeavors for products are most often segmented in projects of unique nature. But since those projects aim at creating software and/or hardware deliverables they cannot be disassociated from their underlying, recurring development processes that can act as leverage points for overall organizational improvements, in turn. Thence, we pitched our investigations on assorted representatives:

1) Project Management Guidance: Shaping up as the sum of knowledge within the profession of project management irrespective of the sector of industry applied to, the Project Management Body of Knowledge (PMBOK) [14] describes processes and knowledge areas in the manner of a best practices collection. Thence, its extensive international significance is reflected in the approved IEEE standard 1490-1998.

2) Process Frameworks: Being probably the most popular and internationally established process frameworks in the IT area, the sub models of Capability Maturity Model (CMM) and its successor CMM Integration (CMMI) of the Software Engineering Institute (SEI) forming an all-embracing model are based upon a collection of best practices, too. Instead of uncoupled and abstract theoretic models, especially the direct successor of CMM, namely CMMI-SE/SW, is intended to help an organization in process appraisal by means of capability and maturity levels as well as to spot on activities required for a successful software and systems development.

Beyond the aforementioned ones there are other important process frameworks, as well. Among them there is V-Modell® XT. It also turns out to be a mainstream process model and standard for establishment and improvement of an organization’s project management and project to exhibit a lasting effect on its success by improving both, product and process quality. Financed by the German Ministry of the Interior and intrinsically developed for the defense sector in 2004, V-Modell® XT emanated from its predecessor V-Model of 1997. Besides tailoring the general model pursuant to an organization’s project environment or specifics, the model identifies involved roles, activities to be covered in the course of project execution as well as the appearance and usage of information products and measurement.

3) Software Measurement Guidance: Generally, in the field of software measurement there is a triumvirate of acts already scribed afore: First of all, the international standard ISO/IEC 15939 prescribes an exemplary software measurement process that is flexible, tailorable, adaptable to different user's needs, and applicable to all engineering and management disciplines. Secondly, PSM in its function as foundation of ISO/IEC 15939 provides a measurement guidebook and a process description, as well. At last, concepts from both were adopted for the Measurement & Analysis (M&A) process area of CMMI showing the efficient usage of measurement in improving organizational processes. Moreover, Measurement-CMM [8] can be regarded as a generic model for appraisal and improvement of software measurement processes.

3.2 People

The analysis of different guidance and/or frameworks resulted in three factors:

1) Organizational Structure: PMBOK describes the structure of an organization to have a remarkable influence on the availability of resources to a project or on the terms under which they become available. Since measurement according to ISO/IEC 15939 involves two different roles apart from the virtual measurement user, namely a measurement analyst and librarian, administrative resources additional to the executive project staff causing extra (also financial) effort have to be provided. The degree of how much a project manager or person with similar tasks is disburdened from measurement and other administrative tasks can be read off from the organizational structure as presented in PMBOK [14]. The more time and resources are spent for measurement, one can almost act on the assumption that measurement's quality improves. Referring to PMBOK, structures occurring in performing organizations span functional, matrix (functional, balanced, projectized, or composite), and projectized ones with matrix structure probably being predominantly in practice.

2) Organizational Hierarchies: According to Kerzner [15] in conventional organizations one can find a five-level hierarchy that intergrades employees, teams, departments, divisions, and the organization. In connection with the organizational structure the hierarchy determines not only the measurement's quality but even its downstream tasks and extent.

3) Measurement Related Project Roles: As mentioned above, ISO/IEC 15939 describes the necessity of three roles involved in measurement activities: User, analyst, and librarian. To confront these roles with project planning and execution we again analyzed V-Modell® XT. Its authors make mention of 29 general project roles that need not necessary to be filled with different individuals. According to the notes of the model we could at first filter roles involved in measurement and consequently map them to the three roles of ISO/IEC 15939. Then, in order to derive where measurements or experiences emerge or are being used and to check their respective update frequency we merely assigned the measurement activities to the pertinent persons in charge by means of the model. For reasons of space we have to omit the results of the performed mapping and assignment and refer to [16] for details.

3.3 Infrastructure

In terms of commonly applied infrastructures the results of our previously performed field study [9] concerning prevalent MDB in practice indicated the application of manual (record keeping) techniques, monolithic tools, spreadsheet applications, and versatile database systems or repositories for measurement collection and storage.

Besides the discovered infrastructures we emphatically propose the usage of a data warehouse (DW). Since DWs do not only collect information from operative systems and perform its consolidation but even filter information of no importance as well as arrange it by topics and reveal essentials by analysis tools at regular intervals, they perfectly suit for measurement and evaluation purposes.

3.4 Data Management Issues

In order to conceive high-class metrics data bases fundamental data management criteria have to be met or taken into account, additionally. There are several interrelated issues having major impacts on MDB that shall be summarized beneath: quality, disposability, states, acquisition cost, volume, volatility, access and/or update frequency, and even structural aspects, after all.

1) Quality: In the same manner as a product's quality depends on many factors like on the quality of its design and production process, the involved people and infrastructures, data quality does. Wand and Wang [17] showed that data quality is a multidimensional concept with a lack of general agreement on its dimensions. Because most aspects being essential for internal (design and operation) and external (use and value) views concerning MDB are covered by their definition, we adhere to it.

2) Disposability: We experienced that in the measurement context one has to distinguish between organizational and technical disposability of measurements.

Since individuals are involved or their work products appraised during the measurement process, organizational disposability has to deal with an irrevocable obligation to observe confidentiality and privacy. Apart from an optimal exposure to measurement or its acquisition there are two extremes: On the one hand, single personnel might be an encumbrance (commonly called data capitalism) or on the other hand they might be too exuberant and unwarily (called data communism) — both phenomena are unwanted.

In contrast, aspects of technical disposability sound rather unemotively and intuitively: Amongst others, appropriate access opportunities ensuring access protection and a sufficient data transmission speed form base requirements.

3) States: Findings in literature [18] substantiate a classification of measurements according to their usage levels in:

- operative (active and critical for business application bearing high requirements for performance availability and disaster recovery),

- strategic (inactive, though essential for business application but will not stop operations, moderate requirements for performance availability and disaster recovery), and
- archive ones (needed for lessons learned, reference, legal and training purposes only, relaxed requirements for performance and availability, moderate disaster recovery requirements).

4) Acquisition Cost: As a fact, two factors must not be disassociated from each other: the measurement's origin and its acquisition cost. In virtue of Camp [19] a trade-off between internal and external sources has to be taken into account. While internal measurements are most probably very cost intensive to obtain but in qualitative concerns of higher value, external ones are easier to gather coupled with a potential lack of explanatory power.

5) Volume: Depending on the business usefulness explained later on, there are three different volumes we want to distinguish between: low (e.g. project measurement), moderate (e.g. product measurement), and high (e.g. process measurement) ones.

6) Volatility: Again depending on the business usefulness explained later on, there are three different data volatility levels we want to divide into: static (e.g. product measurement), semi-static (e.g. version measurement), and volatile (e.g. engineering) data.

7) Access/Update Frequency: According to our investigations detailed further in [16], the frequencies in which measurements and/or experiences are used or updated can be characterized best by three categories: low (only once), moderate (at every decision point or periodically), and high (day-by-day) ones.

3.5 Applicable Operations

Another important factor for MDB strongly connected with its business usefulness turns out to be the spectrum of provided or in a second instance supported operations on the measurements. It comprises basic and enhanced data operations as well as analysis procedures such like statistics, visualization, and various search variants or knowledge discovery techniques like data mining.

1) Basic and Enhanced Data Operations: As in most technical literature, Heuer and Saake [20] regard the five operations 'store, find, manipulate, delete, and rename' that form the foundation of conventional DBMS as elementary ones for data operations. In addition to the previously mentioned basic ones there are advanced data operations allowing more complex analyses on the measurements. As well-known from relational DBMS [20] one divides into unary operators (aggregation, projection, selection, scan), binary operators (different kinds of union, intersection, difference, and compositions).

2) Statistics: As analyzed in our study [9], especially for the analysis of measurements, there is a variety of meaningful statistical operations a MDB has to support or at least provide interfaces to appropriate programs. Withal, Berthold and Hand [21] explain and recommend the appliance of both linear models (regression, analysis of variance, logistic regression, and analysis of survival data) and multivariate analyses (e.g. principal components analysis, correspondence analysis, multidimensional scaling, cluster analysis and mixture decomposition, and latent variable or covariance structure models).

3) Visualization: In order to be useful for different endusers, the results of the statistical analyses should be illustrated in a suitable manner. Therefore, Berthold and Hand [21] describe related techniques (standard 2D/3D, geometricallytransformed, icon-based as well as dense pixel or stacked) and potential interactions (dynamic projection, interactive filtering, zooming, distortion, or linking and brushing).

4) Search: In case not only numerical measurements but also textual ones like lessons learned have to be dealt with, traditional search procedures mostly collapse. Thus, more complex search procedures should be provided [21]: calculus-based (gradient as per Gauss-Seidel, extrapolating gradient, or simplex), enumerative (exhaustive search or dynamic programming), stochastic (Monte Carlo, tabu, stochastic programming, or population-based incremental learning), and Bayesian methods (inference, modeling, or networks).

5) Knowledge Discovery in Databases (KDD): Representing the highest form of evolution of operations on MDB, KDD in its interdisciplinary nature entails manifold topics. Among them range the above mentioned like databases, statistics, visualization, and others like artificial intelligence, machine learning, etc. Subsuming, Frawley, Piatetsky-Shapiro, and Matheus define the term KDD as ". . . the nontrivial extraction of implicit, previously unknown, and potentially useful information from data." [22]

The virtual challenge of the interactive and iterative multistep KDD process is to extract high-level knowledge from low-level data investigating huge sets of data. A sub-step of KDD, well-known as Data Mining, applies data analysis and discovery algorithms producing a particular enumeration of patterns or models on the data. Withal, Data Warehousing helps set the stage for KDD by collecting and cleaning operative data to make them available for online analysis or decision support systems and by providing appropriate access facilities. [23]

3.6 Business Usefulness

Generally, the comparison of the two common process frameworks CMMI and V-Modell® XT showed off the existence of three different fields of application for MDB. Though, both frameworks intend the same meanings they use slightly different category denotations. We will confine to CMMI's categorization and notion about the usefulness of MDB in: engineering, project management, and process management.

3.7 Effort to be Scheduled

Since MDBs are applied for economic reasons, across the board, the effort that has to be allowed for it has to be considered and the money should be well spent. So we pitched on merely a couple of cost factors thought to have a major impact as there are installation (e.g. planning, software and hardware acquisition cost), operation (e.g. skill adaptation training), and maintainance (e.g. down time).

4 A Proposal for a Metrics Data Base Maturity Model

Even though maturity models like CMM or CMMI have been extensively developed, applied, and improved during the last two decades, their concepts are comparatively older. Long time ago in 1943, the recognized psychologist Maslow [24] proposed such a model for human needs development that can be seen as pioneering work for contemporary successors. Ever since Maslow's approach the basic aims of maturity models were not subject to changes and can generally be subsumed as the appraisal of an entity's capabilities to fulfill a certain task or perform a process in a first instance and consequently as the assistance in improving the appraised abilities. Hence, a form of maturity matrix is applied that arranges a number of defined levels, which represent the current fulfillment rating, as well as requirements that have to be met in order to proceed to the next level. This structured procedure is therefore meaningful because it rests upon the assumption that better work processes lead to better work products and because the aspect of measuring efforts is not abstracted away from its results.

Keeping that motivation at the back of our mind and having shown the necessity for guidance and improvement of prevalent and more or less sophisticated MDB as well as important aspects influencing an MDB's maturity, we started conceiving a MDB^{MM} to fill that gap.

4.1 Primary Objectives

With regard to the general maturity model concept the primary objectives of the proposed MDB^{MM} are dedicated to the improvement of the (software) measurement collection process and can be characterized by two aspects:

1) An organization shall be enabled to appraise its capabilities to collect and meaningfully analyze (software) measurements, and
2) is in a second step intended to assist an organization in improving acquisition processes and spotting on activities required for a more successful software development by means of paving a way for improvement concerning measurement collection rather than prescribing the concrete realization.

Thence, an organization scoring best on the MDBMM will stand out because:

- it completely adheres to internationally accepted guidance or best practices concerning project management, process frameworks and software measurement;
- the importance of organizational structures, hierarchies, and measurement related project roles is correctly mapped on measurement's acquisition and staff is fully committed;
- the technological infrastructure being most appropriate is utilized;
- a trade-off between data management issues such as quality, disposability, states, acquisition cost, volume, volatility, and access or update frequencies is performed;
- the full range of operations possible with the collected amount of data can be applied in order to extract best business usefulness;

- it is able to approximately determine the effort to be scheduled for installation, operation, and maintenance even in ever-changing environments.

Nevertheless, attention should be paid to the fact that the model assumes a need for measurement and consequently a measurement process however performed.

4.2 Specification of the Model

Like a number of other maturity models, the MDB^{MM} coarsely alludes to Maslow's Hierarchy of Needs and basically avails itself to the staged representation of SEI's CMMI: For the present development step of the MDB^{MM} we used a subset of CMMI's general components merely consisting of maturity levels.

1) Maturity Levels: As aforementioned in section three our investigations showed off that a number of factors have a major impact on an MDB's maturity that could be incorporated in the conceived maturity levels.

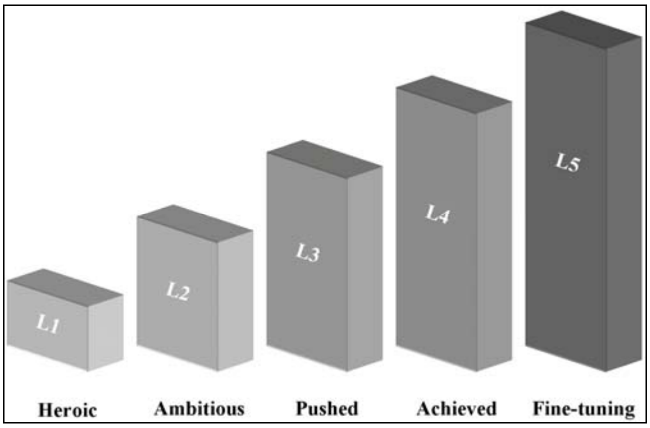


Figure 1: Levels of the MDB^{MM}

Withal, as summarized in figure 1 and table I these levels can be described by different attributes assigned to these factors and strikingly subsumed by their respective denotation:

1. Heroic: The first level of the model features organizations that have no process areas implemented resulting in an absolute dependence on *heroic* individual's endeavors acting in an ad hoc manner.

Usually, there is neither guidance for project management or process management nor the process of software measurement not to mention measurement collection. Organizational structures or hierarchies are disregarded and no measurement-related project roles assigned. Most often measurements are collected manually or with the aid of monolithic tools. While the operative measurement's disposability is ensured and acquisition costs are negligible, there might be problems with data quality. The volume of the measurement is either very low or also negligible due to high volatility; access and update frequency comport analogous. Similarly, the extent of applicable operations depends on the utilized tool's functional range but will most often focus on basic and enhanced data manipulation and visualization. The business usefulness restricts to the hero's self-assessment and improvement in engineering and virtually misses out because the cognitions limit to each single heroic employee. Indeed, the effort to be scheduled regarding installation, operation, and maintenance does not preponderate.

2. Ambitious: On the second level organizations have sporadic or insular measurement acquisition processes in place driven by *ambitious* and open-minded project managers that passed down their knowledge to project leaders. But there is a lack for an acquisition process spanning larger areas.

Presumably, the ambitious project manager is familiar with guidance for project management and has heard of the importance and enforcement of the software measurement process. Though, process frameworks are off scope and alien for him since he is merely responsible for a unique and temporary project. To the organizational structure and hierarchy being directly responsible to the project manager, probably he should have assigned measurement-related project roles. Owing to the fact that measurement is performed and the results are collected and used in a team environment, spreadsheet applications match best to infrastructure requirements but bear problems in ensuring data quality. Because the number of involved employees exceeds a singular one, internal disposability might be hard and expensive while measurements for comparison bought in from external sources are cheap. Analogous to the low volume of both, operative and strategic as well as semi-static project measurements, the access/update frequency responds to a moderate magnitude. The applicable operations are very similar in the field of spreadsheet applications and enable the user to perform basic and enhanced data manipulation, basic statistics, and visualizations. Following the trend, even the business usefulness increases: Alongside engineering purposes like self- and team-assessment as well as -improvement measurement can serve in project management e.g. for project estimation and tracking. Ultimately, the effort to be scheduled is minimal, too, since cheap standard software in an insular environment is only used by a few people.

3. Pushed: In case a functional manager in charge has recognized the importance and advantages of software measurement and also its structured acquisition, his practice definition will be *pushed* and has to be applied in his department by both ambitious and closed-minded project managers and their subordinates in the course of software product creation by multiple projects. Indeed, the software development process is neither observed nor measured.

Project management and software measurement guidance and standards are most often fully applied but process frameworks do not concern the functional manager. The importance of the organizational structure and hierarchy associated with him is internalized and thus measurement-related roles in every initiated project are established. Because the number of involved personnel has increased and multiple projects are put into motion, a more sophisticated kind of infrastructure like for instance central DBMS suits best regarding infrastructural aspects. Its utilization ensures a high quality of measurements in operative, strategic, and even archive states. Unfortunately, disposability and the increased measurement acquisition effort linked with higher costs might be serious issues. Additionally, the moderate volume of data from the entire volatility spectrum as well as different access/update frequencies can be problematically but handled best by the proposed infrastructure. In the same way, not only basic applications but also statistics, visualizations, complex searches, and even basic KDD mechanisms can be applied or interfaced to appropriate applications. Thence, business usefulness now stretches across engineering and management of multiple but cohesive projects in order to create a software product. To be honest, the cost for purchasing appropriate software and hardware as well as for installation, operation including skill adaptation training, and maintenance commensurate with the augmented usefulness and sophistication.

4. Achieved: The fourth level can be characterized by the fact that the measurement acquisition process is no longer merely subject to a department but to a whole division. Thereby a kind of commitment, understanding, and qualified enforcement could be *achieved* within the division in the course of time and has led to a practice policy of a division manager enabling efficient and now even process measurement and appraisal.

Most probably, the full range of guidance is needed, understood, and applied while the organizational structure and hierarchy are in line with measurement-related roles in all projects. As project, products, and processes of the entire division are measured and collected, federated or distributed DBMS or repositories fulfill technical requirements and ensure data quality. While measurement’s disposability and acquisition cost suffer from serious problems, full ranges of states, volumes, volatilities, and access/update frequencies have to be handled. Now, also the entire spectrum of applicable operations should be focused on or interfaced to: basic and enhanced data operations, statistics, visualization, searches, and KDD techniques. Business usefulness spans all facets of engineering, (multiple) project management, and process management. Consequently, one does not need to be astonished about the complexity and related financial effort this level causes concerning installation, operation, and maintenance.

5. Fine-tuning: When the measurement acquisition process has proved itself in a division, it seems to be just a consequence to adopt that process in the organization and to perform fine-tuning to align it to different division's needs and improve its performance as well as its cost-value ratio.

The organization avails itself to all available guidance, applies them where feasible and introduces suggestions for improvement. Withal, the organizational structure and hierarchies are mapped to measurement related roles and/or responsibilities thereby optimizing capacities with similar functions. Even the infrastructure is optimized by the enforcement of e.g. the data warehouse concept additive to DBMS or repositories. The optimization of the measurement acquisition process is reflected in better data quality, easier disposability and lower acquisition cost. More attention is paid to the occurring states and volatility occurrences which are thus better understood and treated. The volume of the data and its access/update frequency can be handled more structured because the operative data is preprocessed and stored redundant in DW. Yet, operations can be applied more efficiently and meaningful since DWs are conducive to complex KDD techniques. Besides engineering, project and process management purposes, the organization is now qualified to perform overall organizational measurements, its acquisition as well as evaluation. On the other hand one has to mind, that the effort that has to be allowed for has to comensurate with cost-value ratio. Thus installation, operation, and maintenance of the infrastructure and process being elaborate and cost-intensive have to be analyzed and improved, too, without doubt.

Table 1: Overview of Maturity Levels of the Proposed MDB^{MM}

	L1 Heroic	L2 Ambitious	L3 Pushed	L4 Achieved	L5 Fine-tuning
Guidance					
- Project Management	Disregarded	Concerned	Fully Applied	Fully Applied	Fully Applied + Optimizing
- Process Management	Disregarded	Disregarded	Disregarded	Fully Applied	Fully Applied + Optimizing
- Software Measurement	Disregarded	Concerned	Fully Applied	Fully Applied	Fully Applied + Optimizing
People					
- Organizational Structure	Disregarded	Concerned	Internalized	Internalized	Internalized + Optimizing
- Organizational Hierarchy	Disregarded	Concerned	Internalized	Internalized	Internalized + Optimizing
- Measurement Related Project Roles	All-in-one	Assigned in Team	Inherent Part of Department	Established and Inherent in Division	Established, Inherent, and Combined
Infrastructure					
- Infrastructure	Manual / Monolithic Tools	Spreadsheet Applications	Central DBMS	Federated or Distributed DBMS, Repositories	Federated or Distributed DBMS, Repositories, Data Warehouses
Data Management Issues					
- Quality	Very Problematic	Slightly Problematic	High	High	Optimal
- Disposability	All-in-One	Moderate	Hard	Very Hard	Very Hard
- Acquisition Cost	Negligible	A Bit Expensive	Moderate	High	High
- States	Operative	Operative/Strategical	Operative/ Strategical/Archive	Operative/ Strategical/Archive	Operative/ Strategical/Archive
- Volume	Low/ Negligible	Low	Low, Moderate	Low, Moderate, High	Low, Moderate, High
- Volatility	Volatile	Volatile, Semi-Static	Volatile, Semi-Static, Static	Volatile, Semi-Static, Static	Volatile, Semi-Static, Static
- Access/Update Frequencies	Low	Moderate	Low, Moderate, High	Low, Moderate, High	Low, Moderate, High
Applicable Operations					
- Basic/Enhanced	Tool Dependent	Basic	Predestinated	Predestinated	Predestinated
- Statistics	Disregarded	Basic	Interfaces to Tools	Interfaces to Tools	Interfaces to Tools
- Visualization	Tool Dependent	Basic	Interfaces to Tools	Interfaces to Tools	Interfaces to Tools
- Search	Disregarded	Disregarded	Predestinated	Predestinated	Predestinated
- KDD	Disregarded	Disregarded	Basic	Basic	Predestinated
Business Usefulness					
- Engineering	Self-Assessment/-Improvement	Self-/ Team - Assessment/-Improvement	Self-/ Team - Assessment/-Improvement, SQA	Self-/ Team - Assessment/-Improvement, SQA	Self-/ Team - Assessment/-Improvement, SQA
- Project Management	Disregarded	Estimation and Tracking (Single Project)	Estimation and Tracking (Multiple Projects)	Estimation and Tracking (Distributed, Multiple Projects)	Estimation and Tracking (Distributed, Multiple Projects)
- Process Management	Disregarded	Disregarded	Disregarded	Software Process Improvement	Overall Organizational Process Improvement
Effort to be Scheduled					
- Installation	Negligible	Minimal	Moderate	High	Immense
- Operation	Negligible	Minimal	Moderate	High	Immense
- Maintainance	Negligible	Minimal	Moderate	High	Immense

5 Conclusion

This paper has introduced a first structured approach to conceive a Metrics Data Base Maturity Model (MDB^{MM}) intended to support organizations in both, appraising and improving the software measurement collection process thereby applying more meaningful metrics data bases.

Together with an investigation in aspects affecting an MDB's maturity, we could start bringing together best practices and important influences. As a result we could produce a first approach towards a maturity model for MDB by initially characterizing maturity levels.

Besides an adjustment to potential objections, future research should consider a more thorough description of the model turning its attention to additional components as known from CMMI sub models like process areas, specific and generic goals or specific and generic practices. Over and above, an evaluation in practice is desirable and hence also strived for.

References

[1] V. Basili, G. Caldiera, and H. D. Rombach, *Encyclopedia of Software Engineering*. New York, USA: John Wiley & Sons, Inc, 1994, vol. 2, ch. The Goal Question Metric Approach, pp. 528–532.

[2] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, and F. Hall, *Practical Software Measurement: Objective Information for Decision Makers*. Reading, MA, USA: Addison-Wesley, 2001.

[3] ISO/IEC, *"ISO 15939 - information technology - software engineering - software measurement process,"* 2001. [Online]. Available: http://www.iso.org/

[4] T. Hall and N. Fenton, "Implementing effective software metrics programs," *IEEE Software*, vol. 14, no. 2, pp. 55–65, 1997.

[5] M. Daskalantonakis, V. R. Basili, and R. Yacobellis, "A method for assessing software measurement technology," *Quality Engineering*, vol. 3, no. 1, pp. 27–40, 1991. [Online]. Available: http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.38.pdf

[6] P. Comer and J. Chard, *"A measurement maturity model,"* Software Quality Journal, vol. 2, no. 4, pp. 277–289, December 1993.

[7] L. C. Briand, C. M. Differding, and H. D. Rombach, "Practical guidelines for measurement-based process improvement," Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, Tech. Rep. ISERN-96-05, May 1996. [Online]. Available: http://wwwagse.informatik.uni-kl.de/pubs/repository/briand96c/guidelines-96.pdf

[8] F. Niessink and H. van Vliet, "Towards mature measurement programs," in *Proceedings of the Euromicro Working Conference on Software Maintenance and Reengineering*. IEEE Computer Society, 1998, pp. 82–88. [Online]. Available: http://www.cs.vu.nl/ hans/publications/y1998/CSMR98.pdf

[9] R. Braungarten, M. Kunz, and R. Dumke, “An approach to classify software measurement storage facilities,” University of Magdeburg, Department of Computer Science, Magdeburg, Germany, Tech. Rep. 2, January 2005. [Online]. Available: http://ivs.cs.uni-magdeburg.de/sw-eng/agruppe/forschung/paper/Rene.pdf

[10] *“Capability Maturity Model ® Integration (CMMI), version 1.1,”* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2002-TR-012 ESC-TR-2002-012, March 2002. [Online]. Available: http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr012.pdf

[11] F. M. of the Interior, *V-Modell XT*, F. M. of the Interior, Ed. Federal Republic of Germany, 2004. [Online]. Available: http://www.kbst.bund.de/static/pdf/V Modell XT Komplett.pdf

[12] DataFlux, “Enterprise data management maturity model,” Whitepaper, June 2004. [Online]. Available: http://www.dataflux.com

[13] D. Laney, “Ascending the information maturity model: Part 1 data quality,” Online, March 2002, mETAGroup. [Online]. Available: http://watch.state.wi.us/Home/links/metapdfs/020315%20Maturity%20Model.pdf

[14] P. M. Institute, “A guide to the project management body of knowledge,” Online, 1996.

[15] H. Kerzner, Project Management: A Systems Approach to planning, scheduling and controlling, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2003.

[16] R. Braungarten, M. Kunz, A. Farooq, and R. R. Dumke, “Towards meaningful metrics data bases,” in *Proceedings of the 15th International Workshop on Software Measurement*, Montreal, Canada, September 2005 (to appear).

[17] Y. Wand and R. Y. Wang, “Anchoring data quality dimensions in ontological foundations,” *Communications of the ACM*, vol. 39, no. 11, pp. 88–95, November 1996. [Online]. Available: https://www.crg2.com/iqconference/documents/publications/TDQMpub/WandWangCACMNov96.pdf

[18] W. H. Inmon, C. Imhoff, and R. Sousa, *Corporate Information Factory*, R. Elliott, Ed. New York, USA: Wiley Computer Publishing, 1998.

[19] R. C. Camp, *Benchmarking: The Search for Industry Best Practices that Lead to Superior Performance*. Milwaukee, WI, USA: ASQC Quality Press, June 1989.

[20] A. Heuer and G. Saake, *Datenbanken - Konzepte und Sprachen*, 2nd ed. Bonn, Germany: International Thomson Publishing, 2000.

[21] M. Berthold, *Intelligent Data Analysis: An Introduction*, 2nd ed., M. Berthold and D. J. Hand, Eds. Secaucus, NJ, USA: Springer-Verlag, 2002.

[22] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, “Knowledge discovery in databases: An overview.” in *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991, pp. 1–30. [Online]. Available: http://www.kdnuggets.com/gpspubs aimag-kdd-overview-1992.pdf

[23] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *Ai Magazine*, vol. 17, pp. 37–54, 1996.

[24] A. Maslow, “Conflict, frustration, and the theory of threat,” *Journal of Abnormal and Social Psychology*, vol. 38, pp. 81–86, 1943.

Obvious Outliers in ISBSG Repository of Software Projects: Exploratory Research

Dominic Paré, Alain Abran

École de Technologie Supérieure
dominic.pare.1@ens.etsmtl.ca, alain.abran@etsmtl.ca

Abstract. *This paper discusses the issue of outliers in the repository of software projects of the International Software Benchmarking Standards Group - ISBSG. The criteria used for the identification of outliers is whether the productivity is significantly lower and higher, that is with significant economies or dis-economies of scale, in relatively homogeneous samples. Once the outliers identified, other project variables are investigated by heuristics to identify candidate explanatory variables that might explain such outliers' behaviors.*

1 Introduction

In software engineering, software projects productivity can vary considerably. It is then interesting to analyze the cause of these significant variations in order to be able to explain why the productivity of these projects is much higher or much lower than the average. The International Software Benchmarking Standards Group (ISBSG) [1] designed and maintains a repository of software projects. For productivity analysis and for estimation purposes, it is important on the one hand to identify outliers which have productivity behaviors significantly different from all other projects and, on the other hand, to try to discover next which factors have such a large influence (positive or negative) on the productivity of these projects.

This article identifies outliers in the ISBSG repository as well as candidate variables which could explain major differences in productivity by comparison to other projects in the same samples. This paper is structured as follows: section 2 presents on overview of the ISBSG repository, section 3 the identification of outliers for the samples selected, section 4 a discussion on these outliers and, section 5, a summary and discussion.

2 ISBSG repository

ISBSG makes available to industry and researchers, at a reasonable cost, an Excel data file which contains 92 variables for each of the projects in its repository, such as effort (in hours), functional size of the software (measured according to various standards: Function Points, COSMIC-FFP - ISO 19761, MKII), programming language, etc. [2].

The ISBSG repository is a multi-organizational, multi-application domain and multi-environment data repository, that is, its data content is fairly heterogeneous in projects characteristics. Data from either Release 8 (R8) with 2027 projects or Release 9 (R9) with 3024 projects are used for the various analyses reported here. Obviously, the analysis should not be carried out on all the projects simultaneously. To get a minimum of homogeneity in the samples to be analyzed, the following two criteria are taken into account: same functional sizing method and same programming language.

For the first criterium, projects measured with the IFPUG function points method have been selected since in ISBSG R8, close to 90% of the projects had been measured with the IFPUG method.

For the second criterium, the projects with the same programming language were grouped together in distinct samples. In ISBSG R8, there were only 6 programming languages with more than 30 projects, 30 being the number of points for considering a sample of a reasonable size for statistical purposes; only these samples were kept for further analysis. Table 1 presents the number of projects for each of the following programming languages with over 30 projects: COBOL, C, Visual BASIC, C++, SQL and Oracle[1]. For all other alternative programming languages within the ISBSG repository, there was an insufficient number of projects for our purposes.

Table 1. ISBSG R8 -Programming language with over 30 projects

Programming language	Number of projects
Cobol	413
C	139
Visual Basic	103
C++	101
SQL	90
Oracle	87
Total	933

3 Identification of Outliers

In Figure 1, the functional size in function points (FP) is on the X-axis and the effort in hours on the y-axis. Figure 1 is typical of data sets available in software engineering, that is with an increasing dispersion of data, (referred to as heteroscedasticity) [3,4,5].

A number of outliers can been observed in Figure 1, with either very high productivity while others have very low productivity for projects of equivalent size.

[1] These are the programming languages as recorded in the ISBSG repository. Some data collectors might have associated an environment (eg. ORACLE) to a programming language.

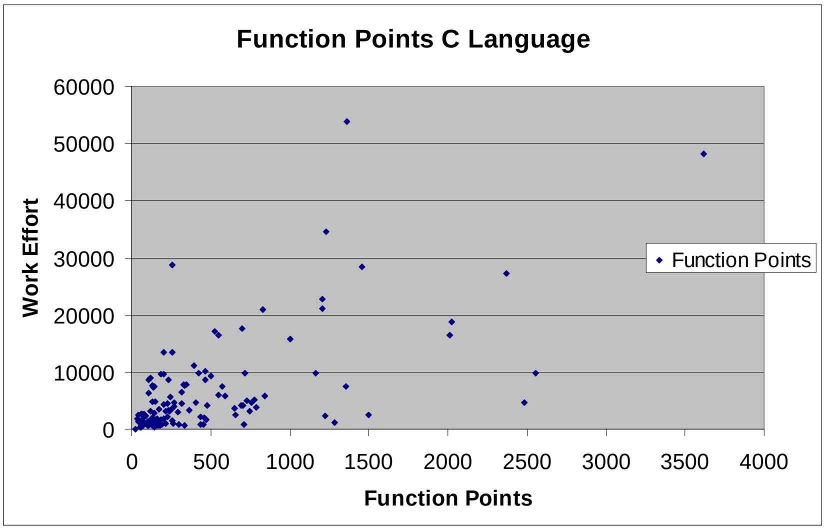


Figure 1. Data set with heteroscedasticity

Figure 2 points outsome projects in COBOL2 – R9 that have a large functional size with almost no corresponding effort: for illustrative purposes, seven (7) outliers were selected which appear to have very large economies of scale. These 7 outliers within a functional size range of 1000 to 2500 FP did not cost more than many projects 10 to 20 times smaller, thereby appearing to benefit from very large economies of scale (by a factor in the 10 to 20 range). The most probable cause is that there are some other variables that could explain such a minimal effort for such large size for these projects

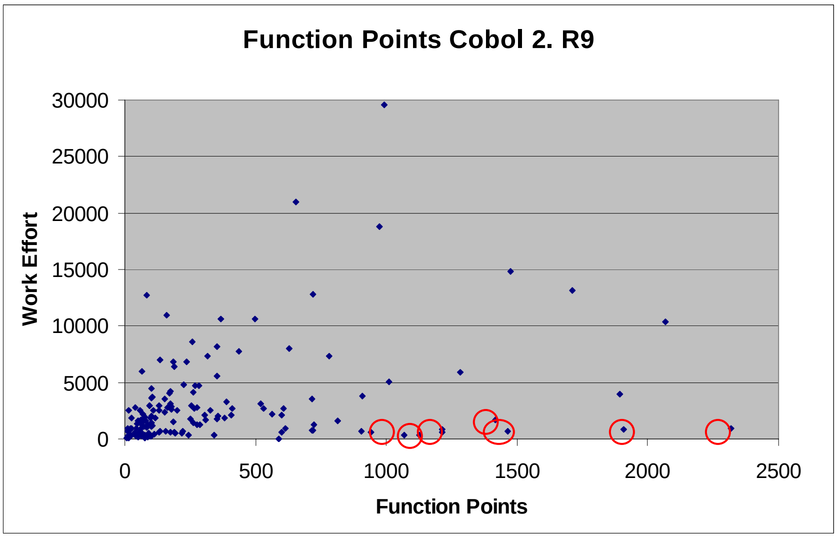


Figure 2. Visual identification of outliers with very large economies of scale

Figure 3 points out next to some projects in C language with large effort with relatively small functional size. Again for illustrative purposes, 3 outliers were selected that could qualify as having somewhat large dis-economies of scale, in particular for the outlier in the 300 FP range with a cost at least 10 times more than projects of similar size. The other two outliers identified graphically do not have such a large effort discrepancy, while still present.

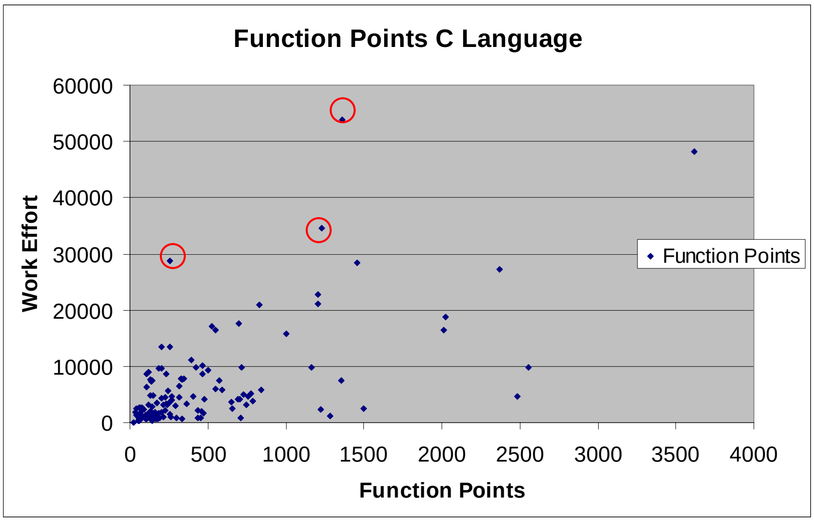


Figure 3. Visual identification of outliers with dis-economies of scale

4 Investigation of Outliers

Once the outliers identified, they are next compared to other projects of similar size or effort to explore if there exists patterns in the other variables recorded that might explain such outliers.

For the analysis of the ISBSG data repository, a good number of distinct tests selected by heuristics on some of the variables available in the repository were carried out on both R8 and R9 releases. In practice, only 8 tests gave results allowing a practical interpretation leading to the formulation of candidate hypotheses to be tested later with more robust statistical analyses.

4.1 Outliers with economies of scale

The analyses of the outliers with very large economies of scale are presented in tables 2 to 4, by programming language:

Table 2: COBOL - R8: 53 projects, including 10 outliers.
Table 3: C - R9: 118 projects, including 7 outliers. Table 4: COBOL2 - R9: 115 projects, 14 outliers.

In these tables, the variables tested by heuristics are on the left hand-side column, and the value most often observed in the outliers for such a variable tested, in the next column. The other two columns present the ratio of observations of this value over the samples, first within the subset of outliers, and finally within the sample to the exclusion of the outliers.

For instance, in Table 2 for COBOL projects, the first variable tested is the Data Quality Rating assigned to a project by the ISBSG repository manager. It can then be observed that the worst value for this variable, that is D = poor quality (column 2) is present in 10 out of 10 outliers (column 3) and only in 1 out of the other 43 projects Column 4) within the sample of projects in COBOL; that is 100% of the outliers have data considered of very poor quality, while only 2% of the other projects in COBOL have a poor data quality rating.

Table 2. Economies of scale: COBOL - R8 (N=53)

Variables tested	Value observed	Ratio of outliers	Ratio of non-outliers
Data Quality Rating	D	**10 / 10** (100%)	**1 / 43** 2,3%
Resource Level	2	**10 / 10** (100%)	**12 / 43** (28%)
Organization type	Insurance	**10 / 10** (100%)	**12 / 43** 28%
Reference table approach	Counted as inputs	**10 / 10** (100%)	**7 / 43** (16,3%)

In tables 2 to 4, several variables have been identified by heuristics as partially responsible for the outliers behavior in terms of project productivity ratios. The ISBSG definitions of these various variables are presented next:

- Data Quality rating: Quality of the data, as evaluated by the ISBSG repository manager.
- Resource Level: Personnel included in the recording of effort.
- Organization type: Type of organization which sent the data.
- Reference table approach: IFPUG Function Points version used to count the tables of codes in the software[2].
- Operating system: Operating system (O/S) on which the software measured runs.
- Primary database system: The main database management system (DBMS) for the software measured.

The values admissible for the "Data Quality Rating" are:

A = data submitted was assessed as being sound.

B = appears fundamentally sound but there are some factors which could affect the integrity of the data.

[2] This is a peculiarity of the IFPUG method: depending on which IFPUG version is selected for the measurement of Tables of code, there can be large differences in the number of Function Points.

C = Due to significant data not being provided, it was not possible to assess the integrity of the submitted data.

D = Due to one factor or a combination of factors, little credibility should be given to the submitted data.

The values admissible for the Resource Level are:

1 = development team only
2 = development + support teams
3 = development + support teams + operators
4 = development + support teams + operators + customers

In Table 2, all of the outliers share the same values for the 4 variables identified: they all (eg. 100%) have a poor data quality rating, their effort include hours for both direct development staff and support staff, are insurance projects and they have used for size measurement an IFPUG version that takes into account each code table.

For the non outliers (Table 2), these characteristics are much less frequent (from 2 to 28 % of the projects).

For the sample with the projects in C (Table 3), there are two candidate explanatory variables for the economies of scale: the AIX Operating System and Sysbase as the primary DBMS which appear in around 50% of the outliers, and only 4% of the non outliers.

Table 3. Economies of scale: C - R9 (N=118)

Variable tested	Valeur observed	Ratio of outliers	Ratio of non-outliers
Operating System	AIX	**3 / 7** (42,9%)	**4 / 89** (4,5%)
Primary Database System	Sybase	**4 / 7** (57,1%)	**4 / 111** (3,6%)

For the sample with the projects in COBOL2 (Table 4), there are again four candidate explanatory variables for the economies of scale: they are the same as for the C sample.

Table 4. Economies of scale: COBOL2 - R9 (N=115)

Variable tested	Value observed	Ratio of outliers	Ratio of non-outliers
Data Quality Rating	D	**13 / 14** (92,9%)	**8 / 101** (7,9%)
Resource Level	2	**14 / 14** (100%)	**36 / 101** (35,6%)
Organization type	Insurance	**14 / 14** (100%)	**21 / 101** (20,7%)
Reference table approach	Counted as inputs	**14 / 14** (100%)	**21 / 101** (20,7%)

4.2 Outliers with dis-economies of scale

The results of the analyses of the outliers with dis-economies of scale, that is with a very high effort for comparable projects of smaller functional size, are presented in Tables 5 to 9.

Table 5: C - R8: 40 projects, 6 outliers

Table 6: Java - R9: 24 projects, 4 outliers

Table 7: COBOL - R8: 412 projects, 7 outliers

Table 8: C - R9: 16 projects, 4 outliers

Table 9: SQL - R9: 26 projects, 4 outliers.

In tables 5 to 9, four additional variables have been identified by heuristics as partially responsible for the outliers' behavior in terms of project productivity ratios. The ISBSG definitions of these variables are presented next:

- Standard FP: IFPUG standard used to count the points of function.
- Max TEAM size: Maximum number people who worked on the project at the same time (peak time).
- Lines of code: Number of lines of source code produced by the project.
- Project elapsed time: Duration, in months, to complete the development of the project.

In Table 5 for the C sample, the two most discriminative variables for dis-economies of scale are the Max team size greater than 10 people and Lines of code greater than 100 000, that is projects of relatively large size when compare to the full sample of C projects.

Table 5. Dis-economies of scale: C - R8 (N=40)

Variable tested	Value observed	Ratio of outliers	Ratio of non-outliers
Data Quality Rating	B	**6 / 6** (100%)	**24 / 34** (70,6%)
FP Standard	CPM 4.0	**3 / 6** (50%)	**7 / 34** (20,6%)
Max team size	> 10	**4 / 6** (66,7%)	**4 / 34** (11,8%)
Lines of code	> 100 000	**2 / 6** (33,3%)	**2 / 34** (5,8%)

In Tables 6 and 7 for the Java and COBOL samples, a single discriminative variable has been identified by heuristics for dis-economies of scale for both COBOL and C samples, that is, projects with a Max team size greater than 10 people.

Table 6. Dis-economies of scale: Java - R9 (N=24)

Variable tested	Value observed	Ratio of outliers	Ratio of non-outliers
FP Standard	IFPUG 4	**4 / 4** (100%)	**2 / 20** (10%)

Table 7. Dis-economies of scale: COBOL - R8 (N=412).

Variable tested	Value observed	Ratio of outliers	Ratio of non-outliers
Max team size	> 10	**5 / 7** (71,4%)	**27 / 405** (6,7%)

Table 8. Dis-economies of scale: C - R9 (N-16)

Variable tested	Value observe	Ratio of outliers	Ratio of non-outliers
Max team size	> 10	3 / 4 (75%)	3 / 12

Finally, in Table 9 for the SQL sample, the two most discriminative variables for dis-economies of scale are a resource level that includes staff in addition to the development and support teams and a project elapsed time of over 15 months in duration.

Table 9. Dis-economies of scale: SQL - R9 (N=26)

Variable tested	Value observed	Ratio of outliers	Ratio of non-outliers
Resource Level	> 2	3 / 4 (75%)	1 / 22 (4,5%)
Project Elapsed time	> 15 months	3 / 4 (75%)	2 / 22 (9,1%)

5 Summary & Discussion

This paper has discussed the issue of outliers in the repository of software projects of the International Software Benchmarking Standards Group - ISBSG. The criteria used for the identification of outliers is whether the productivity is significantly lower and higher in relatively homogeneous samples, that is projects with significant economies or dis-economies of scale. Once the outliers identified, other project variables were investigated by heuristics to identify candidate explanatory variables that might explain such outliers' behaviors.

Candidate variables identified as potentially related to large economies of scale in the ISBSG repository for some programming languages have been identified as: resource level 2, insurance as the organization type and the peculiarity of the Reference table approach in the IFPUG Function Points sizing method. The D rating for the data quality assigned to the outliers project is a somewhat confounding factor: it is not a data collected by an organization, but rather a judgment of the ISBSG repository manager who has indeed identified an unusual effort relationship with respect to size, but which does not provide any clue into the whys of such a pattern nor does it provide confirmation that the data is erroneous.

Candidate variables identified as potentially related to large dis-economies of scale in the ISBSG repository for some programming languages have been identified as: maximum team size larger than 10 people, lines of code greater than 100 000, project duration greater than 15 months and effort data which includes not only development and support staff, but as well operators and customers project related effort. The specific version of the IFPUG Function Points method is also a variable identified as a candidate explanatory variable.

Of course, this list of candidate explanatory variables is far from being exhaustive: further research is required on the one hand for more robust methods for identifying in a systematic manner the outliers and, on the other hand, for investigating causes of such outliers' behaviors. Such further research will be challenging and time consuming.

Practitioners, however, can derive immediate benefits from this exploratory research in the following way: monitoring of the candidate explanatory variables can provide valuable clues for early detection of potential project outliers for which most probable estimates should be selected not within a close range of values predicted by an estimation model, but rather at their upper or lower limits: that is, the selection of either the most optimist or most pessimist value that can be predicted by the estimation model being used.

References

[1] ISBSG, *Estimating, Benchmarking & Research Suite Release 8 & 9*, International Software Benchmarking Standards Group – ISBSG, Australia, 2005.

[2] ISBSG, International Software Benchmarking Standards Group, www.isbsg.org

[3] B.A Kitchenham, N.R. Taylor, "Software Cost Models", *ICL technical journal*, Vol. 4, no 1, May 1984, pp. 73-102., B.,

[4] A. Abran, P.N. Robillard, "Function Points Analysis: An Empirical Study of its Measurement Processes," *IEEE Transactions on Software Engineering*, Vol. 22, no 12, 1996, pp. 895-909.

[5] A. Abran, I. Silva, L. Primera, "Estimation Models for Functional Maintenance Projects – Field Studies", in Journal of Software Maintenance: Research and Practice, Vol. 14, 2002, pp. 31-64.

Software e-Measurement in the WWW

Reiner R. Dumke, Martin Kunz, Antje Riekehr

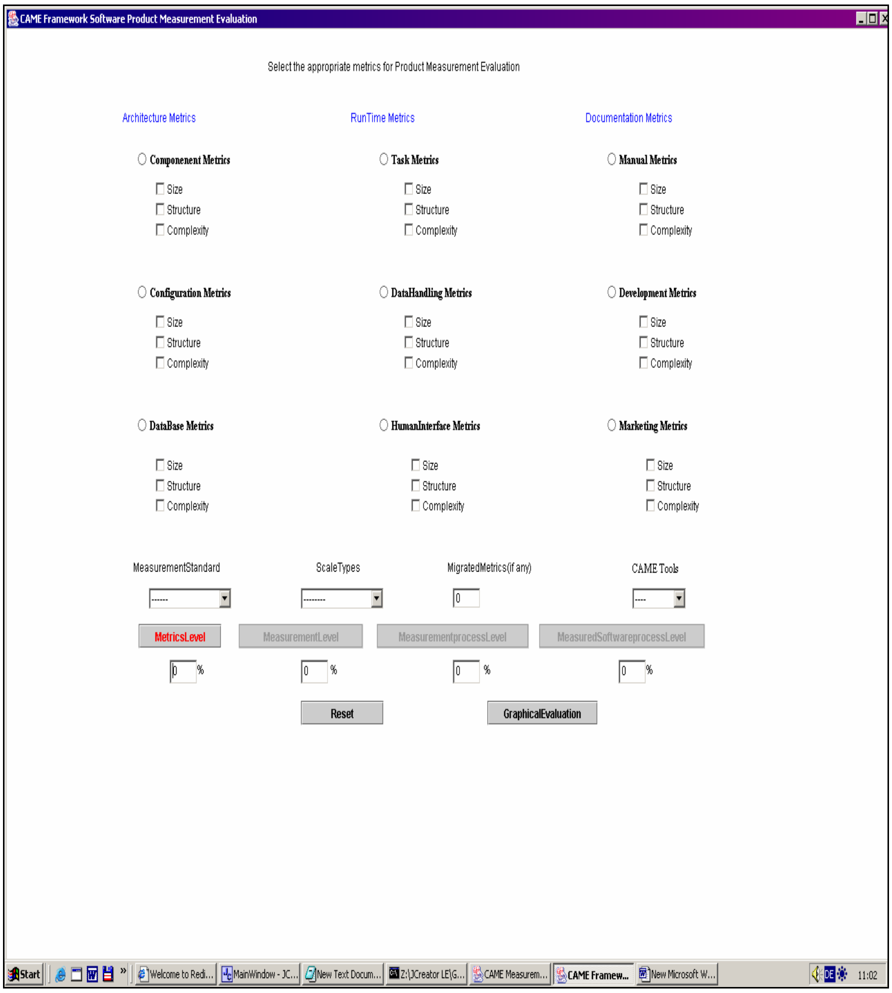
University of Magdeburg, Germany
http://ivs.cs.uni-magdeburg.de/sw-eng/agruppe/

The following overview includes some metrics tools and informations about implemented software measurement methods in the World Wide Web. The structure of this survey is based on the different themes and measurement approaches in an alphabetic order.

1 CAME-based Measurement Evaluation

CAME stands for *Choice, Adjustment, Migration* und *Efficiency* and summarizes an apporach in order to evaluate the software measurement level in a chosen (IT) area.

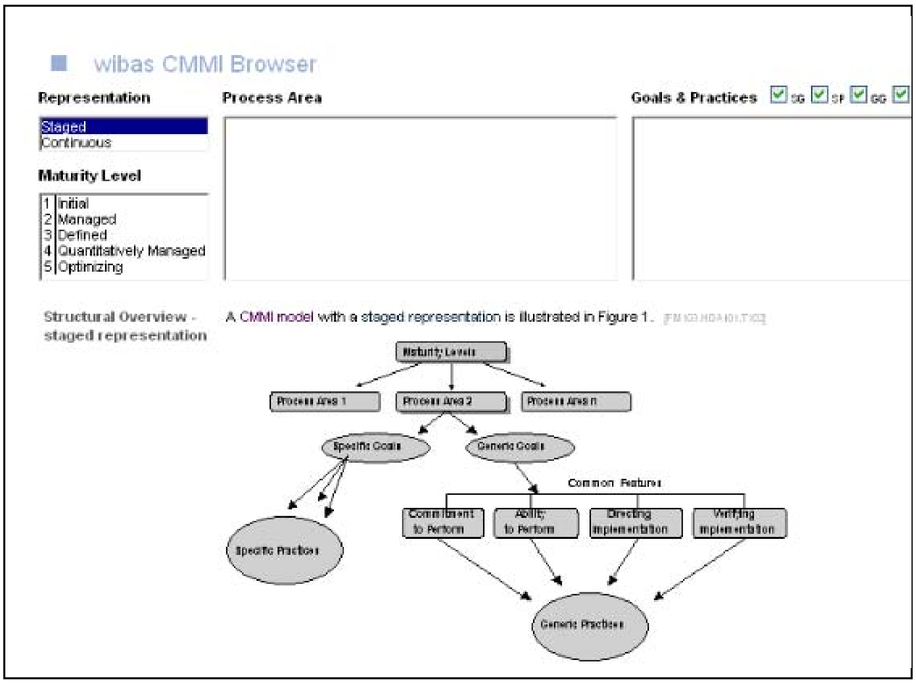
`http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/`



2 CMMI (Capability Maturity Model Integration)

- ***Description of the CMMI:***

  http://www.wibas.de/cmmibrowser/index.php

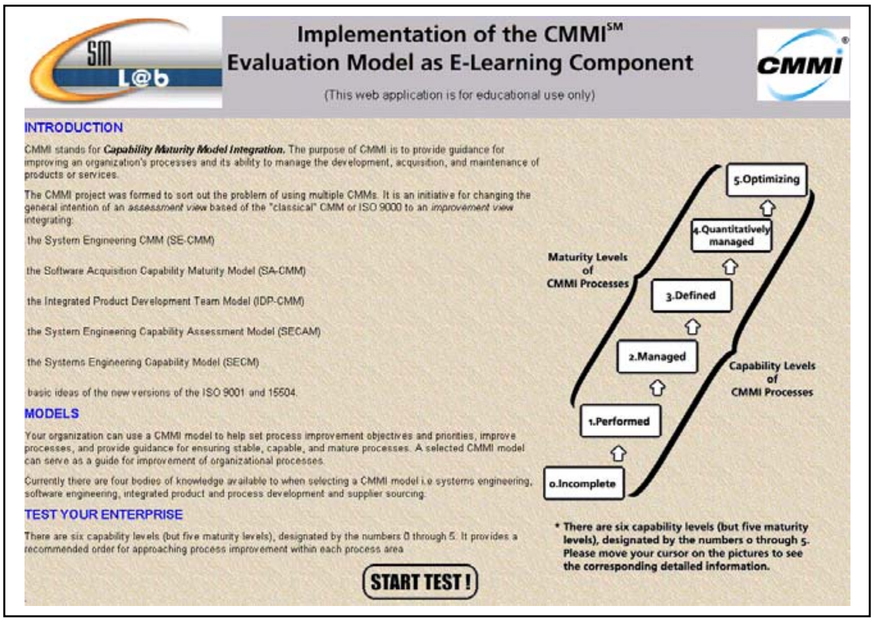


- ***CMMI Overview:***

  http://www.betterproductdesign.net/maturity.htm

- ***CMMI Exercise:***

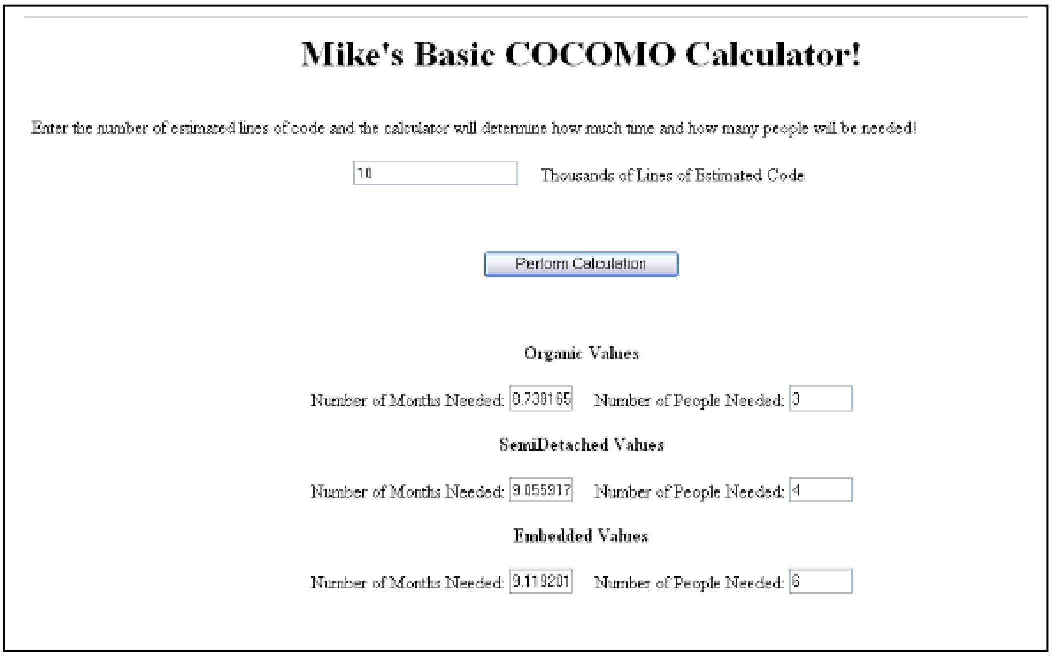
  http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/



3 COCOMO (Constructive Cost Model)

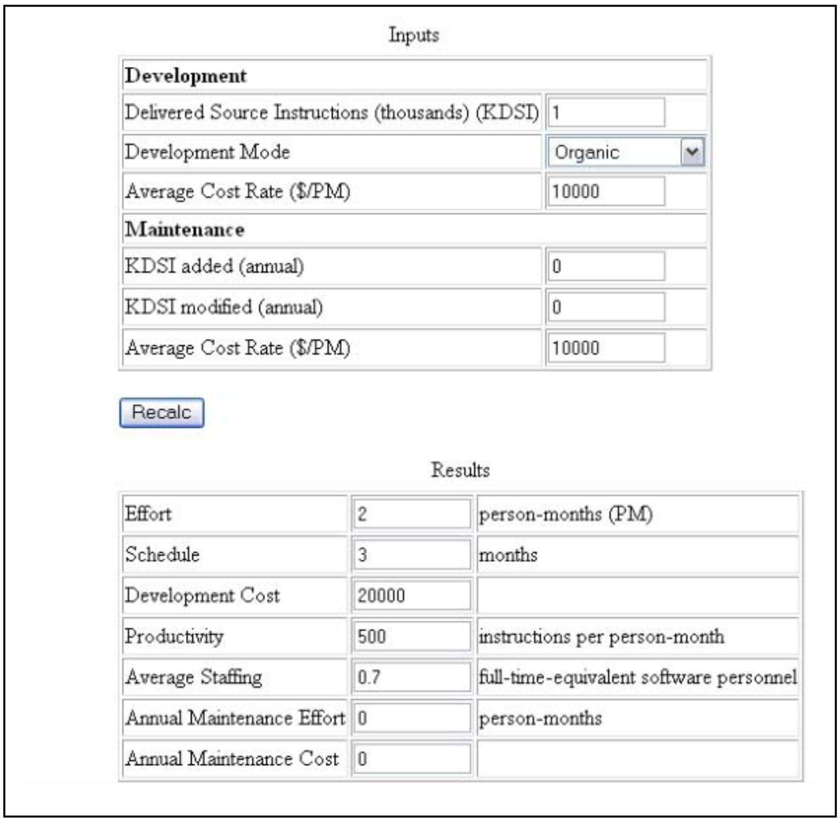
- ***COCOMO Calculator:***

  http://www.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/baker/cocomo.html

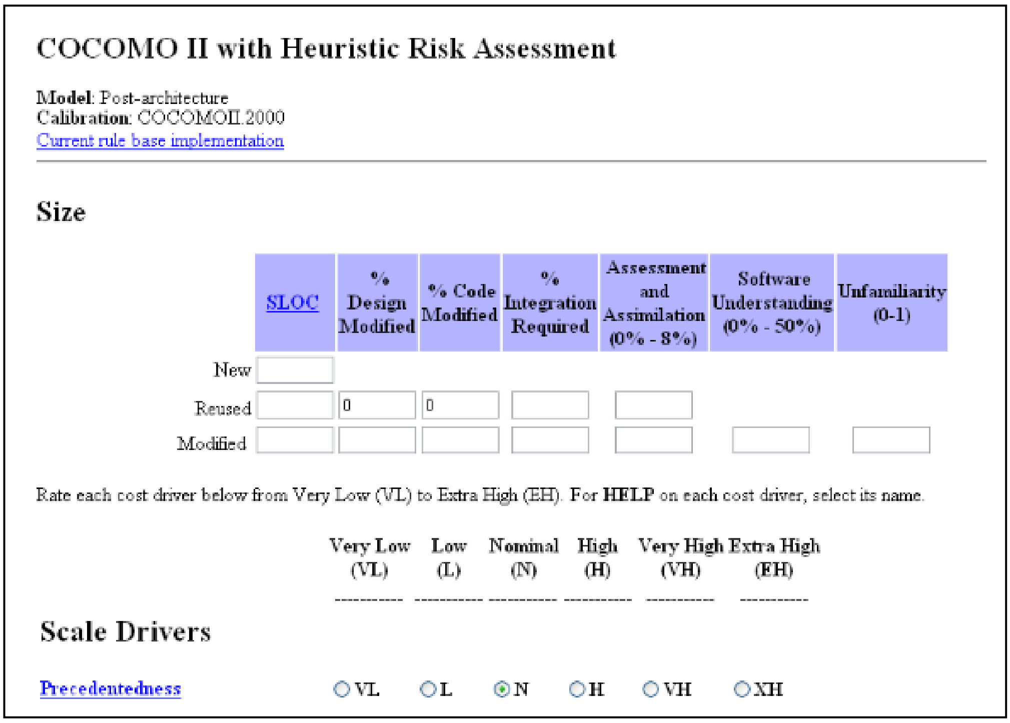


- ***COCOMO Estimation:***

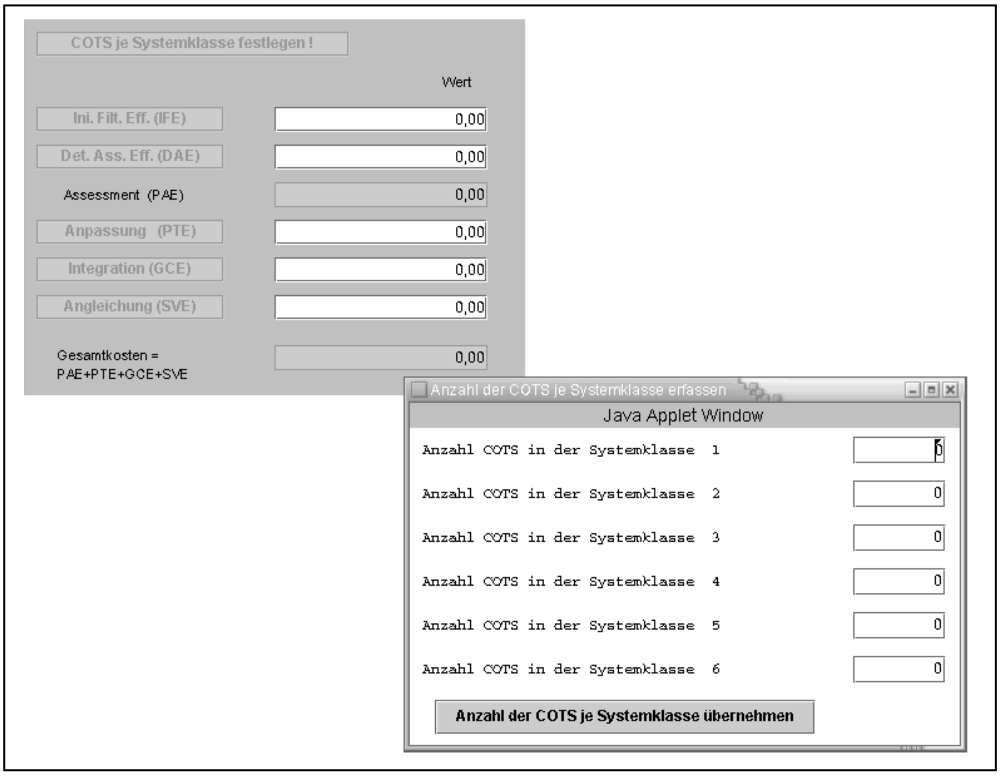
  http://www1.jsc.nasa.gov/bu2/COCOMO.html



- ***COCOMO II extension:***
  http://sunset.usc.edu/research/COCOMOII/expert_cocomo/expert_cocomo2000.html



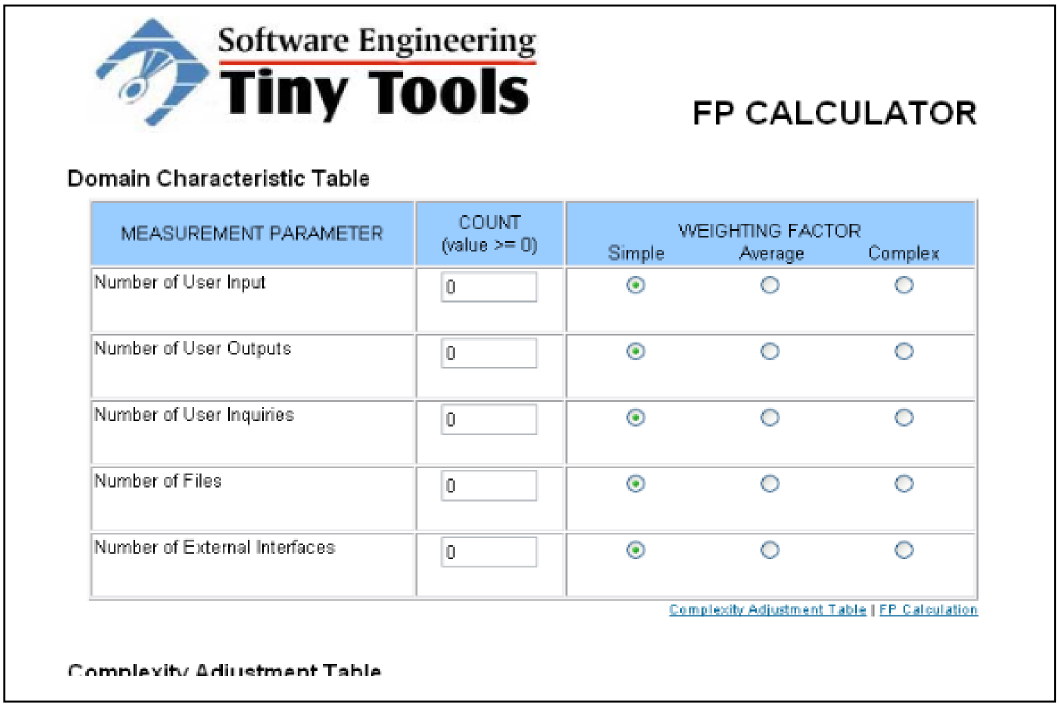
- ***COCOTS-based estimation:***
  http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/



4 Functional Size Measurement (FSM)

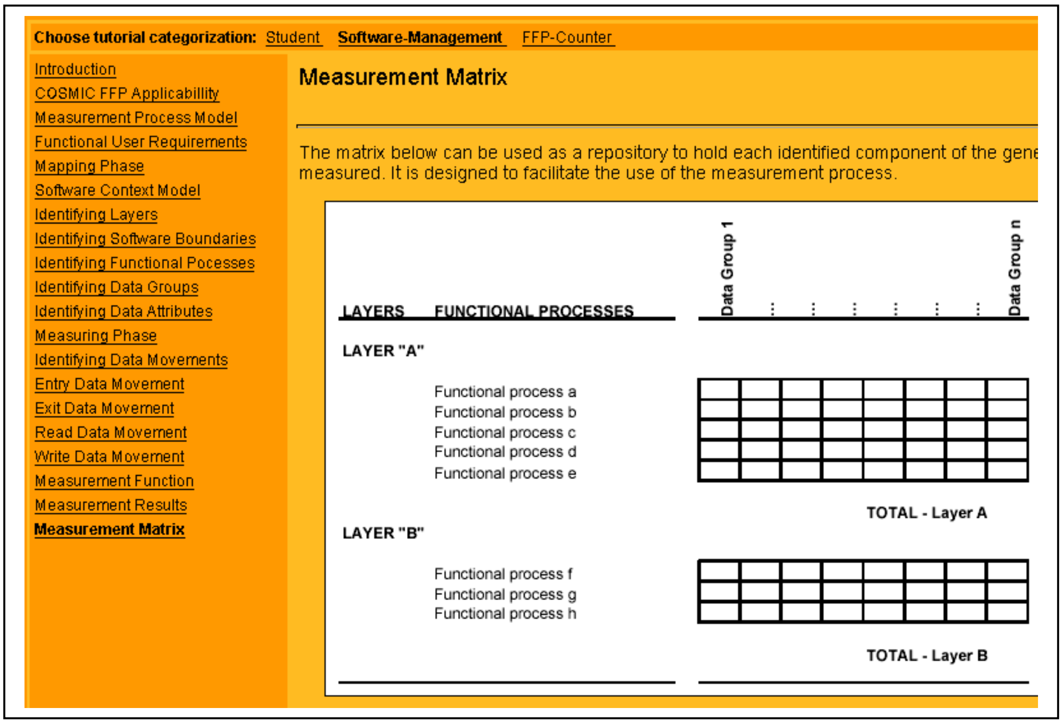
- ***Function Point Calculator:***

  http://www.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/harvey/FP_Calc.html



- ***COSMIC FFP:***

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



- ***Information about Function Points:***

Function Points FAQ:

http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm

Function Points Overview

http://yunus.hun.edu.tr/%7Esencer/size.html

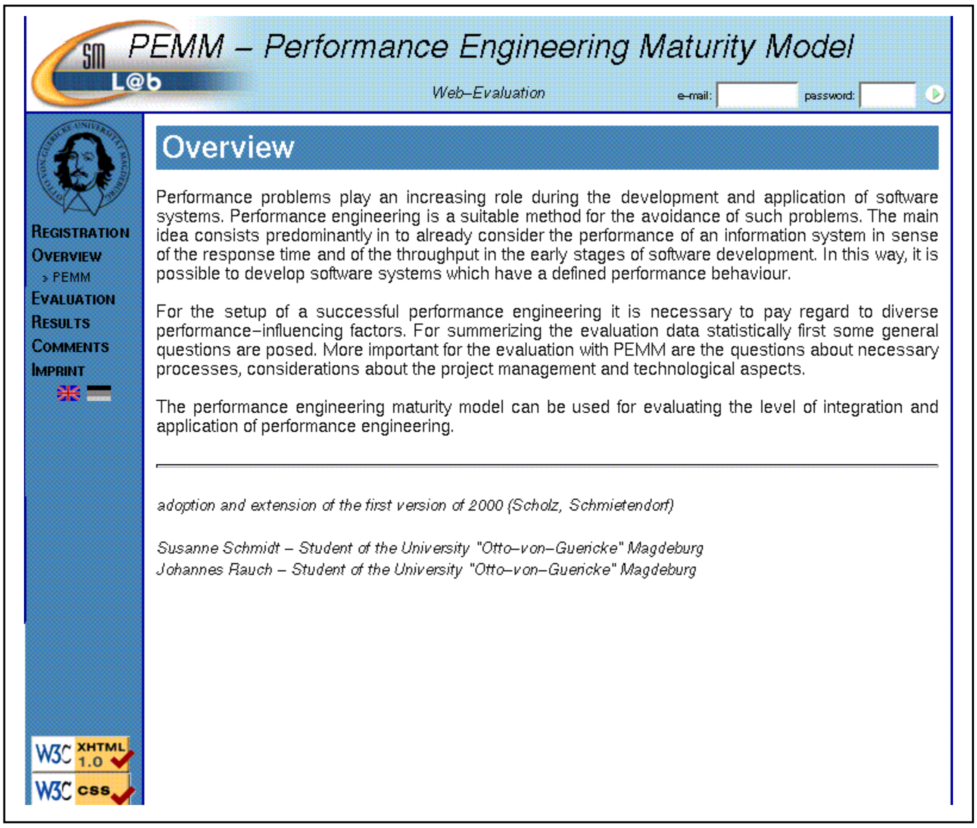
Overview about Functional Size Measurement

http://www.geocities.com/ResearchTriangle/Campus/6025/fpa/fpa.htm#p0

5 PEMM (Performance Engineering Maturity Model)

Please consider the new implementation of the PEMM-based evaluation in the WWW at the SML@b.

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



Ebert, C.:

Systematisches Requirements Management
Anforderungen ermitteln, spezifizieren, analysieren und verfolgen

dpunkt.verlag, August 2005 (320 Seiten)
ISBN 3-89864-336-0

Projekte scheitern häufig wegen unzureichendem Requirements Management. Meist waren schon zu Beginn die Anforderungen nicht ausreichend geklärt und damit konnte auf deren Änderungen auch nicht richtig reagiert werden. Das Buch bietet einen Überblick über Theorie und Praxis des Requirements Management. Es beschreibt, wie Anforderungen entwickelt, gesammelt, dokumentiert und im Projekt verfolgt werden. Die grundsätzlichen Methoden, Verfahren, Werkzeuge und Notationen des Requirements Management werden übersichtlich behandelt. Sie werden durch konkrete Beispiele aus der Projektarbeit illustriert.

Als Beispiel einer modernen Methode der Anforderungsbeschreibung werden Use-Case-Szenarien in der UML-Notation verwendet. Praktische Fallstudien unterstützen die konkrete Umsetzung.

Leser: Produktmanager, Projektleiter, Softwareentwickler

Weitere Informationen finden Sie unter

`http://www.dpunkt.de/buch/3-89864-336-0.html`

Sneed, H.M.:

Software-Projektkalkulation – Wissen was Projekte wirklich kosten

Hanser-Verlag, 2005 (228 Seiten)
ISBN 3-446-40005-2

Wer einmal die Kosten oder die Zeit für ein Software-Projekt falsch kalkuliert hat, weiß, dass kein Unternehmen sich das öfter leisten kann. Projektkalkulation ist eine Überlebensfrage der Software-Industrie. Für Auftragnehmer wie für Auftraggeber ist die richtige Kalkulation unabdingbar für den Projekterfolg.

Die meisten Techniken für Aufwandsschätzung, die in der Praxis verbreitet sind, eignen sich nur bei IT-Projekten für eine Neuentwicklung. Geht es in Ihrem Projekt jedoch um Wartung, Migration, Integration oder Sanierung, so müssen Sie darauf abgestimmte Methoden einsetzen. Dieses Buch zeigt, welche Techniken der Aufwandsschätzung für welche Art von Projekten zu nutzen sind.

Deek, F.P.; McHugh, J.A.M.; Eljabiri, O.M.:

Strategic Software Engineering

Auerbach Publications, 2005 (333 pages)
ISBN 0-8493-3939-1

Strategic Software Engineering: An Interdisciplinary Approach presents software engineering as a strategic, business-oriented, interdisciplinary endeavour, rather than simply a technical process, as it has been described in previous publications.

The book addresses technical, scientific, and management aspects of software development in a way that is accessible to a wide audience. It provides a detailed, critical review of software development models and processes, followed with a strategic assessment of how process models evolved over time and how to improve them. The authors then focus on the relation between problem-solving techniques and strategies for effectively confronting real-world business problems. They also analyze the impact of interdisciplinary factors on software development, including the role of people and business economics. The book concludes with a brief look at specialized system development.

The diverse backgrounds of the authors, encompassing computer science, information systems, technology, and business management, help create this book's integrated approach, which answers the demand for a comprehensive, interdisciplinary outlook that covers all facets of how software relates to an organization.

Contents:

Provides a detailed, critical review of software development models and processes, introduces and critiques the basic software development process and key risk-reduction models, Examines the theme of process improvement and explores recent trends in software process models, relates classic problem-solving concepts to software development and addresses how software tools influence problem-solving, explains how the focus of development has shifted from technical to business contexts, discusses people-related drivers for development, focuses on the role of costs and economics in software engineering.

El Emam, K.:

The ROI from Software Quality

Auerbach Publications, 2005 (279 pages)
ISBN 0-8493-3298-2

The ROI from Software Quality provides the tools needed for software engineers and project managers to calculate how much they should invest in quality, what benefits the investment will reap, and just how quickly those benefits will be realized. This text provides the quantitative models necessary for making real and reasonable calculations and shows how to perform ROI analysis before and after implementing a quality program. The book demonstrates how to collect the appropriate data and easily perform the appropriate ROI analysis.

Taking an evidence-based approach, this book supports its methodology with large amounts of data and backs up its positioning with numerous case studies and straightforward return-on-investment calculations. By carefully substantiating arguments numerically, this volume separates itself from other works on ROI.

Contents:

Explores options that allow managers to prioritize in pursuit of quality, enables quality benchmarking by including referenced examples of quality practices and implementations, explains in detail how to justify ROI calculations, delivers concrete data on the benefits of specific software engineering practices, provides a comprehensive analysis of the quality and security of open source software (OSS), includes implementation guidelines for using ROI within a software development organization, contains more than 200 tables for easy reference.

Abran, A.; Bundschuh, M.; , Büren, G.; Dumke, R. (Eds.):

Software Measurement – Research and Application

Springer Publ., Aachen, 2004 (602 pages)
ISBN 3-8322-3383-0

This proceedings of the joined conferences, the 14th International Workshop on Software Measurement (IWSM 2004) and the DASMA MetriKon 2004, try to reflect a bit of all the concepts developed and the experiences made when measuring software. They are of particular interest to software engineering researchers, as well as to practitioners, in the areas of project management and quality improvement programs, for both software maintenance and software development.

Ebert, C.; Dumke, R.; Bundschuh, M.; Schmietendorf, A.:

Best Practices in Software Measurement

Springer Publ., 2004 (320 pages)
ISBN 3-540-20867-4

The software business is challenging enough without having to contend with recurring errors. One way repeating errors can be avoided is through effective software measurement. In this book is offered a practical guidance built upon insight and experience. The authors detail knowledge and experiences about software measurement in an easily understood, hands-on presentation and explain many current ISO standards (see also `http://metrics.cs.uni-magdeburg.de/`).

Chrissis, M.B.; Konrad, M.; Shrum, S.:

CMMI – Guidelines for Process Integration and Product Improvement

Addison-Wesley, 2004 (663 pages)
ISBN 0-321-15496-7

This book is the definitive reference for the most current release of CMMI models. To use a CMMI model available on the SEI Web site, users must choose from among multiple models based on their organization's improvement needs. This book provides a single source for all CMMI model information. Readers can get started without having to select a model first – all of the choices are compiled in one place and explained in detail.

The book begins with background information needed to understand the content and structure of these integrated models and how to use them. A case study illustrates their implementation in a real environment. A variety of practical material, such as glossary and index, is also provided. The bulk of the book comprises the content of all CMMI models, covering the 25 process areas (PAs) that span the product life cycle, including detailed best practices.

Tayntor, C.B.:

Six Sigma Software Development

Auerbach Publications, 2003 (322 pages)
ISBN 0-8493-1193-4

Six Sigma Software Development illustrates how Six Sigma concepts can be applied to all aspects of the evolving system development process, including the traditional waterfall model, support of legacy systems, and also the more recent development innovations such as rapid application development, packaged software implementation, and outsourcing. A primary focus is placed on eliminating defects and improving customer satisfaction through the use of tools that help to ensure that requirements are clearly defined, understood, and met.

The volume begins with a basic primer on Six Sigma and uses a case study to provide a clear explanation of its concepts and applications. It then explains the relevance of Six Sigma to the system development process, quality assurance, and the Software Engineering Institute Capability Maturity Model (SEI CMM) – mapping the concepts and tools to all aspects of application development. Finally, Six Sigma Software Development shows how Six Sigma can be used for more then a single project, in that the concepts can be applied to measure, manage, and improve the performance of your entire IT department.

Contents:

Provides a clear, concise explanation of Six Sigma concepts and their application supported by a case study, present an informed, realistic proposal for the use of Six Sigma tools to evaluate the overall performance of the IT department, explores the relevance of Six Sigma to the system development process and to quality assurance, mapping the concepts and tools to all aspects of development.

Keyes, J.:

Software Engineering Handbook

Auerbach Publications, 2003 (874 pages)
ISBN 0-8493-1479-8

With decreasing software budgets and increasing demands from users and senior management, technology directors need a complete guide to the subject of software engineering. The successor to the bestselling Software Engineering Productivity Handbook, the Software Engineering Handbook fills this need.

Written by an expert with over 25 years of practical experience in building systems, the text covers the full spectrum of software engineering methodologies, techniques, and tools and provides details on how to reach the goals of quality management in a software engineering environment. It includes a wide variety of information, from guidelines for the Malcom Baldrige Quality Award to the IEEE measures for reliable software. Sixty-five field-tested how-to chapters provide techniques, guidelines, and philosophies that will assist developers in implementing quality and productivity programs.

The author supplies a wealth of information and advice in a multitude of areas including management of resources, methods, quality, and metrics. The book concludes with 19 appendices filled with guides, templates, forms, and filled-out examples that illustrate important software engineering techniques such as: software requirement specification, software design specification, and a complete test plan including use of automated estimation tools.

Kütz, M.:

Kennzahlen in der IT

dpunkt.verlag, 2003 (312 Seiten)
ISBN 3-89864-225-9

Moderne Management- und Steuerungsansätze basieren auf Kennzahlen und Kennzahlensystemen. Dies gilt auch im IT-Controlling.

Dieses Buch entwirft ein zeitgemäßes Kennzahlenverständnis für den IT-Bereich. Vor dem Hintergrund der Balanced Scorecard, dem wichtigsten methodischen Ansatz für die strategische Steuerung von Organisationen, werden die vorhandenen Ansätze in verschiedenen Teilbereichen sowie unterschiedliche Kennzahlensysteme aus der Literatur kritisch gewürdigt, systematisiert und zusammengefasst. Dabei geht es vor allem um die Steuerung von IT-Dienstleistern, weniger um technische IT-Systeme.

Betriebswirtschaftliche, kosten- und leistungsbezogene Aspekte dominieren. Vor allem Fragestellungen des Benchmarkings, also des organisationsübergreifenden Leistungsvergleichs, spielen eine wichtige Rolle. Als konkretes Ergebnis wird der Versuch eines neuen Kennzahlenkanons für IT-Organisationen gewagt.

Das Buch wendet sich an Manager und Controller in der Praxis, die konkret mit der Bewertung von IT-Systemen oder –Dienstleistungen befasst sind.

Wells, T.D.:

Dynamic Software Development

Auerbach Publications, 2003 (246 pages)
ISBN 0-8493-1292-2

Dynamic Software Development: Managing Projects in Flux defines the principles, practices, skills, and techniques needed to manage a dynamic software development environment. At a hands-on level, the book helps managers define the project goal and the actual situation, plan progress, manage developers, and monitor productivity. At a higher level, the book helps managers determine a strategic framework, ease workflow in the development environment, obtain funding, increase economic return, and implement leadership by consensus.

Targeted at those who manage information systems, corporate information, and developers, the book features a section at the end of each chapter to help them apply and customize the recommended techniques to their specific organizations. This reference addresses recent approaches to building applications such as Extreme Programming, Adaptive Software Development, and “lightweight” methodologies. By noting the failure of similar techniques I the past, the author emphasizes that such ideas can only achieve their true potential via the common, consistent management techniques outlined in Dynamic Software Development.

Contents:

Offers a detailed, specific guide for managing the entire development effort, including new development and software maintenance, supplies a strategy for managers to effectively control the development effort without imposing artificial and burdensome constraints on the developer, illustrates how to customize and apply the techniques provided to each specific organization, includes effective, non-linear management techniques that can respond to the intensifying demand for Web-based applications, and the growing pressure on development teams to improve their offering.

Preprints/Technical Reports:

Dumke, R.; Coté, I.; Andruschak, O.: *Statistical Process Control (SPC) – A Metrics-Based Point of View of Software Processes Achieving the CMMI Level Four*. University of Magdeburg, 2004

April, A.A.; Dumke, R.R.; Abran, A.: *SM^{mm} Model to Evaluate and Improve the Quality of the Software Maintenance Process*. University of Magdeburg, 2004

Dumke, R.; Schmietendorf, A.; Zuse, H.: *Formal Description of Software Measurement and Evaluation*. University of Magdeburg, 2005

Braungarten, R.; Kunz, M.; Dumke, R.: *An Approach to Classify Software Measurement Storage Facilities*. University of Magdeburg, 2005

see as pdf files:

`http://ivs.cs.uni-magdeburg.de/sw-eng/agruppe/forschung/Preprints.shtml`

IASTED SE 2005:

IASTED International Conference on Software Engineering 2005
February 15-17, 2005, Innsbruck, Austria
see: `http://www.iasted.org/conferences/2005/innsbruck/se.htm`

SEPG 2005:

17th Software Engineering Process Group Conference
March 7-10, 2005, Seattle, Washington
see: `http://www.sei.cmu.edu/sepg/main.htm`

SMEF 2005:

Software Measurement European Forum
March 16-18, 2005, Rome, Italy
see: `http://www.iir-italy.it/smef2005/`

CSMR 2005:

9th European Conference on Software Maintenance and Reengineering
March 21-23, 2005, Manchester, UK
see: `http://www.rcost.unisannio.it/csmr2005/index2.html`

EASE 2005:

9th International Conference on Empirical Assessment in Software Engineering
April 11-13, 2005, Staffordshire, UK
see: `http://ease.cs.keele.ac.uk/call_for_papers.html`

ASQT 2005:

Arbeitskonferenz Softwarequalität und Test 2005
April 27-29, 2005, Klagenfurt, Austria
see: `http://www.ifi.uni-klu.ac.at/Conferences/ASQT2005/`

SPICE 2005:

5th International SPICE Conference on Process Assessment and Improvement
April 28-29, 2005, Klagenfurt, Austria
see: `http://www.ifi.uni-klu.ac.at/Conferences/SPICE2005`

PQST 2005:

International Conference on Practical Software Quality & Testing
May 2-6, 2005, Las Vegas
see: `http://www.psqtconference.com/2005west/`

WMM 2005:

1st Workshop on Web Measurement and Metrics
May 10, Chiba, Japan
see: `https://www.cs.auckland.ac.nz/wmm05/`

WWW 2005:

International World Wide Web Conference
May 10-14, 2005, Chiba, Japan
see: `http://www2005.org/`

IWPC 2005:

International Workshop on Program Comprehension
May 15-16, 2005, St. Louis
see: `http://www.ieee-iwpc.org/iwpc2005/`

PROMISE 2005:

International Workshop on Predictor Models in Software Engineering
May 16, 2005, St. Louis, Missouri
see: `http://promise.site.uottawa.ca/`

ICSE 2005:

3rd Workshop on Software Quality
May 17, 2005, St. Louis, Missouri
see: `http://attend.it.uts.edu.au/icse2005/`

REBSE 2005:

Workshop on Realising Evidence-Based Software Engineering
May 16, 2005, St. Louis, Missouri
see: `http://evidence.cs.keele.ac.uk/rebse.html`

SIGMetrics 2005:

ACM SIGMetrics - Performance 2005
June 6-10, 1005, Banff, Alberta, Canada
see: `http://www.cse.cuhk.edu.hk/~sigm2005/`

PE2005:

6. Workshop Software Performance Engineering
10. Juni 2005 in Berlin,
see: `http://ivs.cs.uni-magdeburg.de/~gi-peak/`

ESEPG 2005:

10th European Software Engineering Process Group Conference
June 13-16, London, UK
see: `http://www.espi.org/sepg/`

PROFES 2005:

6th International Conference on Product Focused Software Process Improvement
June 13-16, 2005, Oulu, Finland
see: `http://profes2005.oulu.fi/`

QAST 2005:

1st Workshop on Quality Assurance and Software Testing
June 27-30, 2005, Las Vegas
see: `http://people.cs.und.edu/~reza/QAST05.htm`

WOSP 2005:

5th International Workshop on Software & Performance
July 11-15, 2005, Las Palmas, Spain
see: `http://wosp2005.uib.es/`

QATWBA 2005

2nd International Workshop on Quality Assurance and Testing of Web-Based Applications
July 25-28, 2005, Edinburgh, UK
see: `http://aquila.nvc.cs.vt.edu/compsac2005/`

ICWE 2005:

5th International Conference on Web Engineering
July 25-29, 2005, Sydney, Australia
see: `http://www.icwe2005.org/`

SPPI 2005:

Software Process and Product Improvement - 31th Euromicro Conference
August 30 - September 3, 2005, Porto, Portugal
see: `http://www.sea.uni-linz.ac.at/SPPI2005/`

IWSM 2005:

15th International Workshop on Software Measurement
September 12-14 in Montreal, Canada
see: `http://www.lrgl.uqam.ca/workshops/iwsm2005/`

QFD 2005:

17th Symposium on Quality Function Deployment
September 15-23, 2005, Portland, Oregon
see: `http://www.qfdi.org/call_for_papers.htm`

METRICS 2005:

10th International Symposium on Software Metrics
September 19-22, 2005, Como, Italy
see: `http://metrics2005.di.uniba.it/`

QSIC 2005:

International Conference on Software Quality
September 19-21, Melbourne, Australia
see: `http://www.ict.swin.edu.au/conferences/qsic2005/`

ISMA 2005:

1th Annual International Software Measurement and Analysis Conference
September 18-23, 2005, New Orleans
see: `http://www.ifpug.org/press/2005ConferenceAnnouncement.htm`

SOQUA 2005:

International Conference on Software Quality
September 19-22, Erfurt, Germany
see: `http://www.mathematik.uni-ulm.de/sai/jmayer/soqua05/`

QEST 2005:

International Conference on Quantitative Evaluation of SysTems
September 19-22, Torino, Italy
see: `http://www.qest.org/`

3WCSQ 2005:

World Congress on Software Quality
September 26-30, 2005, Munich, Germany
see: `http://www.isqi.org/isqi/eng/conf/wcsq/3/`

UKSMA 2005:

16th Annual UKSMA Conference - Managing your Software (through Measurement)
October 13, 2005, London, UK
see: `http://www.uksma.co.uk/`

MetriKon 2005:

DASMA Workshop
November 15-16, 2005, Kaiserslautern
see: `http://www.metrikon.de/`

ISESE 2005:

ACM-IEEE 4th International Symposium on Empirical Software Engineering
Nov 17-18, 2005, Noosa Heads, Australia
see: `http://attend.it.uts.edu.au/isese2005/cfp.htm`

see also: **OOIS, ECOOP** and **ESEC** European 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://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://www.lrgl.uqam.ca/**
  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 of 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 have 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

- **http://www.qsm.com/**
  Homepage of the Quantitative Software Management (QSM) in the Netherlands

- **http://www.iese.fhg.de/**
  Homepage of the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany
- **http://www.highq.be/quality/besma.htm**
  Homepage of the Belgian Software Metrics Association (BeSMA) in Keebergen, Belgium

- **`http://www.cetus-links.org/oo_metrics.html`**
  Homepage of Manfred Schneider on Objects and Components

- **`http://dec.bournemouth.ac.uk/ESERG/bibliography.html`**
  An annotated bibliography of object-oriented metrics of the Empirical Software Engineering Research Group (ESERG) of the Bournemouth University, UK

News Groups

- **`news:comp.software-eng`**
- **`news:comp.software.testing`**
- **`news:comp.software.measurement`**

Software Measurement Associations

- **`http://www.dasma.org`**
  DASMA Deutsche Anwendergruppe für SW Metrik und Aufwandsschätzung e.V.

- **`http://www.aemes.fi.upm.es`**
  AEMES Association Espanola de Metricas del Software

- **`http://www.cosmicon.com`**
  COSMIC Common Software Measurement International Consortium

- **`http://www.esi.es`**
  ESI European Software Engineering Institute in Bilbao, Spain

- **`http://www.mai-net.org/`**
  Network (MAIN) Metrics Associations International

- **`http://www.sttf.fi`**
  FiSMA Finnish Software Metrics Association

- **`http://www.iese.fhg.de`**
  IESE Fraunhofer Einrichtung für Experimentelles Software Engineering
- **`http://www.isbsg.org.au`**
  ISBSG International Software Benchmarking Standards Group, Australia

- **`http://www.nesma.nl`**
  NESMA Netherlands Software Metrics Association

- **http://www.sei.cmu.edu/**
  SEI Software Engineering Institute Pittsburgh
- **http://www.spr.com/**
  SPR Software Productivity Research by Capers Jones
- **http://fdd.gsfc.nasa.gov/seltext.html**
  SEL Software Engineering Laboratory - NASA-Homepage
- **http://www.vrz.net/stev**
  STEV Vereinigung für Software-Qualitätsmanagement Österreichs
- **http://www.sqs.de**
  SQS Gesellschaft für Software-Qualitätssicherung, Germany
- **http://www.ti.kviv.be**
  TI/KVIV Belgish Genootschap voor Software Metrics
- **http://www.uksma.co.uk**
  UKSMA United Kingdom Software Metrics Association

Software Metrics Tools (Overviews and Vendors)

Tool Listings

- **http://www.cs.umd.edu/users/cml/resources/cmetrics/**
  C/C++ Metrics Tools by Christopher Lott
- **http://mdmetric.com/**
  Maryland Metrics Tools
- **http://cutter.com/itgroup/reports/function.html**
  Function Point Tools by Carol Dekkers
- **http://user.cs.tu-berlin.de/~fetcke/measurement/products.html**
  Tool overview by Thomas Fetcke
- **http://zing.ncsl.nist.gov/WebTools/tech.html**
  An Overview about Web Metrics Tools

Tool Vendors

- **http://www.mccabe.com**
  McCabe & Associates

- **http://www.scitools.com**
  Scientific Toolworks Inc.

- **http://zing.ncsl.nist.gov/webmet/**
  Web Metrics

- **http://www.globalintegrity.com/csheets/metself.html**
  Global Integrity

- **http://www.spr.com/**
  Software Productivity Research (SPR)

- **http://jmetric.it.swin.edu.au/products/jmetric/**
  JMetric

- **http://www.imagix.com/products/metrics.html**
  Imagix Power Software

- **http://www.verilogusa.com/home.htm**
  VERILOG (LOGISCOPE)

- **http://www.qsm.com/**
  QSM

METRICS NEWS

VOLUME 10 **2005** **NUMBER 1**

CONTENTS

ISSN 1431-8008