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@acm.org
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*
christofebert@ieee.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 Mathias Lother
The journal is published in one volume per year consisting of two numbers.

CALL FOR PAPERS

Université du Québec
École de technologie supérieure

IWSM2004

14th International Workshop
on Software Measurement

MetriKon2004

DASMA Metrik Kongress

of the **DASMA**- Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung
GI FG 2.1.10 - German Interest Group on Software Metrics and the
CIM - Canadian Interest Group on Metrics
COSMIC – Common Software Measurement International Consortium
MAIN – International Network of Metrics Associations
In cooperation with:
University of Magdeburg, Germany, École de Technologie Supérieure - Université du Québec, Canada, and T-Systems, Germany

November 3-5, 2004, Berlin, Königs Wusterhausen

http://iwsm2004.cs.uni-magdeburg.de, http://www.dasma.org

THEME & SCOPE

Software measurement and metrics application are some of the key technologies to control or to manage the software development process. Measurement is also the foundation of both sciences and engineering, and much more research in software is needed to ensure that software engineering be recognized as a true engineering discipline in order to keep IT companies successful in the marketplace.

TOPICS OF INTEREST

We encourage submissions in any field of software measurement, including, but not limited to

- Software metrics foundations
- Practical measurement application
- Measurement processes and resources
- Empirical case studies
- Measurement acceptance
- Software estimation
- Measurement services
- Functional size measurement
- Software process improvement
- Metrics validation
- Measurement data bases
- Web metrics
- Measurement tool support and infrastructures
- Measurement experience and guidance
- Theory of measurement
- Measurement paradigms
- Enterprise embedded solutions

PROGRAM COMMITTEE

Alain Abran, ÉTS Ecole de Technologie Supérieure, Montreal, Canada
Günter Büren, Büren & Partner, Nuremberg, Germany
Luigi Buglione, Athos Consulting, Italy
Manfred Bundschuh, AXA Cologne, Germany
François Coallier, ÉTS Ecole de Technologie Supérieure, Montreal, Canada
Jean-Marc Desharnais, ÉTS Ecole de Technologie Supérieure, Montreal, Canada
Xavier Dolado, Universidad San Sebastian, Spain
Reiner Dumke, University of Magdeburg, Germany
Christof Ebert, Alcatel, Paris, France
Bernd Gebhard, Bayrische Motorenwerke, Munich, Germany
Hans-Georg Hopf, GSO-Fachhochschule, Nuremberg, Germany
Klaus Lewerentz, TU Cottbus, Germany
Marek Lezak, Lucent Technologies, Nuremberg, Germany
Roberto Meli, DPO, Italy
Dirk Meyerhoff, SQS Software Quality Systems, Cologne, Germany
Andreas Schmietendorf, T-Systems Berlin, Germany
Harry Sneed, SES Munich/Budapest, Hungary
Charles Symons, Software Measurement Service Ltd, Edenbridge, UK
Hannuu Toivoinen, Nokia, Finland
Horst Zuse, TU Technical University Berlin, Germany

SUBMISSIONS

Authors should send proposed *short papers (2 to 4 pages)* by e-mail by June 14st, 2004 to

Alain Abran	***Günter Büren***	***Reiner Dumke***
ETS, Montreal, Kanada	Büren & Partner, Nuremberg	University of Magdeburg, Germany
aabran@ele.etsmtl.ca	gb@bup-nbg.de	dumke@ivs.cs.uni-magdeburg.de

CONFERENCE TIMETABLE

Submission deadline of paper:	June 14, 2004
Notification of acceptance:	August 2, 2004
Final paper deadline:	September 20, 2004
Conference date:	November 3-5, 2004

CONTACT

DASMA e.V.: Romy Gampe (Tel. +49-9126-282441, info@dasma.org);
Uni Magdeburg: Dagmar Dörge (Tel. +49-391-6718664, doerge@ivs.cs.uni-magdeburg.de)

Local Arrangements Chair: Andreas Schmietendorf, andreas.schmietendorf@t-systems.com

FURTHER INFORMATION

For the latest news about the Workshop including the author guidelines and templates see the following Web site:

http://www.dasma.org ***http://iwsm2004.cs.uni-magdeburg.de***

Aufruf zur Einreichung von Beiträgen zum 5. Workshop des Arbeitskreises PEAK der GI FG 2.1.10 am 14. Mai 2004 bei Siemens München

Im zunehmenden Maße gilt es im Rahmen der Softwareentwicklung Beziehungen zwischen Architekturentscheidungen und den daraus resultierenden Performanceeigenschaften des späteren IT-Systems zu erkennen und unter Zuhilfenahme entsprechender Modelle, Methoden und Tools soweit wie möglich zu determinieren. Darüber hinaus besteht der Bedarf den Kostenrahmen einer neuen Lösung im Sinne einer Aufwandsschätzung sowohl aus Sicht der Entwicklung, als auch aus Sicht des potentiellen Betreibers zu identifizieren, auch für diese Aufgabenstellung können Methoden des Performance Engineering herangezogen werden. Der Arbeitskreis Performance Engineering der GI-Fachgruppe 2.1.10 (Software-Messung und -Bewertung) beschäftigt sich mit dieser anspruchsvollen Themenstellung und versteht sich darüber hinaus als Schnittstelle zwischen der Softwaretechnik, der Rechnerarchitektur und der Wirtschaftsinformatik. Der 5. Workshop (PE2004) wird sich unter anderem mit den folgenden Themen beschäftigen:

- *Determinierung und methodische Entwicklung der Performance Eigenschaften im Rahmen serviceorientierter Architekturen bzw. von Web Servcices.*
- *Fallstudien zur Performance-Evaluierung im Rahmen der Softwareentwicklung eingesetzter Technologien, wie z.B. Middleware, Datenbankmanagementsystemen.*
- *Identifizierung des "Business Sucess Factor" für die Anwendung der Methoden des SPE in der industriellen Softwareentwicklung.*
- *Erfahrungen mit Notationen, Methoden und Tools zur Modellierung (z.B. UML 2.0) des Performanceverhaltens von Soft- und Hardwaresystemen.*
- *Aufgabenstellungen des Performance Engineering bei der Entwicklung von komponenten- und agentenbasierten Systemen.*
- *Integration der Themenstellungen des Software Performance Engineering in die Curricula der Universitäten und Hochschulen.*

WORKSHOP-BEITRÄGE

Praktiker und Wissenschaftler, die auf dem Gebiet des Performance Engineering und artverwandter Aufgabenbereiche aktiv sind, werden gebeten, Beiträge in einem der Formate doc, rtf, pdf, ps einzureichen. Der Umfang der Beiträge sollte 3000 Wörter nicht übersteigen. Die Formatierungsrichtlinien der Webseite des Arbeitskreises zu entnehmen. Die Beiträge sollen in Kurzvorträgen (Vortragslänge ca. 30 Minuten zzgl. 10 Minuten Diskussion) präsentiert werden. Angenommene Beiträge werden in einem Tagungsband veröffentlicht.

Bitte senden Sie ihre Beiträge per E-mail an: andreas.schmietendorf@t-systems.comIn Ausnahmefällen auch auf dem Postweg an das Tagungssekretariat:

Dagmar Dörge, Universität Magdeburg, FIN-IVS, Universitätsplatz 2, 39106 Magdeburg

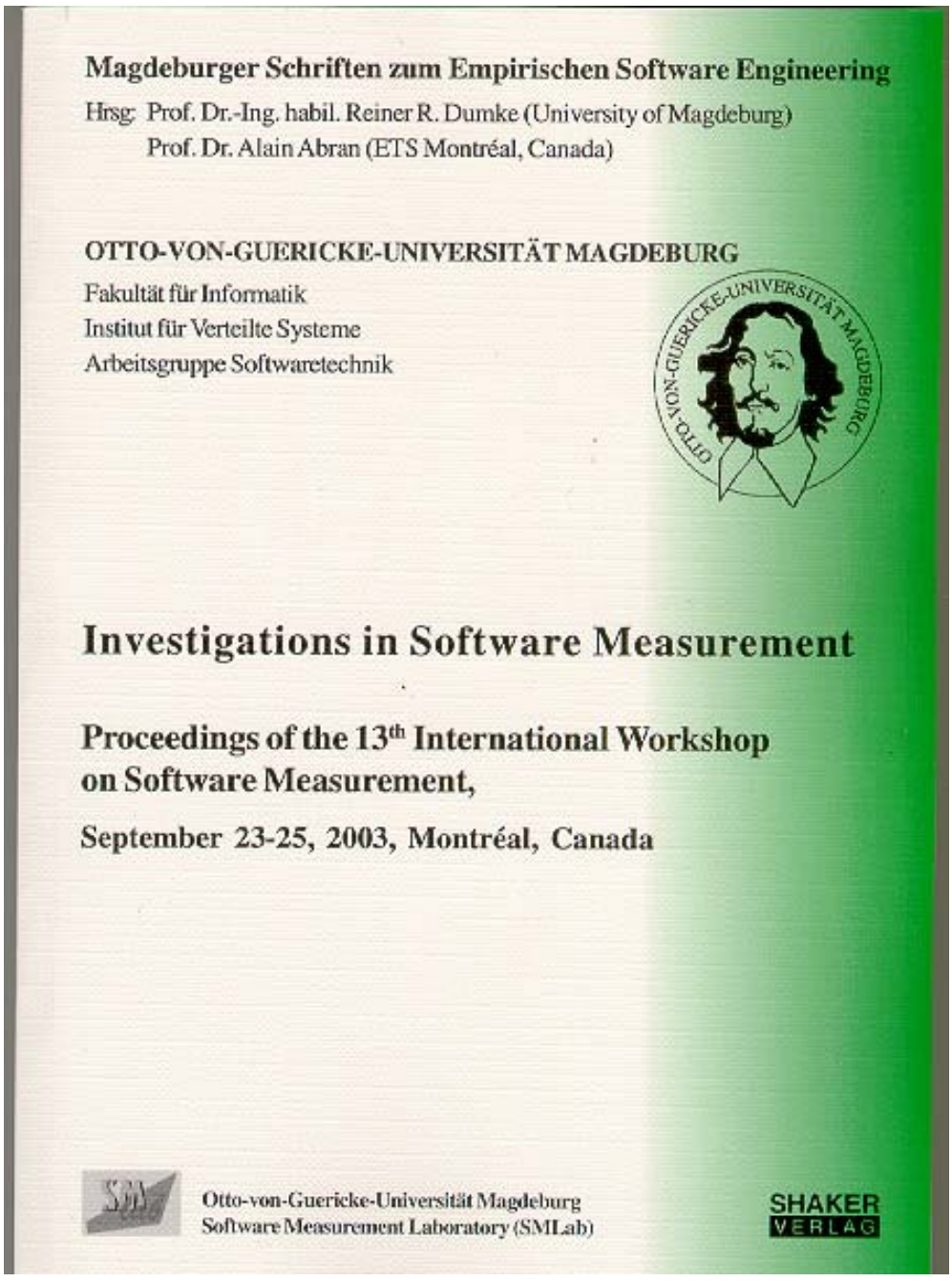
TEILNAHME / ANMELDUNG

Interessenten werden gebeten, sich auf der Webseite des Workshops elektronisch anzumelden. Bei Anmeldung bis zum 1. Mai 2004, wird eine Teilnahmegebühr von 50.- Euro (GI-Mitglieder: 40.- Euro) erhoben, danach 60.- Euro bzw. 50.- Euro.

PROGRAMMKOMITEE	**TERMINE**
Prof. R. Dumke, Universität Magdeburg Dr. R. Gerlich, BSSE System and Software Engineering Prof. G. Henselmann, Fachhochschule Schmalkalden H. Herting, T-Systems, Benchmarklabor Darmstadt Prof. R. Hopfer, HS für Technik und Wirtschaft Dresden Prof. K. Kabitzsch, Technische Universität Dresden Prof. F. Lehmann, Universität der Bundeswehr München Prof. C. Rautenstrauch, Universität Magdeburg Dr. A. Schmietendorf, T-Systems Nova, EZ Berlin Dr. A. Scholz, Booz Allen Hamilton Dr. D. Stoll, Lucent Technologies Prof. F. Victor, Fachhochschule Köln	22.03.2004: Einreichung von Beiträgen 05.04.2004: Annahme/Ablehnung 11.04.2004: finales Workshop-Programm 20.04.2004: druckreife Beiträge 14.05.2004: Workshop in München

Workshop Report

Our 13th Workshop on Software Measurement **IWSM2003** took place in Montreal, Canada in September 2003. The following report gives an overview about the presented papers. Furthermore, the papers are published in the following Shaker book:



ISBN 3-8322-1880-7

[3] ***Workshop Report***

Software Maintenance Capability Maturity Model (SM-CMM): Process Performance Measurement

Alain April [1], Alain Abran [2], Reiner R. Dumke [3]
[1] Bahrain telecommunications Company, Bahrain, april_alain@hotmail.com
[2] École de Technologie Supérieure, Montréal, Canada , aabran@ele.etsmtl.ca
[3] Otto von Guericke University of Magdeburg, Germany, dumke@ivs.cs.uni-magdeburg.de

Abstract: *Software maintenance constitutes an important part of the total cost of the lifecycle of software. Some even argue that this might be the most important component of the cost, even though customers often do not perceive the added value of software maintenance. A proposed approach to highlighting the added value of maintenance is to provide the customer with process performance measures aligned with the key activities performed by the maintenance organization. Such performance measures could then form the basis for a clear agreement on the expectations, and outcomes, of these activities.*

Process Performance management and measurement requires that processes be chosen based on their impact on the quality and the performance of the software maintenance organization It also requires that measures be identified and established and that a reference point (baseline) and a target be set for each measure. Finally, they require that data be collected in order to develop and use process performance prediction models. In this paper, we introduce best practices, for the first three maturity levels, to help the maintainer organization assess its process performance. These practices constitute a subset of our proposed Software Maintenance Capability Maturity Model (SM-CMM).

Toward an Ontological Formalization for a Software Functional Size Measurement Method's Application Process[1]: The COSMIC-FFP Case

Valéry Bévo, Ghislain Lévesque, Jean-Guy Meunier
bevo_wandji.evariste@courrier.uqam.ca,
Levesque.Ghislain@uqam.ca, Meunier.Jean-Guy@uqam.ca
L.R.G.L. – L.A.N.C.I, UQAM

Abstract: *Despite significant improvements in recent years (efforts at standardization, for example), software functional size measurement (FSM) is still not an obvious activity, especially in the application of measurement methods. Appropriate expertise for each method is needed, and those implementing existing software FSM systems in the industry face two major technical difficulties: a tiresome procedure for applying them which sometimes requires the help of experts, who may not always be available [Carpers 96], and a lack of support tools to help measurers in their task. With respect to this last point, Sue Black & David Wigg [Black et al. 99] have noted: "The software industry needs software measurement tools which can be used to compute measures across several platforms and languages to provide flexibility and usability." The design of such tools is necessarily based on: (1) identification of all the concepts handled in a method's measurement procedure, as well as the relationships between these concepts (domain ontology); (2) identification of all the tasks*

[1] Throughout this paper, the expression "a method's measurement procedure" will be used for "a software FSM method's application process", and vice-versa.

associated with a method's measurement procedure, as well as the links between these tasks (task ontology). This paper explores an ontological perception of a method's measurement procedure (a FSM method's application process). The foundations for an ontological formalization of the procedure are provided: An emphasis is put on the domain and task ontologies associated with the procedure. In addition to the fact that they can be used for the design of measurement tools, such ontologies could be helpful in providing a better understanding of the measurement procedures of related methods, and serve as consensual rallying points to structure, represent, exchange and interpret information, "things" ("concepts") related to the measurement procedure and measurement in general. It can be seen, in turn, as constituting the foundation for a body of knowledge associated with each method's measurement procedure. The formalism used for presenting these ontologies should be appropriate. The object-oriented formalism is used here. The expressiveness, simplicity and popularity of this formalism are the main factors on which the choice was based. Tasks and concepts are described according to the CommonKADS methodology [Schreiber et al. 99].

Web-based Support for White Box Software Estimation

Alain Abran[1], René Braungarten[2], Reiner R. Dumke[2]
[1] Département de génie électrique et génie logiciel,
École de technologie supérieure Montréal (Québec), Canada
[2] Otto-von-Guericke University Magdeburg, Germany

Abstract: *Most commercial estimation tools can be considered as black boxes in that they do not provide details of the samples used to build their estimates. With the availability of the ISBSG international repository of 2000+ software projects, it is now feasible to develop white box estimation models which provide additional insights into the strengths and limitations of software estimates. This paper presents two Web-based software prototypes developed to support white box software project estimation.*

Assessment of Measurement Indicators in Software Process Improvement Frameworks

Luigi Buglione [1 2], Alain Abran [1]
[1] École de Technologie Supérieure – ETS, 1100 Notre-Dame Ouest, Montréal,
Canada H3C 1K3
[2] SchlumbergerSema Via Riccardo Morandi 32,I-00050 Rome, Italy
lbuglione@rome.sema.slb.com, aabran@ele.etsmtl.ca

Abstract: *Measurement is progressively becoming a mainstream management tool to help ICT organizations plan, monitor and control. However, measurement itself is not a mature domain of knowledge in software engineering. The assessment of proposed measurement indicators in these process improvement models is investigated, and a methodology is proposed for the design of a measurement indicator assessment grid. A case study on the use of this assessment grid is presented and results discussed.*

A Quality Model for Web-based Environments: GUFPI-ISMA viewpoint

[10] ***Workshop Report***

Luigi Buglione[2]*, Francesco Gasparro, Enrico Giacobbe, Claudio Grande, Salvatore Iovieno, Angelo Scarcia, Habib Sedehi, Gianluigi Raiss*
GUFPI-ISMA SMC (Software Measurement Committee)
Gruppo Utenti Function Point Italia – Italian Software Metrics Association
URL: http://www.gufpi.org
Via Mosca 32, I-00100 Rome, Italy

Abstract: *The rapid and penetrating development and diffusion of the Internet technology during the '90s has placed the World Wide Web (WWW) in a strategic position for the Information & Communication Technology (ICT) business, converting a simple and static informative content provided from the original html pages onto a new distributed application platform. The evidence for this figure is the pervasive diffusion of the e-commerce, and the turnover it has generated from the several offered services: banking, finance, until the retail sales in the private sector, as well as the distribution of services in the public sector.*

Thus, it becomes more and more fundamental to evaluate the effectiveness and efficiency of web-related products and services, and several perspectives can be taken into account, mostly those from the two main stakeholders:

- *Internal view (the software producer): a web-agency will gather qualitative and quantitative measures and metrics in order to take under control its production process and allows the calculation for estimating the effort needed for the next project. The interest object is therefore the "product".*
- *External view (the customer): an end user will observe the web from a functional viewpoint and it will be relevant to gather his/her impressions about the effectiveness and efficiency of what experienced, in order to be retained, withheld from the software producer. The interest object for measurement is therefore the "service".*

Paying attention to the first perspective, two main ways to approach the measurement for web environment has been proposed: a functional and a not-functional ones. Traditional Functional Size Measurement approaches such as Function Points Analysis (FPA) seem to do not have an optimal fit with web environments that deal with links, multimedia, and so on. Thus, a new Technical Committee within the Italian Function Point User Group - Software Metrics Association (GUFPI-ISMA) has been created and is working on it, focussing on the non-functional side of web measurement. Software Measurement Committee (SMC) roadmap will be presented, as well as its main outcomes: metrics set, measurement plan, metric tools and suggestions for SPI and the way they can be applied for case studies, deserving attention to a design of experiment and first indications from trials.

From IT-Centric to Business-Centric Productivity Measurement

Radenko Corovic
SAAQ - Quebec Government, radenko.corovic@saaq.gouv.qc.ca

Abstract: *Information technology was long time considered as a creative activity to which the economic rules cannot be applied. The principal issue was technology and the only measure of success was the efficiency of technological solutions. Is this solution useful for the organization from the business point of view, is it really the best solution and, particularly, how will this solution affect the overall organizational productivity, were questions in which the IT department was not much interested.*

But now, the companies are more and more concerned by the impact of IT investments on overall productivity. They want to know what the investments in information technology will bring as a

[2] Corresponding Author. Email : luigi.buglione@gufpi.org, Tel : +39-06-8307.4472, Fax :+39-06-8307.4200

business value. It goes beyond the classic comprehension of ROI as the criteria for IT investments and it's obvious that the classical financial measures are not sufficient to respond adequately to this question.

This paper is written to put more emphasis on the measure of productivity in IT and especially, how to measure the IT contribution in organizational productivity, the most difficult thing for companies.

(Extended) Functional size measurement methods are also applicable in enhancement projects

Ton Dekkers
Sogeti Nederland B.V.
Goudsbloemvallei 50, 5237 MK Den Bosch, Netherlands
ton.dekkers@sogeti.nl

Abstract: *Looking at the distribution of the costs of IT, the largest part of the budget is allocated to maintenance and enhancement projects. New development comprises between 30 to 50% of IT costs. Functional size measurement methods are mostly used for new development only. With some extensions to common size measurement methods like Function Point Analysis and COSMIC Full Function Point one can tackle almost all IT activities. Furthermore the same productivity rates (performance) can be used in enhancement projects also. Over the last 10 years we used the extended measurement method based on Function Point Analysis very successfully in a great number of projects.*

Approximation techniques for measuring Function Points

Jean-Marc Desharnais, Alain Abran
École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest, Montréal, Canada H3C 1K3
jmdeshar@ele.etsmtl.ca, aabran@ele.etsmtl.ca

Abstract: *The generic concepts of Function Points Analysis were published in the late 1970s, and later more detailed measurement rules were developed to improve consistency of measurement. Due to lack of good software documentation, it is not always possible to apply all the detailed rules, and measurers must fall back on approximation techniques. This paper presents an analysis of two such techniques: Function Points Simplified and "backfiring" with a ratio of lines of code per Function Point. Two verification criteria were selected from ISO 14143-3: accuracy and convertibility. Results from empirical studies with five data sets are reported.*

An approach for the further development of the measuring system for the software quality assurance of Bosch Diesel Systems

Ingo Hofmann[1, 2], Michael Zimmermann[1], Reiner R. Dumke[2]
1Robert Bosch GmbH, Germany
Ingo.Hofmann2 | Michael.Zimmermann@de.bosch.com
[2]Otto-von-Guericke-University Magdeburg, Germany
dumke@ivs.cs.uni-magdeburg.de

[2] ***Workshop Report***

Abstract: *In software development – even in quality assurance – many decisions are still made instinctively. Lack of transparency, however, makes these decisions difficult to communicate and error-prone. A better solution would be to make decisions that are logical and comprehensible, which are based on facts. These decisions are not only simpler to communicate, but also reduce the frequency of wrong decisions. This, however, requires significant data about the software quality and its influencing factors, which, in turn, requires a measurement-based software quality assurance.*

In the past, many attempts were undertaken to initiate adequate measuring systems to collect significant data. Many of these attempts failed due to non-acceptance by the users, owing to the obvious disproportion of effort and benefit.

The present article outlines the basic approaches at Bosch Diesel Systems (DS) to adapt the available measuring system to the rising demands on software quality assurance. An important aspect to be particularly considered is that of acceptance. With specially considering the aspects of dynamics and learning, the special customer-oriented software development is born in mind. Furthermore, the measuring system must provide proof of its ability and applicability before its final implementation. For this purpose, experiments are performed in a pilot environment.

The aim is to develop and test an approach for the improvement of the existing measuring system, which is accepted by the users and will help in the software quality assurance to move to objective decisions.

A prototype Web-based implementation of the QEST model

Alain Abran[1], Martin Kunz[2], Reiner R. Dumke[2], Luigi Buglione[1 3]
[1] École de Technologie Supérieure – ETS, 1100 Notre-Dame Ouest,
Montréal, Canada H3C 1K3, aabran@ele.etsmtl.ca

[2] Otto-von-Guericke University Magdeburg – GERMANY
makunz@cs.uni-magdeburg.de, dumke@ivs.cs.uni-magdeburg.de
[3] SchlumbergerSema, Via R.Morandi 32, I-00050 Rome - ITALY
lbuglione@slb.com

Abstract: *This paper presents and describes a Web-based implementation of a three-dimensional software quality measurement model. The implementation is based on the 2003 version of the ISO quality model for software products: ISO 9126. The prototype presented includes all the 120+ measures proposed in the ISO standard, as well as weight assignments, target values, current project values and automated calculations for a three dimensional representation of quality performance, based on the geometrical tetrahedron formula of the QEST model.*

Integrated Validation Process of Software Measure

Miguel Lopez[1], Valérie Paulus[1], Naji Habra[2]
[1]CETIC asbl, Rue Clement Ader, 8, 6041 Gosselies Belgium
{malm,vp}@cetic.be, http://www.cetic.be
[2] University of Namur, Rue Grandgagnage, 21, 5000 Namur Belgium
nha@info.fundp.ac.be
http://www.info.fundp.ac.be/~software-quality

Abstract: *The validity of the measures used in software engineering is a critical matter about which no consensus has yet emerged, although it has prompted hard discussion. There is a need for*

unambiguous definitions of the mathematical properties that characterize the major measurement concepts. Such a mathematical framework could help to generate consensus among the software engineering community.

The goal of this paper is to provide a formal validation process for software measurement. It presents a global measurement framework that integrates theoretical and empirical validation processes based on measurement theory. The concept underlying the framework is to formalize some properties of the measure to be analyzed, and then to verify the conformity of these properties to the measure by means of formal experimentation.

This validation process determines a contextual validity (scope) defined by the set of factors or validity conditions that impact the validity of the measure. The paper develops two case studies which, under specified conditions, invalidates the Coupling Between Objects (CBO) a measure of coupling and validates, under other conditions, the response time as a measure of the time behaviour (sub-characteristic of the efficiency).

Applicability of COSMIC Full Function Points for BOSCH specifications

Mathias Lother[1], Reiner R. Dumke[1], Thomas Böhm[1], Horst Herweg[2], Willy Reiss[2]
[1] University of Magdeburg,
{lother,dumke}@ivs.cs.uni-magdeburg.de
http://ivs.cs.uni-magdeburg.de/sw-eng/us/
[2]Robert Bosch GmbH, Postfach 300240, 70442 Stuttgart

Abstract: *In the area of software development software measurement gains increasing importance in order to manage and to improve the underlying processes and models as well as the software quality. Especially in the automotive industry with their embedded control software there is the need for high-quality software development processes to ensure the delivery of the software in time and budget as well as to meet predefined resources requirements.*

In order to manage and control the software development the management needs metrics for different aspects. A lot of these desired metrics and methods, e.g. effort estimation, resources consumption (software size / memory unit), market value (price / size), software quality (defects / size) and productivity (size / time), depend on the size of the software. For several important reasons the metric Lines of Code (LOC) does not fulfill the needs for an adequate size measure at Bosch/Germany. Therefore, the measurement of the software size in another, a functional manner is currently an important challenge.

This paper presents a feasibility study for the applicability of COSMIC Full Function Points (FFP) for Bosch-specific function definitions. Besides the presentation of a characteristic FFP measurement procedure model, that has been developed, the particularities of the FFP adaptations to these special specification documents will be shown. Furthermore, experiences (e.g. time effort) will be shared, solutions for open and environment-specific problems (e.g. development of variants, algorithmic complexity) are suggested and different possible application areas for the derived Functional Size Measurement (FSM) values will be discussed.

Measuring Consistency of the Analysis Model: An XML Approach

K. Meridji, O.Ormandjieva
Computer Science Department
Concordia University

Montreal, Quebec, Canada
{k_meridj,ormandj}@cs.concordia.ca

Abstract: *Software engineering is a discipline whose aim is the production of quality software, delivered on time, within budget, and satisfying user expectations. The control on the software development would allow increasing the quality of the final product. To keep the development of the product within the given time and budget resources, the quality has to be controlled starting right from the initial phase of the development process, software requirements analysis and specification. The analysis model describes the static and dynamic views on the boundary of a system under development, as seen from outside the system. With UML, the connection between the views is not obvious; therefore the consistency between static and dynamic views needs to be checked formally. In this paper, we propose a formal representation of the analysis model in XML, and the definition, validation and measurement of the consistency between the static and dynamic views based on their formal representation.*

Functional metrics using COSMIC-FFP for object-oriented real-time systems

Shin'ichi Nagano, Tsuneo Ajisaka
Department of Design and Information Science, Wakayama University
nagano@tta.or.jp, ajisaka@sys.wakayama-u.ac.jp

Abstract: *We propose a formal procedure of measuring the amount of functions required in a real-time system under object-oriented analysis.*

In the scheme of our measuring procedure, the Shlaer-Mellor method is used as an object-oriented method and the products should be written in the Executable UML. The measure is COSMIC-FFP, which is a real-time enhancement of the IFPUG method. We propose a procedure in which required functionality is automatically measured for the products denoted in the Executable UML, without concerning the measurement expertise.

This paper shows the detail of the proposed metrics, and its suitability and effectiveness through a case study.

Rewriter's note: Very little of your original document contains meaningful English sentences. The entire rewrite is a guess. As such, there will be some errors in the ideas that I express, and not all of the paragraphs will be coherent. You should identify the correct sentences of the rewrite and revise the others. Then send the completed revision of the manuscript for a second check.

QUIM: a Tool and a Knowledge Map for Usability Measurement

Harkirat Kaur Padda, Ahmed Seffah and Jovan Strika
Human-Centered Software Engineering Group
Department of Computer Science, Concordia University
1455, Maisonneuve W., Montréal, Québec, Canada, H3G 1M8
seffah@cs.concordia.ca, http://hci.cs.concordia.ca

Abstract: *In this paper, we describe QUIM - Quality in Use Integrated Measurement – an environment that aims to bridge the gap between expert-based usability evaluation practices and software engineering quality models for quantifying and measuring usability or quality in use – the user perspective of software quality. The QUIM knowledge map is based on the decomposition of usability factors into measurable criteria and metrics. QUIM includes 10 factors namely efficiency, effectiveness, satisfaction, productivity, learnability, safety, trustfulness, accessibility, universality and*

usefulness. These factors are mapped to a total of 27 criteria that are measured through more than 125 usability metrics we collected from different human computer interaction models such MUSiC and GOMS as well as software engineering standards such as ISO 9126 and 9241. Empirical rules for understanding and interpreting metrics and criteria are also part of QUIM. As a software tool, QUIM helps developers and managers to visually browse and customize the QUIM knowledge map for his/her needs. QUIM can used to build specific models such as measuring the safety issue for interactive critical systems, Web accessibility or user performance or satisfaction with mobile services. QUIM editor is useful for anyone who wants to learn how usability can be measured, how to specify software usability, as well as how usability measurement should be integrated in the software development lifecycle and in particular in software testing.

Desing of a Generic Performance Measurement Repository in Industry

Alain Abran, Edgardo Palza
École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest, H3C 1K3 Montréal Québec , Canada,
aabran@ele.etsmtl.ca, edgardo.palza-vargas.1@ens.etsmtl.ca

Abstract: *Understanding, predicting, and controlling performance is a continuous challenge, and static measurement systems are inadequate in dynamic and rapidly changing business environments. In this paper, we propose a generic, flexible and integrated Measurement System Repository to handle continuously changing business conditions, and we report our experience in its design and development at Ericsson Research Canada.*

This Performance Measurement Repository has been developed based on the concept of a data warehouse environment. Reporting features are based on the definition of queries to On Line Analytical Process (OLAP) cubes. OLAP cubes are created as materialized views of the measurement data, and the user functionalities are implemented as analytical drill-down/roll-up capabilities and as Indicator and Trend Analysis capabilities.

Metric based comparison of project lines within the industrial software development

Daniel Reitz#, Reiner R. Dumke*, Andreas Schmietendorf#**
T-Systems International, Entwicklungszentrum Berlin, Wittestraße 30G, D-13476 Berlin
{daniel.reitz, andreas.schmietendorf}@t-systems.com
* Otto-von-Guericke-University Magdeburg, Fakultät Informatik, Institut für verteilte Systeme, Postfach 41 20, D-39016 Magdeburg
{reitz, schmiete, dumke}@ivs.cs.uni-magdeburg.de

Abstract: *In this article we would like to present and discuss an approach to benchmark and compare project lines in the field of industrial software development. The aim of this benchmark is to focus on functional and qualitative aspects too. Furthermore as well as on the post calculation and estimation of consumed efforts. For that purpose we discuss existing methods and metrics, present our*

own developed measurement procedure based on the CMM-model and the COCOMO-method and show selected aspects of the application of our benchmark. Finally the article provides possible conclusions and potential problems.

Empirical Analysis of available Web Services

Andreas Schmietendorf[#], Reiner R. Dumke[*]*
T-Systems International, Entwicklungszentrum Berlin, Wittestraße 30G, D-13476 Berlin, E-Mail: andreas.schmietendorf@t-systems.com
* Otto-von-Guericke-Universität Magdeburg, Fakultät Informatik, Institut für verteilte Systeme, Postfach 41 20, D-39016 Magdeburg, E-Mail: {schmiete ,dumke}@ivs.cs.uni-magdeburg.de

Abstract: *This paper gives a first insight in an ongoing investigation. Our generic aim is to describe the current situation of available Web Services by the use of metrics. The current maturity of offered Web Services should be evaluated in particular. The first section provides an overview about the Web Service technology. The following section describes the aims of the investigation and the measurement aspects in detail. The third section shows selected results of the empirical analysis and gives a first interpretation of the measurements. Finally the reader will find conclusions and planned further investigations. This research is realized by the Software Measurement Laboratory of the University of Magdeburg and the System- and Technology-Development Group of the Development Centre Berlin.*

The contribution of metrology concepts to understanding and clarifying a proposed framework for software measurement validation

Asma Sellami, Alain Abran
École de Technologie Supérieure – ETS, 1100 Notre-Dame Ouest, H3C 1K3 Montréal Québec, Canada,
asma.sellami.a@ens.etsmtl.ca aabran@ele.etsmtl.ca

Abstract: *In the software engineering literature, numerous practitioners and researchers have proposed hundreds of "software measures", or "software metrics". To help industry assess the quality of these proposed measures, various researchers have proposed various approaches to software measurement validation, none of which has yet been widely used by either designers or users of software measures. To tackle this diversity of validation approaches, Kitchenham et al. had proposed a framework for software measurement validation and suggested a critical review of their proposed framework. This paper performs such a review using a key ISO document on measurement, that is the ISO Vocabulary on Metrology as well as a measurement process model derived from an analysis of the individual validation proposals. The metrology concepts in particular have facilitated greater understanding of the set of measurement sub-concepts that must be included in each of the steps from the design of a measurement method to the use of the measurement results.*

Applicability of COSMIC Full Function Points in an administrative environment: Experiences of an early adopter

Frank Vogelezang, Arlan Lesterhuis
Sogeti Netherlands B.V.

frank.vogelezang@sogeti.nl, arlan.lesterhuis@sogeti.nl

Abstract: *Rabobank is reshaping its systems portfolio of Rabobank from dedicated product systems to a network of generic services with a shared data source. In this environment Function Point Analysis no longer fits the sizing needs. An alternative was found in the COSMIC Full Function Points method. Because of the absence of benchmark data a conversion formula was derived for projects that were measurable in both COSMIC Full Function Points (End User Viewpoint) and Function Point Analysis. This conversion formula now reads as:*

$$Y\ (cfsu) = -87 + 1{,}2\ X\ (fp)$$

The correlation coefficient for this conversion formula is 0,99 and the standard deviation in the difference in the Y-value is 59.

To support the estimating process in early stages of systems development the locally calibrated approximate version of COSMIC Full Function Points was derived from the first set of measurements. Our version shows very good resemblance to the version presented in the Measurement Manual. Because these versions were derived in a very different environment this might be an indication that these figures have a more general applicability.

The quality concepts and subconcepts in SWEBOK: An ontology challenge

Cornelius Wille+, Alain Abran, Jean Marc Desharnais, Reiner R. Dumke+
ETS, 1100 Notre-Dame Ouest, H3C 1K3 Montréal Québec , Canada,
(aabran,jmdeshar)@ele.etsmtl.ca
+Otto-von-Guericke-University of Magdeburg,
(dumke,wille)@ivs.cs.uni-magdeburg.de

Abstract: *The Guide to the Software Engineering Body of Knowledge (SWEBOK) has been developed to represent an international consensus formed through broad public participation in the review process and is now close to final approval as ISO/IEC TR 19759. This guide constitutes an integrated structuring of a large set of software engineering concepts developed individually over the past forty years from a large number of distinct viewpoints. The absence of a recognized consensus on software engineering terminology has been a challenging task in building the SWEBOK Guide, and in achieving an international consensus. While major consensus has been reached at the broad taxonomy level of SWEBOK, some work remains to increase terminology consistency at a more detailed level. This paper briefly presents SWEBOK and related terminology issues. We then present the ontology approach to building domain-specific ontologies and show how it can be used to build the SWEBOK ontology and to increase its internal consistency and clarity. A specific example of the benefits of an ontology is presented, along with an analysis of the use of the term 'quality' in the current version of the SWEBOK Guide.*

What can Practioneers learn from Measurement Theory?

Horst Zuse
Technische Universität Berlin, Franklinstraße 28/29, D – 10719 Berlin
Phone: +49-30-314 24788, Mobile: +49 172 304 2063
Internet: http://www.zuse.info, E-Mail: horst.zuse@t-online.de

Workshop Report

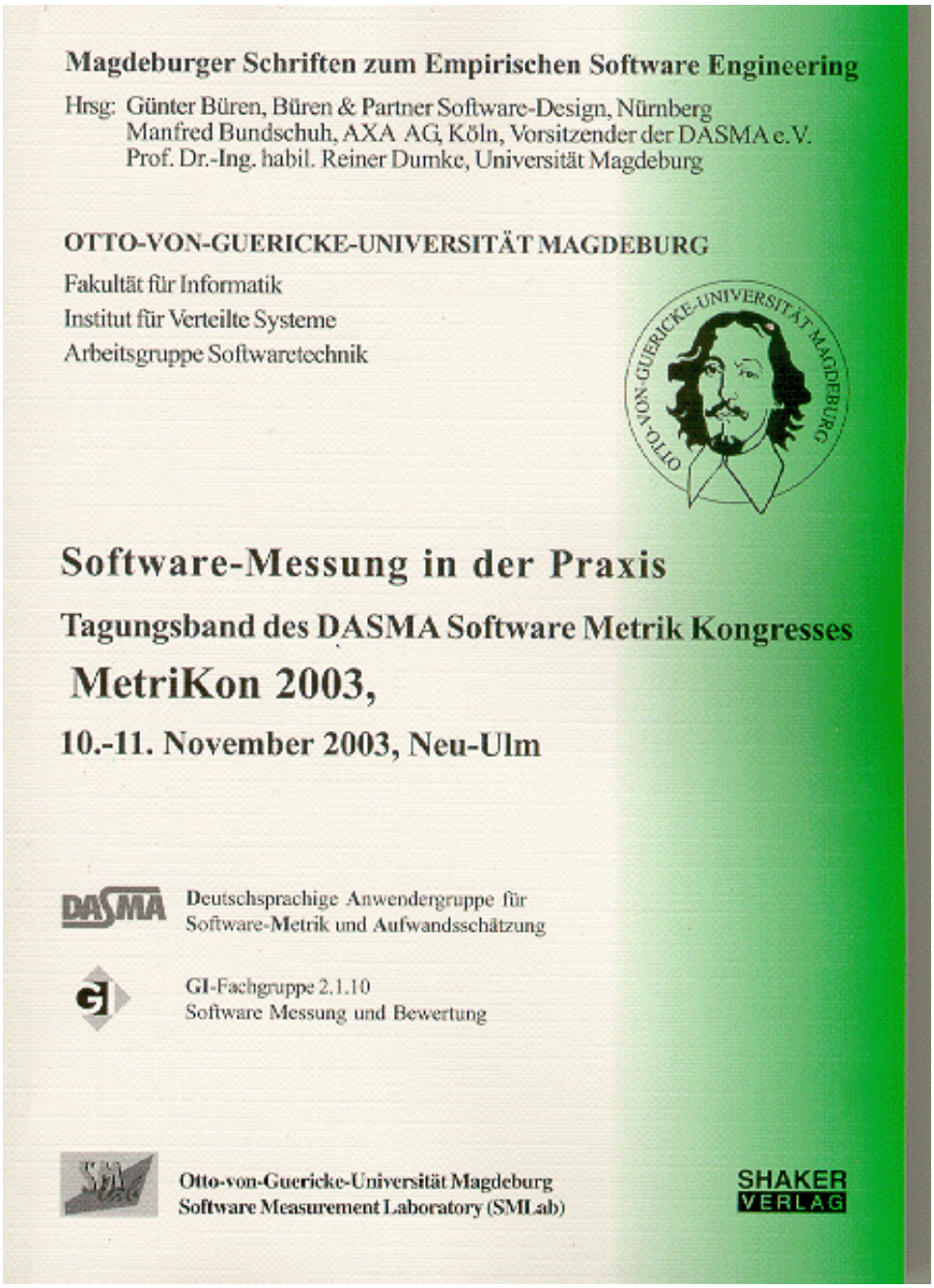
Abstract: *Today it is widely accepted that software measurement is a valuable technique for understanding, guiding, controlling and improving software development. The major problem of measurement in software engineering, but also in the area of artificial intelligence, is a skepticism of using numerical values because there is no satisfaction in the interpretation the numbers and a semantic of the values is missing.*

There exists a well established theory – called measurement theory – which is used with a great success in many scientific areas. In the area of software measurement the influence of measurement theory for the work of practitioneers is still neglected. However, we are convinced, that measurement theory can support the work of practioneers in a wide range. We present ten theses:

1. *Measurement theory gives clear definitions of measures and connects quality (empirical) properties with numerical ones by a homomorphism.*
2. *Measurement theory gives criteria for scales and proper statistical operations. For example: Size measures should assume certain measurement structures, and, be careful with the Halstead measures when you want to use them as size measures.*
3. *The concistency property (among others important for distributed software development) is one of the most important requirements for software measurement and assumes certain measurement structures.*
4. *Additive and strong monotonic transformations measures are useful for imperative languages.*
5. *For object-oriented measures additive measures are not useful (only for the methods). We need for that modified measurement structures (function of belief). Considering software measures, there is a fundamental difference between the object-oriented and imperative paradigms.*
6. *The requirement of wholeness for product measures is a pseudo requirement. However, for prediction it is useful.*
7. *To ignore measurement scales is careless. Scales are everywhere if you want it or not!*
8. *Not every measure can be validated.*
9. *Prediction functions cannot be defined arbitrarily. Very strong conditions have to be valid.*
10. *Units and dimensions of measures are based on scales.*

In the presented paper all the theses will be explained in detail with examples and counterexamples.

The DASMA Metrik Kongress **MetriKon 2004** took place in Ulm, Germany in November 2003. The following report gives an overview about the presented papers. Furthermore, the papers are published in the following Shaker book:



ISBN 3-8322-2146-8

Six Sigma für Software

Thomas Fehlmann
Euro Project Office AG, Zeltweg 50, CH-8032 Zürich
thomas.fehlmann@e-p-o.com

Abstract: *Six Sigma ist ein in der Industrie bekanntes und erfolgreiches Managementkon-zept. Durch gezielte Reduktion von Fehlern und Ausschuss werden Kosten dramatisch reduziert. Das Konzept funktioniert ausschliesslich über Metriken, welche an den Bedürfnissen der Kunden orientiert sind. In der Vergangenheit gab es viele Versuche, Six Sigma für Software nutzbar zu machen. Einige davon gelangen, aber alle mussten das Problem lösen, welche Metriken letztlich für die in der Entwicklung Beteiligten brauchbar sind, um den tatsächlichen „Ausschuss" zu messen. Ein erfolgreicher Ansatz geht über das Projektmanagement. Gemessen wird der Konsens zwischen den am Software-Projekt Beteiligten. Die Messverfahren werden durch den Qualitätsplan festgelegt. Anhand der in vielen Projekten gewonnenen Erfahrungen wird gezeigt, wie das Verfahren funktioniert. Dieser Ansatz gilt nicht nur für Softwareentwicklung, sondern für jede Art von Projekten mit hohem Anteil an Dienstleistungen.*
Ein zweiter Ansatz geht über Quality Function Deployment (QFD). Mit Hilfe von QFD werden Messnetze erstellt, die es ermöglichen, Messungen der Kundenzufriedenheit, des Markterfolgs, der Software – Fehlerdichte sowie Assessments des Reifegrades gemäss einem geeigneten Capability Maturity Model (CMM) mit Hilfe von Kombinatorischen Metriken zu verknüpfen. Ein solches Messnetz ermöglicht es, allen Stellen in der Softwareentwicklung geeignete, auf den Projekterfolg geeichte Metriken zur Verfügung zu stellen. Mit Hilfe solcher geeichten Metriken lassen sich in der Softwareentwicklung kurze und einfache Rückkoppelungszyklen über Soll/Ist – Vergleiche und damit letztlich bedeutende Produktivitätsverbesserungen realisieren. Das Verfahren wird am Beispiel des Metrikprogramms eines Softwarehauses vorgestellt. Die beiden Ansätze lassen sich ohne weiteres kombinieren. Doch gibt es einen dritten Ansatz, der bereits für sich allein ausserordentlich erfolgreich ist: Proposal Management. Darunter versteht man dedizierte Prozesse und spezialisierte Organisationseinheiten („Proposal Centers"), die auf die Initialisierung von Projekten fokussiert sind. Es wird gezeigt, dass Proposal Management auch dann funktioniert, wenn vorwiegend interne Projekte durchgeführt werden.

Zwei Anwendungen von GQM: Ähnlich, aber doch nicht gleich

Thomas Gantner [1] und Kurt Schneider [2]
[1] DaimlerChrysler Forschungszentrum Ulm, Postfach 2360, 89013 Ulm
[2] jetzt: FG Software Engineering, Universität Hannover, Welfengarten 1, Hannover
Thomas.Gantner@DaimlerChrysler.com, Kurt.Schneider@Inf.Uni-Hannover.de

Abstract: *Bei DaimlerChrysler tritt häufig der Wunsch auf, durch messbasierte Begleitung gewisse Effekte zu objektivieren oder Aussagen zu stützen. In Anbetracht dieser Randbedingungen setzen wir GQM (Goal-Question-Metric) als Methode ein, um zu maßgeschneiderten Messprogrammen zu kommen. Zuletzt wurde GQM in zwei Bereichen eingesetzt, deren Absichten hinter der Messung sich scheinbar ähnelten. Im ersten Fall waren jedoch die Fragestellungen viel konkreter als im zweiten. Es stellte sich im Vergleich der beiden Anwendungsfälle die Frage, ob GQM mit beiden Fragestellungen trotz der erheblichen Unterschiede zurecht kommen würde bzw. wie große Anpassungen erforderlich werden würden. GQM ist eine im Prinzip leicht verständliche Methode, und bei wiederholter Anwendung liegt es nahe, durch Wiederverwendung oder gar Automation von GQM-Schritten den Aufwand zu reduzieren. Es zeigte sich jedoch rasch, dass diese Versuche zur Arbeitserleichterung in einem Spannungsverhältnis zu der gleichzeitig sehr willkommenen, allgemeinen Anwendbarkeit der Methode stehen.*
Wir stellen die beiden Anwendungsfälle vor, zeigen an interessanten Schritten der GQM-Methode, wie sie in den beiden Fällen angewandt wurde. Daraus leiten wir unsere Empfehlung ab, den Aufwand von GQM eher durch systematisches Weiterreichen von zuvor durchgeführten GQM-Unterlagen und –Erfahrungen zu reduzieren als durch Automatisierungswerkzeuge.

Software Messen und Bewerten mit GQM-Light

Teade Punter
Fraunhofer IESE, Sauerwiesen 6, 67661 Kaiserslautern, Germany,
punter@iese.fraunhofer.de

Abstract: *Im allgemeinen haben viele Software Unternehmen Probleme mit der Einführung von Qualitätsverbesserungsinitiative. Aber gerade für kleine und mittelständige Unternehmen (KMU) ist sie problematisch durchzuführen. Dies wird weiter kompliziert mit der Tatsache, dass die disziplinierte Qualitätsverbesserungsinitiative – wie Capability Maturity Model (CMMi), SPICE/ISO 15504– sich an große Unternehmen richten. Als Lösung erscheinen die Agilen Methoden –wie eXtreme Programming, Scrum – die eher kundenorientiert und einen Entwicklungsprozess mit weniger Dokumentation vorschreiben. Dieser Artikel behauptet, dass zielorientiertes Messen und Bewerten (Goal Question Metric; GQM) in Zusammenhang mit der disziplinierten und/oder der Agilen Vorgehensweise benötigt ist, um festzustellen, ob ein Verbesserungsprogramm erfolgreich ist. Dazu ist es für Kleine und Mittelständige Unternehmen notwendig, eine leichtere Version des GQM anzuwenden. Der Artikel beschreibt dazu ein Ansatz der als GQM-Light bezeichnet wird.*

Analyse von ABAP- und Java-Anwendungen im Hinblick auf die Software-Wartung

Björn Wolle
CC GmbH, Flachstr. 13, 65197 Wiesbaden
bjoern.wolle@caseconsult.com

Abstract: *Die Kosten- und Aufwandschätzung von Softwarewartungsprojekten basiert wesentlich auf Techniken zur Abschätzung der Anzahl von Quellcodeanweisungen für ein gegebenes Softwareprodukt. Eine besondere Schwierigkeit ist es dabei, hinreichend genaue Schätzungen vor Beginn der Wartungsprozesse zu erstellen. Da sich bei den meisten neuen, in Java oder ABAP implementierten Softwareprodukten die Aktivitäten derzeit noch überwiegend auf die Entwicklung konzentrieren, fehlt es an Erfahrungswerten für den Einsatz etablierter Metriken zur Ermittlung wartungsbezogener Kenngrößen. Dies betrifft selbst grundlegende Größenmetriken wie die LOC-Metrik und die Halstead-Länge. In diesem Beitrag werden die Einsatzmöglichkeiten und Schwachstellen dieser Metriken für Java- und ABAP-Systeme anhand experimenteller Daten diskutiert. Empirisch gefundene Korrelationen zwischen den einfachen Größenmetriken werden vorgestellt. Basierend auf diesen Ergebnissen ergeben sich neue Skalierungsansätze, die eine Umrechnung erlauben und sich für die Schätzung von Programmlängen zum Einsatz für die Softwarewartung eignen.*

(Erweiterte) Funktionalgrößen Meßmethoden sind auch in Verbesserungsprojekten anwendbar

Ton Dekkers
Sogeti Nederland B.V. (Sogeti Deutschland GmbH)
ton.dekkers@sogeti.nl

Abstract: *Betrachtet man die Verteilung der IT Kosten, stellt man fest, dass der größte Teil des Budgets für Wartungs- und Verbesserungsprojekte zugewiesen wird. Neuentwicklungen umfassen 30-50% der IT Kosten. Methoden zur Messung des Funktionsumfangs werden meistens nur bei Neuentwicklungen angewendet. Mit kleinen Erweiterungen bekannter Meßmethoden wie der Function Point Analyse oder COSMIC Full Function Points kann fast jede IT Aktivität analysiert werden. Während der vergangenen 10 Jahre hat Sogeti die erweiterte Meßmethode basierend auf der Function Point Analyse sehr erfolgreich in zahlreichen Projekten angewendet.*

Softwareprofis müssen messen können – Softwarequalität im Unterricht an Hochschulen

Hans-Georg Hopf
Georg-Simon-Ohm-Fachhochschule Nürnberg
Hans-georg.hopf@fh-nuernberg.de

Abstract: *Das Thema Softwarequalität und Softwaremetriken wird nicht nur häufig in der industriellen Praxis stiefmütterlich behandelt, sondern ist auch im Softwareengineering - Unterricht an Hochschulen nicht immer ganz einfach zu vermitteln. Oft liegt die Schwierigkeit darin, für noch nicht entwicklungserfahrene Studenten abstrakte Qualitätskonzepte nachvollziehbar darzustellen und zu vermitteln und Praxisbezüge zu schaffen. Leider finden Softwareengineering Praktika oft erst zu einem Zeitpunkt statt, wo das „theoretische" Wissen aus vorangehenden Lehrveranstaltungen zum Thema Softwareengineering oder Softwarequalität schon in Vergessenheit geraten ist.*
Mit einer e-learning Umgebung ist es möglich, einerseits das Vermitteln des Basiswissens zu unterstützen und andererseits konkretes Wissen für die Durchführung eines Softwareentwicklungsprojekts „on demand" in einer an der Aufgabe orientierten, angemessenen Tiefe verfügbar zu machen. Auf diese Weise werden einmal gelegte theoretische Wissensgrundlagen mit vertieftem praxisrelevantem Wissen bedarfsgerecht vernetzt.
In diesem Artikel werden eine e-learning Umgebung und Lehrmaterial zum Thema Softwarequalität und Softwaremetriken präsentiert, das didaktische Konzept erläutert und seine Umsetzung in einer virtuellen Lehrveranstaltung vorgestellt.

Aufwandschätzung von Projekten - zwischen Fehlanzeige und Perfektion –

Achim Kindler, Wolfram von Schneyder
IMAKA Institut für Management GmbH, Leonberg; ILTIS GmbH, Rottenburg a.N.
akindler@imaka.de; wolfram.von.schneyder@iltis.de

Abstract: *Je größer der Wettbewerbsdruck, desto entscheidender wird der Beitrag von Projekten zum Unternehmenserfolg. Mit diesem Druck steigt auch der Bedarf nach frühzeitigen und zuverlässigen Aufwandschätzungen in den Projekten.*
In der Literatur wird hierfür eine größere Anzahl an Methoden und Verfahren empfohlen. Ein Blick auf den Anwendungsbereich lässt allerdings erkennen, dass sich nahezu alle Methoden und Verfahren auf Projekte zur Software-Entwicklung beziehen.
Der vorliegende Beitrag gibt eine Übersicht über den gesamten Bereich der Aufwandschätzung und beschränkt sich nicht auf Software-Entwicklungsprojekte. Neben einer kurzen Darstellung der bekannten Verfahren wird eine Analyse der Gründe vorgenommen, warum die Entwicklung von Methoden und Verfahren zur Aufwandschätzung in anderen Anwendungsbereichen „hinterherhinkt". Abschließend wird der Handlungsbedarf zum Schließen der bestehenden Lücken in Forschung und Praxis dargestellt.

Softwaregrößenmaße im Kontext des CMMI

Günter Büren
Büren & Partner Software-Design GbR, Nürnberg
gb@bup-nbg.de

Abstract: *Der funktionale Gehalt einer Software, ausgedrückt durch ein geeignetes Größenmaß, ist eine wichtige Basis für die Aufwandsschätzung in Projekten und da-mit für die Projektplanung und Verfolgung. Das CMMI, die neuere Variante des Reifegradmodells CMM, trägt diesem Umstand Rechnung durch die explizite Forderung der Verwendung einer bewährten Methode zur Größenbestimmung bereits für Reifestufe 2. Eine Prozessverbesserung nach den Vorstellungen dieses Modells erfordert daher unbedingt auch eine Implementierung einer für die jeweilige Umgebung geeigneten Methode zur Größenbestimmung von Software.*
Bei genauerer Betrachtung stellt man fest, dass Größenmaße in der Software-entwicklung und in der Prozessverbesserung eine vielfältige Bedeutung haben. Größenmaße bieten die Basis, Verbesserungen des Prozesses erkennbar zu machen, und fördern damit die Erreichung höherer Prozessreifegrade. Größere Prozessreife erleichtert dagegen in aller Regel die Anwendung von Größenmaßen.
Man kann also in der Softwareentwicklung eine Wechselwirkung feststellen zwischen der Anwendung von Größenmaßen und der Prozessreife. Der Beitrag arbeitet die Vorstellungen des Prozessmodells CMMI-SE/SW zum Einsatz von Größenmaßen heraus und macht die Rückwirkungen deutlich, die eine systematische Anwendung von Größenmaßen auf die Software-Prozessverbesserung hat.

Function Point-Zählung von Data Warehouse Projekten

Manfred Bundschuh
AXA Service AG, Köln
Manfred.bundschuh@axa.de

Abstract: *Die Anforderungen an Data Warehouse Systeme unterscheiden sich erheblich von denen an transaktionsorientierte Systeme. Luca Santillo, ein zertifizierter Function Point Zähler der Firma DPD in Rom, hat 2001 einen Vorschlag auf der Tagung in Heidelberg präsentiert, wie Function Point Zählungen bei der Data Warehouse Konzeption mit dem Sternschema durchgeführt werden können. Seine Vorschläge zur Bewertung von Datenbeständen und Transaktionen werden präsentiert.*
Sein Vorschlag stand Pate bei der Entwicklung eines Standards für die FP Zählung in der AXA Service AG, wo das OLAP Design eingesetzt wird. Dieser inhouse Standard für die Bewertung der Datenbestände der OLAP Cubes und der Transaktionen wird ebenfalls vorgestellt. Damit wird gezeigt, daß die FPM auch bei DW Applikationen sinnvoll eingesetzt werden kann.

Kostenverteilung im IT-Life Cycle

*Andreas Schmietendorf *# , Reiner Dumke #*
* T-Systems Nova GmbH, Entwicklungszentrum Berlin
E-Mail: andreas.schmietendorf@t-systems.com
Otto-von-Guericke-Universität Magdeburg, Fakultät Informatik, IVS
E-Mail: {schmiete ,dumke}@ivs.cs.uni-magdeburg.de

Abstract: *Der vorliegende Beitrag beschäftigt sich mit der Vorgehensweise zur Abschätzung potentieller Kosten einer neu zu entwickelnden Softwareanwendung im Bereich kommerzieller Informationssysteme. Dabei wird sowohl auf die Kosten im Rahmen der Entwicklung (Aufwandsschätzung), als auch auf die Kosten im Rahmen des Wirkbetriebs (Systemkonzeptschätzung) eingegangen, so daß der komplette Life Cycle der IT-Anwendung erfaßt wird. In einem ersten Schritt werden klassische Verfahren der Aufwandsschätzung hinsichtlich der möglichen Aussagen bewertet. Um die Bedeutung derzeit durch diese Verfahren nicht erfaßter Kostenelemente zu unterstreichen, wird auf eine empirische Untersuchung der Universität St. Gallen zu den Lebenszykluskosten von Anwendungen, welche innerhalb der Deutschen Telekom AG und der Deutschen Bahn AG durchgeführt wurde, eingegangen. Um neben den Entwicklungskosten auch die Kosten des späteren Wirkbetriebs zum Zeitpunkt der Planung einer neuen Anwendung einschätzen zu können, bedarf es der Verwendung von Methoden des Performance Engineerings zur Systemkonzeptschätzung. Diesbezüglich bietet der Beitrag einen Überblick, welche Kostenaussagen durch Methoden des Performance Engineerings gewonnen werden können. Darüber hinaus wird zur Verbesserung der Situation eine Kombination bzw. Integration von Aufwandsschätz- und Systemkonzeptschätzverfahren vorgeschlagen.*

eMeasurement – Gegenwärtiger Stand und Perspektiven einer Web-basierten Software-Messung

Dumke R., Lother, M., Wille, C., Braungarten, R., Winkler, D.
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik, Institut für Verteilte Systeme
Postfach 4120, D-39016 Magdeburg
Email: {dumke,lother,wille,braungarten,winkler}@ivs.cs.uni-magdeburg.de

Abstract: *Das Gebiet des eMeasurements beim Software Engineering umfasst eigentlich alle Bereiche einer Web-Unterstützung für die Messung und Bewertung. Speziell auf diesem Gebiet ist diese Web-Anwendungsform noch sehr schwach ausgeprägt.*
Der vorliegende Beitrag widmet sich den im Web vorhandenen und im eigenen Umfeld entwickelten Möglichkeiten für das eMeasurement. Zunächst werden Ansätze und Anwendungsformen zum eMeasurement generell betrachtet. Das schließt Bereiche der Physik oder der Elektrotechnik mit ein. Dann werden ausgehend von einer Web-Infrastruktur und dem allgemeinen Messstandard ISO 15939 erste Lösungen zum Software-eMeasurement vorgestellt und bewertet.
Aufgrund von Analysen im Web sowie der Anwendung bereits erkannter Gesetzmäßigkeiten werden vorhandene, aktuelle Beispiele in ihren ersten Ansätzen diskutiert und mögliche Formen und Initiativen in diesem Bereich vorgeschlagen.

M wie Muster oder M wie Metrik – ein GQM² - Ansatz für objektorientierte Systeme

Jörg Robra
Georg-Simon-Ohm-Fachhochschule Nürnberg, Fachbereich efi
joerg.robra@t-online.de

Abstract: *Objektorientierte Entwurfsmuster sollen zur Qualität eines objektorientierten Entwurfs beitragen; aber lässt sich die Verbesserung auch durch objektorientierte Entwurfsmetriken*

nachweisen? Dies ist nur zu erwarten, wenn die Metriken auch das messen, was die Entwurfsmuster versprechen.
Es ist daher erforderlich, die Qualitätsziele zu definieren, die durch den Einsatz von Entwurfsmustern erreicht werden sollen, und dann über einen GQM – Ansatz Metriken zu finden, mit denen sich die Verbesserung des Entwurfs durch den Einsatz von Entwurfsmustern nachweisen lässt.
Es zeigt sich dabei, dass die bekannten Entwurfsmetriken überwiegend ungeeignet sind; daher werden auf der Basis dieses GQM-Ansatzes zwei Metriken neu definiert. Diese zeigen bei den exemplarisch ausgewählten Entwurfsmustern signifikante Verbesserungen für die Fachklassen und eine Verlagerung der Probleme gegen Wiederverwendbarkeit und Flexibilität in Hilfsklassen; dies muss bei der Beurteilung vollständiger Strukturen beachtet werden.

Softwaremetriken im Umfeld der Automobilindustrie

Niklas Fehrling
Robert Bosch GmbH, Postfach 30 02 40, 70442 Stuttgart
Tel. +49 (0)711 811 24124, Mobil +49 (0)160 7029104
Niklas.Fehrling@de.Bosch.com

Abstract: *Inhalt dieser Präsentation ist ein Erfahrungsbericht zur erfolgreichen Umsetzung eines Metriksystems innerhalb unterschiedlicher Software- Entwicklungsbereiche bei der Robert Bosch GmbH. Dargestellt werden die Erfahrungen bei der Einführung ein Zielorientiertes Messystem im Rahmen einer Verbesserungsinitiative basierend auf dem Capability Maturity Model Integration (CMMI) und Beispiele erfolgreich implementierter Metriken als Grundlage systematischer Entscheidungen von der Mitarbeiterebene bis zum Senior Management.*
Es werden nützliche Hinweise und Anregungen für Organisationen zur Verfügung gestellt, die sich auf den Weg zu einer Definition eines SW-Metriksystems gemacht haben. Ebenso werden Organisationen die bereits ein SW-Metriksystem umgesetzt haben und den Erfolg sichern möchten angesprochen.

DASMA-DIPLOMARBEITSPREIS 2003

Melanie Ruhe: Universität Kaiserslautern, AG Software Engineering
und Siemens AG, München, melanie.ruhe@ siemens.com

Thema: *Die präzise und frühzeitige Aufwandschätzung von Web-Anwendungen*

SLA Management – Challenges in the Context of Web-Service-Based Infrastructures

ANDREAS SCHMIETENDORF[#*], REINER DUMKE[*]

[#] *T-Systems International, Entwicklungszentrum Berlin, Wittestraße 30G, D-13476 Berlin, E-Mail: andreas.schmietendorf@t-systems.com*

[*] *Otto-von-Guericke-Universität Magdeburg, Fakultät Informatik, Institut für verteilte Systeme, Postfach 41 20, D-39016 Magdeburg, E-Mail: {schmiete ,dumke}@ivs.cs.uni-magdeburg.de*

Introduction

Although the area of network and system management has experienced a regressive trend over the past two years, the use and management of service level agreements can, according to [Gartner 2002], most certainly be considered an enabler technology for market growth in the area of service-based integration solutions. The use of service level agreements is not yet taken for granted even in the case of traditional IT solutions. An analysis has shown that service level agreements that are comprehensible and, most importantly, measurable by the customer are concluded for only 5 to 10 % of all applications. You may well ask why this low regard for SLAs actually needs to be changed for Web-service-based applications. From the viewpoint of the authors, the following properties inherent in such architectures and the disadvantages associated with traditional technologies have a lot to do with this:

- Web-service-based applications are designed for use across various companies, whereas previous applications were primarily used within a single company.
- Current pressure on costs is giving rise to a customer demand for transparent costs for service provision, and Web-service-based applications support this goal.
- Up to now, SLAs have been primarily based on resource-related measurement variables that were generally incomprehensible to customers in the context of the functions they used.
- In the case of Web-service-based applications, the main focus is on the interaction chain involving measurement variables that are based on functions or business processes and that take the actual customer benefit into account.
- Commercial application systems based on integrated Web services urgently require the services used to be subject to quality assurance (efficiency, security, availability, etc.).

As SLAs for Web-service-based solutions can also be concluded “on demand”, i.e. at run time, this gives rise to further requirements for the underlying technology, the content of agreements, SLA monitoring, and also the measures that need to be drawn up during development. Whereas SLAs used to have a predominantly technical orientation and were mostly the domain of the operators of IT solutions, Web-service-based solutions also require system integrators to take account of the use of SLAs during development. Only if the issue of SLA management is explicitly addressed in development can SLAs that are comprehensible for customers and backed up by measured values be concluded later during actual operation (please also refer to [Duncan 2001]).

For the operators of IT solutions, this situation is associated with a changeover from system management to service management. This is the only way to take better account of the business process supported. The current approach to SLA management is still very much

resource-driven and is barely comprehensible to customers. In order to implement customer-oriented service management, it is vital to be familiar with the business processes supported by the Web-service-based solution. Most operators are currently not even aware of these processes.
After providing an introduction to Web-service technology, this contribution primarily aims to describe the topic of SLA management in such environments. In this context, we will highlight the general contents of SLA agreements, describe the interaction chain for service provision that arises in Web-service-based solutions, and provide a brief explanation of the opportunities offered by the WSLA framework that IBM has developed especially for Web services. Within the various sections we will also look at development tasks – an aspect that is usually neglected but without which the authors consider efficient SLA management to be impossible.

Web Services – Overview

Web-service-based solutions entail using the Internet as middleware for implementing B2B (Business to Business) applications. This could not be done very easily using previous middleware solutions such as CORBA, RMI, or DCE. The rigid coupling and predominantly synchronous communication methods made it necessary to resolve the communications relationships right at the time of development. Using Web-service technology, applications are able to communicate with one another easily and at low cost via the Internet, regardless of the technology used. This is made possible by using the HTTP-based SOAP protocol (Simple Object Access Protocol) which, using HTTP, can also be routed across firewalls. By means of XML-based messages (eXtensible Markup Language), it is possible to achieve both asynchronous and quasi-synchronous communication. Based on the WSDL description (Web-Service Description Language), applications located on the Internet are able to specify services of their own and publish them in suitable directories, for instance UDDI (Universal-Description-Discovery-and-Integration). Central UDDI Business Registries are currently operated by companies such as IBM and Microsoft. In order to access a specific Web service, the initiating system requires its WSDL description, which can also be queried at runtime via UDDI.
Advantages expected from using Web services:

- Simpler and less costly than dedicated integration frameworks
- Higher degree of standardization
- Synchronous and asynchronous communication model across all company boundaries
- Support of the component paradigm
- Simple means of communicating across firewalls
- Widespread acceptance in industry
- Bridge between different technology approaches, such as J2EE and .net.

Web services can be viewed as technical components that reside on the Internet (see [Turowski 2002]), with it being possible to develop entire application systems based on a loose coupling. A Web service should constitute a clearly identifiable part of the business or service process that is being supported. This means that the interface for a Web service is coarsely granular in nature, although no quantifiable dimensions for this feature can be given at present due to a lack of experience. In the case of the requirement for the functions offered

by a Web service to be supported by suitable SLAs, this characteristic is especially important because this means being able to restrict the variety of potential SLAs. The authors believe that such an interface should contain 5 to 10 business functions (excluding elementary functions).
The live operation of a Web service can largely be compared with that of other applications. The essential difference involves use on the Internet of commercially offered Web services, in which case the provider concerned is responsible for the functional and non-functional aspects. For this reason, agreements regarding quality of service are needed between the provider and potential users of the Web service. A distinction can be made here between a static and a dynamic approach. In the case of static agreements, functional and non-functional aspects are determined in their entirety at the time when live operation commences, whereas in the case of a dynamic approach, an appropriate contract is concluded virtually “on demand” at the time of execution.

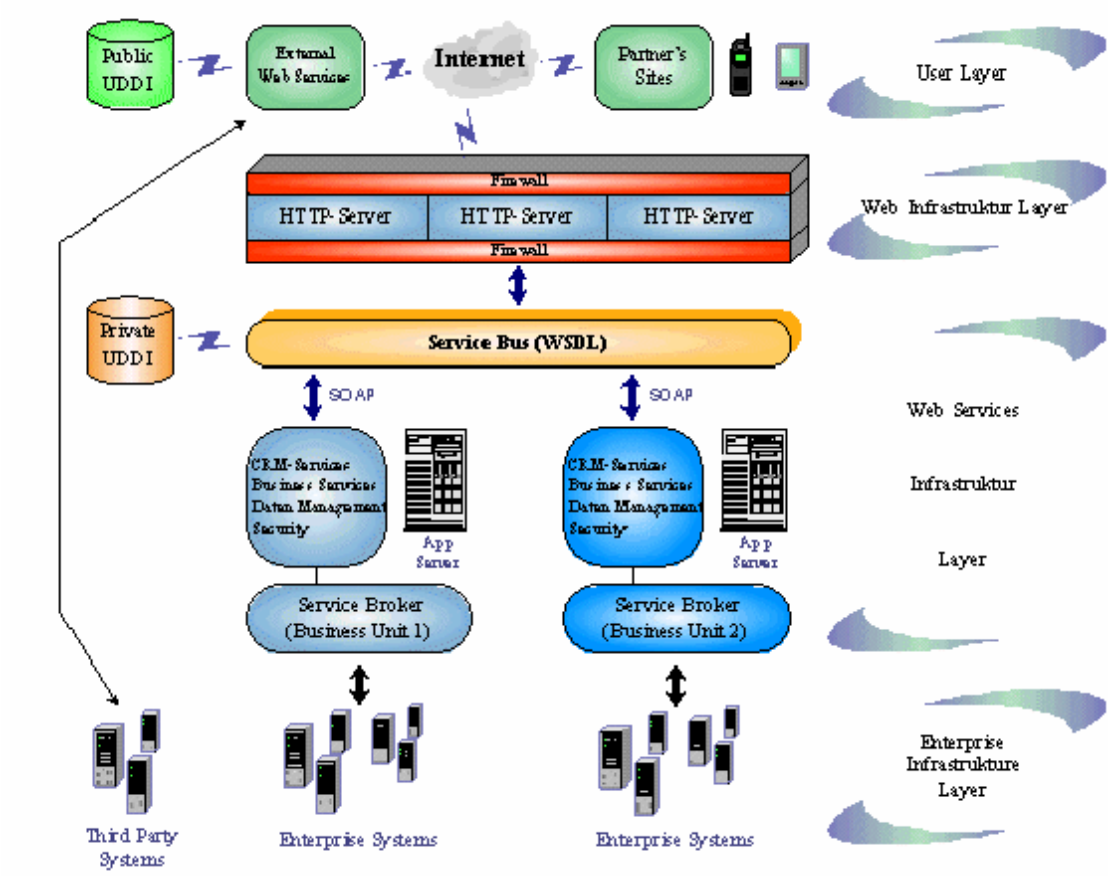


Figure 1: Web service architecture model

In their current form as loosely coupled collections of services, it is more appropriate to view Web services as an ad-hoc solution that can be developed quickly and easily. The present generation of Web services merely allows applications to be integrated at function level. In their present form, they are not transaction-oriented and merely provide fundamental "request/response" functionality. Challenges such as transaction backups, the secure transmission of messages between Web services, control of the process logic between Web services, and the provision of non-functional requirements and therefore the support of effective SLA management have not yet been developed sufficiently. There are nevertheless

initial attempts at standardization with regard to these issues – see also [Schmietendorf 2003]. In the rest of this article we intend to concentrate on the topic of SLA management.

Overview of SLAs

The conclusion of SLAs is based on fixed service and performance agreements between customers and suppliers and creates transparency for both parties in terms of performance and costs. Specific SLAs are used to define the type, scope, and quality of services and to check that specifications are met. As SLAs also include potential sanctions for the event that agreed service parameters are not met, the specifications made in them have a significant effect on the commercial success of a company providing services for a customer. In the context of service-oriented architectures, the benefits of successful service level management can be described as follows:

- The number of conflict situations within supplier relationships can be reduced, resulting in enhanced customer satisfaction.
- The resources used in order to render the service (hardware, personnel, licenses) can be distributed at a detailed level by the provider and therefore used in such a way as to optimize costs.
- Problems can be identified speedily by service level monitoring and the associated cause determined.
- Costs can be made more transparent – on the one hand, the customer only wants to pay for services actually used while, on the other hand, plausible pricing can be guaranteed.

As part of SLA management for Web-service-based applications, a service level agreement must be concluded in addition to providing the actual Web service. As stated by [Heinrich 2001] and [Keller 2002], the following specifications are required:

- Partners involved and the validity of the agreement, i.e. the period over which the service is to be provided
- Specification of the contract components and procedure for any necessary modifications
- Specification of the functional scope and quality of the service to be provided
- Definition of the SLA parameters with which provision of the service will be proved
- Specification of the procedure for determining/calculating the SLA parameters
- The consequences of contract disruptions and legal basis
- Settlement arrangements

It is obvious that the contents described for a service level agreement may vary significantly in terms of the precise details. In the case of Web-service-based applications, it is especially necessary to maintain the relationships between the Web services involved in providing the service and to promote a broad standardization of possible SLA agreements. Both aspects of syntax and aspects of semantics need to be considered. Whereas the aspects of syntax can be dealt with today by using a technology-independent language such as XML, the standardization of semantic aspects is still in its infancy. One example is the task of

interpreting the availability of a Web service that is specified as being 98 %. Users may well ask questions such as whether this takes maintenance time into account, whether it is based on round-the-clock operation, and how potential downtime (i.e. recovery time) is dealt with.
This flexibility must be explicitly taken into account during development and the Web service interface must be supplied with this information. In addition, monitoring must be supported by suitable measurement points so that potential bottlenecks or infringements of the SLA can actually be identified. It is only if these aspects are taken into account when developing a specific Web service that SLA management can be implemented for such a solution.

Service Agreement and Service Provision

Figure 1 illustrates the interaction chain of a Web-service-based application. We have specified different instances of SLAs here – an OLA (Operation Level Agreement) in the context of an internal service, a UC (Underpinning Contracts) in the case of a subcontracted service, and an SLA (Service Level Agreements) with regard to the actual user. Although Web services can also be used directly by appropriate end users, we will proceed by assuming the integration (loose coupling) of Web services to form new application systems. This means that it would actually be more appropriate to speak of UCs rather than SLAs.

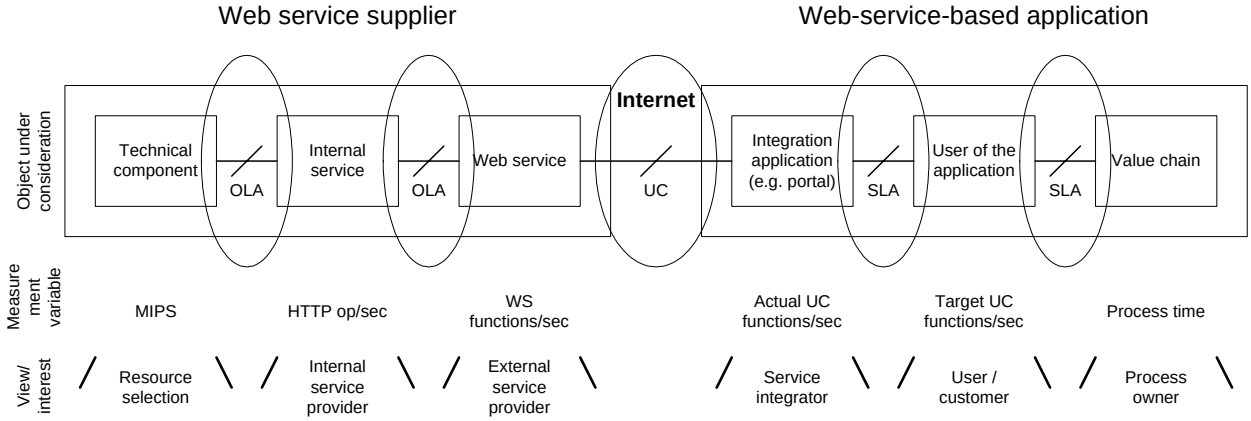


Figure 2: Service-level-based interaction chain

Now that the interface in question has been identified in the interaction chain as a whole, we intend to look at the potential procedure for a service level agreement. The scenario shown in Figure 2 is based on the lifecycle of an SLA that [Debusmann 2002] illustrated in the context of a multi-provider environment.

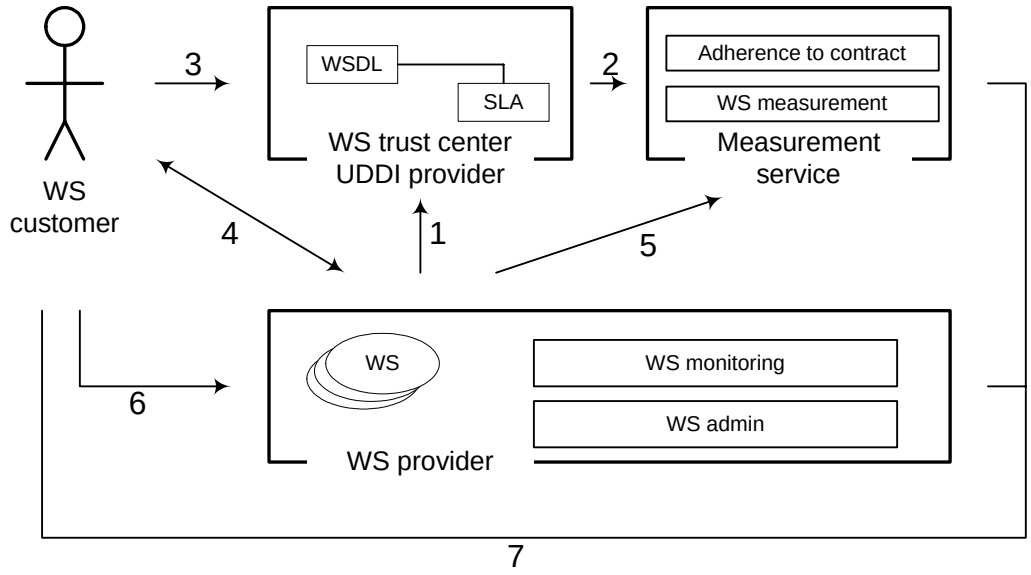


Figure 3: Scenario for using an SLA-supported Web service

The individual steps performed here are as follows:

1. Publication of an available Web service including SLA conditions
2. Inclusion of a measurement service
3. Customer query with regard to an available Web service from an independent supplier
4. Conclusion of a contract between the customer and Web service provider
5. Activation of the measurement service for monitoring adherence to the contract
6. Use of the Web service within the customer's own application
7. Billing of the services provided, taking the SLA specifications into account

Web Service Level Agreements (WSLA)

In order for the requirements described above to be implemented technically, a commercially available solution is required. We will now provide a brief introduction to the WSLA framework that IBM has developed for the purpose of service level management in Web-service-based environments. The WSLA framework is part of the IBM Web Service Toolkit (currently Version 3.2). This framework, available as a prototype, contains both an XML schema definition for describing SLAs and a runtime environment for the actual SLA management. The following descriptions have been drawn up using [Keller 2002] and [Debusmann 2003] as well as a practical test implemented in this environment [WSTK 2002].

WSLA schema specification

The XML-based WSLA schema specification offers a generic basis for a specific SLA language description. Based on experience, the following structural elements are taken into account here (see [Keller 2002]):

Parties section

- Information regarding the contract parties (e.g. contact persons)
- Any subcontractors involved by the supplier

Service Description section

- Definition of the SLA parameters used
- Assignment of the SLA parameters to the services used
- Procedure for determining (measuring/calculating) the SLA parameters

Obligations

- Conditions to be adhered to
- Procedure for dealing with infringements of SLA parameters

Web services runtime environment

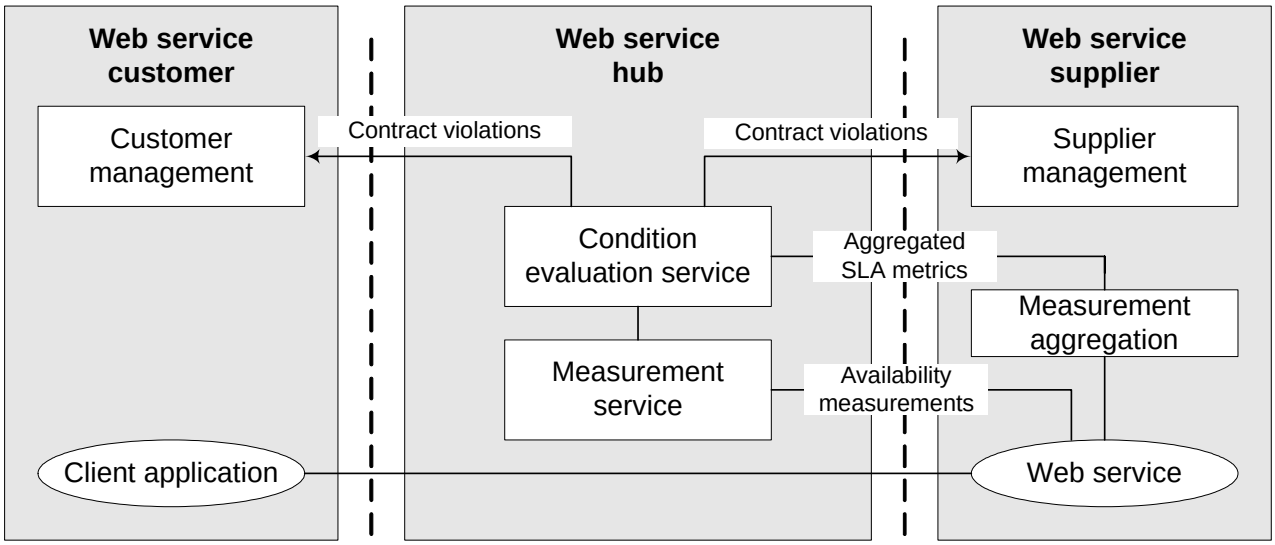


Figure 4: Overview of the WSLA framework runtime environment [Debusmann 2003]

The runtime environment includes a deployment service (for installing and configuring the technical environment), a measurement service (for measuring the quality of service), and a condition evaluation service (for identifying contract infringements). Using these services together with the proposed roles – the party using the service (service customer), the party offering the service (service supplier), and an independent third-party supplier (service hub) – the various tasks in the process of offering and using Web services are covered. The procedure used is tailored to the typical classification of the marketing of conventional products. The process begins with the provider of a service, referred to here as the *service supplier*, whose range of tasks include developing, producing, and offering the service. The result is the service, which can be accessed and used on the Internet via a URL address. The producer does not have a direct connection to the customer in this model. It is the *service hub* that is responsible for making contact with the customer and actually processing the business

transaction. This party can be viewed as a product dealer. The service supplier makes usable services available to the service hub by registering the services in a database. A UDDI directory service could be used for this task, for instance. In the case of the Web-service runtime environment from IBM, this task can be carried out using a Web-browser-based interface. The data required includes a brief description and the WSDL definition of the service. A reference to a demonstration page can also be specified as an additional option. The WSDL information is specified in the form of a URL, via which the actual document can be obtained. The service hub can now create offerings for a customer. A name and the corresponding registered service are then defined for this. In addition, other properties can be defined and the usage period can be specified.

For instance, an SLA can be defined for a required performance level and, in terms of measurement technology, can be based on the throughput volume. For instance, the throughput may be measured in terms of accesses per minute. It is possible to define upper and lower limits that the customer can select later. The priority for adherence to the restrictions is determined by the performance level. The offering can then be presented to the user by the service hub activating it. Services that have already been sold provide the hub with information on service usage – for instance the frequency of the application and the costs involved. So that a customer is able to work with a service, the service hub and the service requester as the user draw up a contract. The customer thereby accepts the offer and determines other details such as the means of payment, the service model, and the expected number of accesses per minute. In addition, the runtime of the agreement can be adjusted. Once the contract has been activated, the user can now access the service in the demonstration environment using a browser.

Guaranteeing Web Service Level Agreements

The technical implementation is realized by Web Service Level Agreements (WSLA) for defining and monitoring SLAs. The runtime control is carried out within the Web service management middleware (WSMM) in order to guarantee a service. Another aspect is the service desk, via which several services can be grouped together automatically and therefore used together. The possible functions allow users to autonomously create, compile, manage, route and search Web services as well as switch between them. The advantage for clients is that this system reduces the complexity of the connection, the interoperability, distribution, and combination of various heterogeneous Web services. The three-layer architecture that maps the respective roles intercommunicates via SOAP messages. The system's help functions are implemented as handlers in Apache Axis. These are general supporting services that are executed before or after a Web service is initiated. They include identification, routing, and logging services. Axis provides the option of starting these tasks via handlers, separately from the services, before or after executing a Web service. The effect of such a handler is completely transparent for the client and service, which means that the external view is the same both with and without the handlers.

The IBM demonstration environment uses this option to implement management functions. For the client, therefore, it doesn't matter whether the partner with whom he communicates is a hub or the service itself because the access is the same. This supplier level is divided into two layers in the architecture: the hub and the service supplier. A client communicates with the hub, which assumes responsibility for identifying the client, executing the auxiliary

services, and forwarding data to the correct service. This is done in a type of pipeline. When a client sends a query to the hub, a profile service, started by a handler, assigns the client a profile ID for further processing. The value is inserted in the message context so that the ID can be used by other handlers. The next step entails a contract service checking the client's authorization to access a particular service. A measurement service is then prepared to determine the performance. The query is saved in the subsequent management request service for statistical purposes. The concluding Web service management middleware controls the time at which the data is forwarded to the actual service. The contract ID is queried for this, allowing the associated WSLA conditions to be evaluated. Based on these performance requirements and the current load of the service supplier, the WSMM decides when the query can be submitted to the service desk for execution. If this time is reached, the service desk assumes responsibility for routing to the actual service. The WSMM and service desk work together as a means of load distribution so that the specified WSLA restrictions can be adhered to as far as possible. The service's reply is not forwarded directly, either. The reply does not reach the client until the measurement service, management, and WSMM handler have been executed.

Monitoring the SLA parameters

The WSLA descriptions add SLA capabilities to the Web services. In a formal way, the performance requirements are defined by the WSLA language. This enables suitable monitors to evaluate these definitions and determine whether the current measured values correspond with the specifications (service level targets). To this end, performance guarantees are defined for the Web service operations and business processes via the WSLA. The recording of performance measurement is unambiguous and therefore allows violations of performance guarantees to be established. Third parties can be incorporated into the evaluation and monitoring process. This allows a supplier of services to use the specification to define SLAs and reach an agreement with the customer. Several performance levels are defined, together with the associated templates. A customer selects the required SLA from this portfolio when concluding a contract. For the supplier, these specifications provide an indication of the resources needed to operate the Web services and the priority of the service for this user. The values are important for the customer for the purpose of setting the measurement and control systems correctly.

Control systems (condition evaluation) establish whether the SLAs are adhered to. In the event that the contractually agreed SLA parameters are violated, warnings are output. With a view to enabling this evaluation, the information relevant for data collection is input to the measurement service. This is done when the contract is activated. An estimate regarding fulfillment of the SLA can then be made using the current measurements. If the specifications are violated or if an error status exists, the actions of the handling agreement take effect. This may involve an error message, for instance. The control system is made up of the following elements: a data collection system for obtaining measurement data, the measurement service, a measurement component for compiling metrics from the measured values – as specified in the WSLA – and a system for comparing the actual calculated metrics with the specifications.
The current network load is controlled by the Web service management middleware. With regard to the assignment of the Web services to the relevant clients, this is based on the concluded SLAs with a view to ensuring that the specified restrictions are adhered to as far as possible. In addition, the current network load is taken into account as a parameter for

planning the Web services. A single service can be assigned various performance levels, which appear transparent from the viewpoint of the client and service. The individual functions of the WSMM include resource assignment, scheduling, and protection against overload for individual queries, as well as the optimization of slow timescales and the coordination of queries. A standardized comparative performance value is assigned to each class of workload using the SLA. This value contains an assessment function that estimates how probable it is that the actual values will exceed or fall below the specified values. For planning purposes, the middleware uses simple procedures for modeling the load and assessing the performance of a Web service using queue systems or similar.
The Web services are connected, distributed, and compiled via the service desk in the Web Service Toolkit. This is done using clusters that connect several services via a shared access point. As a result, an individual cluster is able to determine which service a query starts, trigger appropriate performance measurements for the service processing, and control the handling of downtime. The choice of service to be started affects not only availability but also the quality of service specifications in the WSLA and business rules. The decision is made automatically based on a service policy. The service desk permits an abstract view for suppliers and users for the purpose of processing client accesses that are received in parallel in heterogeneous environments, with the underlying implementation being hidden from users.

Summary

The Web services currently (date: February 2003) offered on the Internet usually only have a rudimentary description of their functional and non-functional properties, which means that determined properties can usually not be assumed for the Web services offered. Numerous offerings are only available temporarily and do not have a commercial character.

From the viewpoint of the authors, it is vital that the opportunities provided by SLA management in the environment of Web-service-based applications are taken into consideration during development. In the process, development is based not only on the tasks related to process modeling and analysis but also on the actual software development. Web services that take both functional behavior and non-functional properties into account and, most importantly, guarantee these during execution need to be positioned successfully on the Internet before commercial solutions can be implemented based on this technology.

The generic WSLA language and the associated architecture offer the opportunity to cover a wide range of negotiating situations between potential contract parties. However, the WSLA approach only addresses selected problems (primarily issues regarding performance) when it comes to guaranteeing a defined service level. The vast majority have to be taken into account during software development. The task for software development is to create Web services with determined properties. From the viewpoint of the authors, this requires the use of agent technology or, in an initial approach, the instrumentation of technically founded user functions.

Sources

[Debusmann 2003] Debusmann, M.; Keller, A.: *SLA-driven Management of Distributed Systems using the Common Information Model*, IFIP/IEEE International Symposium on Integrated Management (IM 2003), Colorado Springs, USA, 24.-28.3.2003, to be published

[Duncan 2001] Duncan, H.: *The Computing Utility: Real-Time Capacity on Demand*, CMG Conference, Anaheim/USA, 2001

[Gartner 2002] Gartner Symposium – *ITxpo 2002*, Documentation CD-ROM, Cannes/France, November 2002

[Heinrich 2002] Heinrich, L. J.: *Informationsmanagement* – 7th edition, R. Oldenbourg Verlag, Munich, Vienna 2002

[Keller 2002] Keller, A.; Ludwig, H.: *The WSLA Framework: Specifying and Monitoring Service Level Agreements for web Services*, IBM Research Report, May 2002

[Schmietendorf 2003] Schmietendorf, A.; Lezius, J.; Dimitrov, E.; Reitz, D.: *Web-Service-basierte EAI-Lösungen*, in Knuth (Hrsg.), M.: Web Services, Software & Support Verlag, Frankfurt/Germany, planned for 1st quarter of 2003

[Turowski 2002] Turowski, K.: *Vereinheitlichte Spezifikation von Fachkomponenten*, Memorandum des GI-AK 5.10.3, February 2002

[WSTK 2002] IBM *Web Service Toolkit*, URL: http://www.alphaworks.ibm.com/tech/webservicetoolkit

Thanks

We wish to thank our placement student, Friedhelm Röhl, for his support in writing this article. He meticulously progressed the prototypical implementation of the travel portal described in Section 2 using a Web-service-based solution and gained initial experience in working with the WSLA approach.

Functional Size eMeasurement Portal - Umfrageergebnisse und aktueller Stand

MATHIAS LOTHER

Otto-von-Guericke-Universität Magdeburg, Fakultät Informatik, Institut für verteilte Systeme, Postfach 41 20, D-39016 Magdeburg, E-Mail: lother@ivs.cs.uni-magdeburg.de

Einleitung

Die am 1. September 2003 initiierte webbasierte Befragungsaktion für die Realisierung eines webbasierten Portals wurde vor allem von der deutschsprachigen Community (insbesondere der DASMA) unterstützt. Anbei finden sich ausgewählte Ergebnisse der Umfrage sowie im darauf folgenden Teil einige Informationen zum derzeitigen Entwicklungsstand des Portals. Die gesamte Ergebnisauswertung der Umfrage findet sich unter *fsmportal.cs.uni-magdeburg.de*, die mit Hilfe von Screenshots angedeutete Funktionalität des Portals wird in näherer Zukunft (März/April) ebenfalls unter dieser Adresse freigeschaltet.

Umfrage – Ausgewählte Ergebnisse

Bei der Bewertung der Umfrageergebnisse ist zu berücksichtigen, dass die Befragungsaktion auf fakultativer Basis erfolgte und sich die Teilnehmer im Fachkreis der eMeasurement-Interessenten befinden und so tendenziell positiv diesem Themenkomplex gegenüberstehen. Allerdings ist es ja gerade dieser Personenkreis, der die Zielgruppe und damit die potentiellen Nutzer des zu realisierenden Portals repräsentiert. Weiterhin ist die relativ geringe Teilnehmerzahl (29 auswertbare Datensätze) zu berücksichtigen. Um eine Begründung für den Bedarf und die Akzeptanz eines eMeasurement Portals zu finden, wurde in dem Fragebogen zuerst danach gefragt, ob ein derartiges Portal überhaupt sinnvoll erscheint.

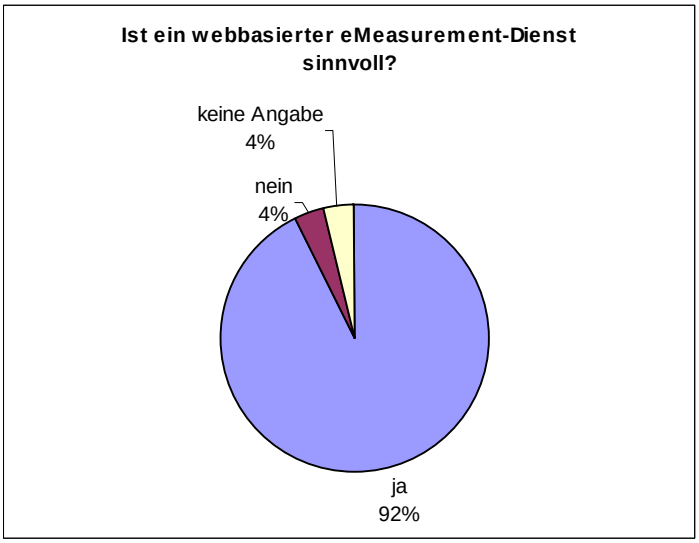


Abbildung 1: Ergebnis - Ist ein FSeMP sinnvoll?

Da die Befragten also offensichtlich ein derartiges Portal als sinnvoll einschätzen, stellt sich daher die Frage, ob es schon ein vergleichbares Angebot eines eMeasurement Portals von

einem anderen Anbieter gibt. Dies ist deshalb wichtig, um die Marktchancen eines solchen Portals besser ausloten zu können. Nur wenn die Konkurrenzsituation am Markt dies zulässt, ist mit einer adäquaten Nutzermenge zu rechnen. Folglich wurde deshalb gefragt, ob ein ähnlicher webbasierter eMeasurement-Dienst den Befragten bekannt ist.

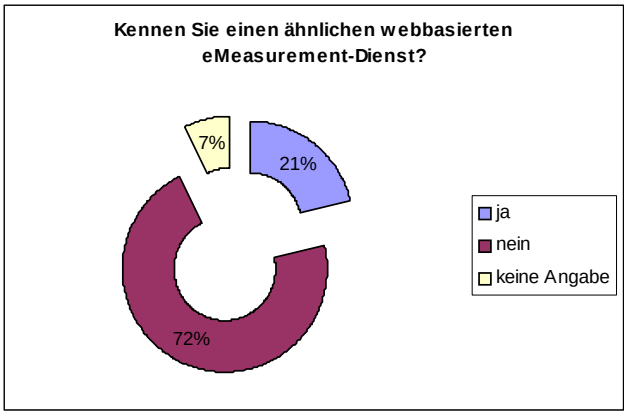


Abbildung 2: Ergebnis - Konkurrenzsituation am Markt

Wie aus obiger Abbildung ersichtlich ist, kennen über 70 Prozent der Befragten keinen vergleichbaren Dienst. Einige der Teilnehmer verwiesen als Alternative auf die ISBSG. Das zu realisierende Portal soll jedoch kein Konkurrenzangebot zur ISBSG darstellen, vielmehr soll es die von der ISBSG zur Verfügung gestellten Services ergänzen. Es lässt sich daher festhalten, dass Konkurrenzangebote zu dem zu realisierenden Portal, zumindest bis zum jetzigen Stand der Ergebnisse, nahezu nicht bekannt sind. Um die gewünschte Grundfunktionalität des Portals zu verdeutlichen, wurde zunächst eine Abgrenzungsanalyse durchgeführt, die die in der folgenden Abbildung dargestellten Ergebnisse lieferte.

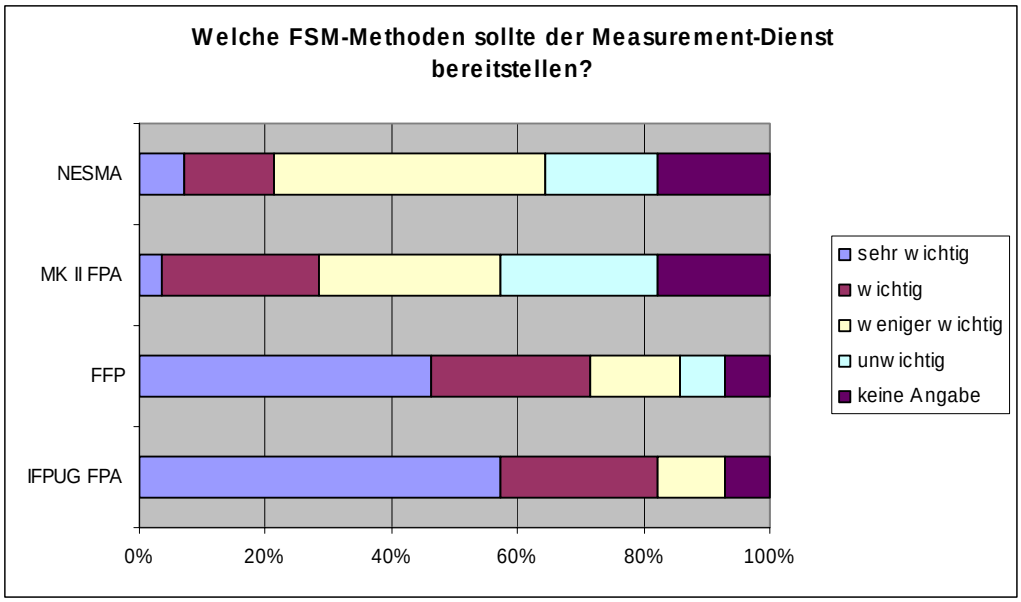


Abbildung 3: Ergebnis - Verwendbare FSM-Methoden

Demnach kommt der IFPUG FPA-Methode in den Augen der Befragten die größte Bedeutung zu. Über 80 Prozent halten diese für sehr wichtig oder wichtig. Auf Rang zwei folgt die FFP-Methode, bei der im Übrigen bei der Befragung nicht zwischen FFP 1.0 und COSMIC FFP unterschieden wurde, mit ca. 70 Prozent. Als nächstes war von Interesse, welche Funktionalität ein solcher Measurement-Dienst bereitstellen sollte. Auch hier wurde den Befragten eine Vorauswahl angeboten. Die Ergebnisse sind in folgender Abbildung visuell dargestellt.

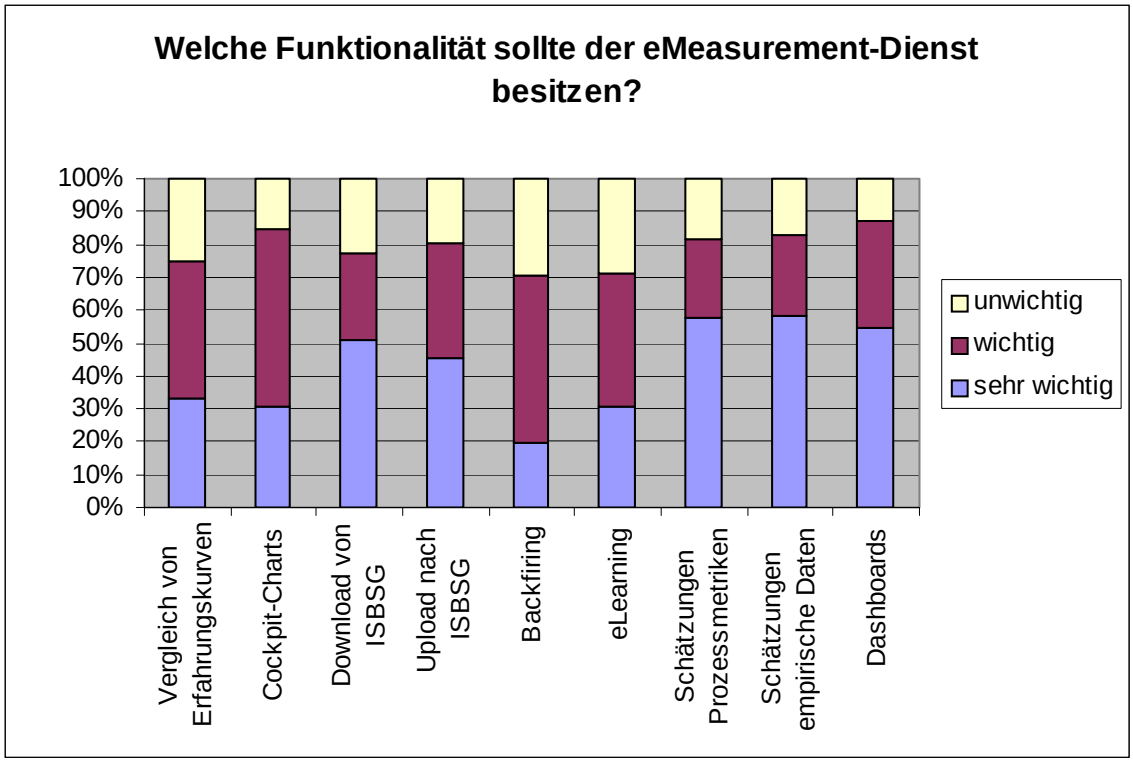


Abbildung 4: Ergebnis - Funktionalität des Portals

Demnach hält die überwiegende Mehrheit der Befragten im Grunde genommen alle zur Auswahl stehenden Funktionalitäten für wichtig oder gar sehr wichtig. Bei der Analyse der Ergebnisse dieses Themenkomplexes stechen insbesondere 3 Funktionalitäten hervor:

- die ***Dashboards***,
- die ***Schätzungen*** auf Grundlage von empirischen Daten und Prozessmetriken,
- sowie die ***Cockpitcharts***.

Im weiteren Verlauf der Umfrage wurden vielfältige Aspekte eines solchen Portals detaillierter hinterfragt, wie

- Vergleich von Erfahrungskurven
- Up/Download zu Projektdatenbanken wie ISBSG
- Cockpitcharts
- Backfiring

– Schätzung aufgrund von Erfahrungskurven
– Sicherheit etc.

Die genauen Umfrageergebnisse finden sich unter `fsmportal.cs.uni-magdeburg.de`

FSeM-Portal – Aktueller Stand

Die folgenden Screenshots deuten den derzeitigen funktionalen Umfang der Realisierung an. Das FSeM-Portal richtet sich zunächst an die drei folgenden Einsatzgebiete:

- Projekte zur ISBSG uploaden und sich gegen die in der dortigen Datenbank abgelegten Projekte benchmarken (Abbildung 5 und 6) lassen,
- Business-Daten verschiedenster Art als Chart aus eigenen Projektdaten generieren lassen (Abbildung 7),
- Aufwandsschätzungen mit Hilfe des Estimation Tools und der ISBSG-Datenbank durchführen (Abbildung 8),
- oder aber eine Kombination aus den verschiedenen Möglichkeiten.

Für den zweiten und dritten Anwendungsfall gibt es hierbei folgende Einschränkungen zu berücksichtigen: Die Visualisierung von Businessdaten macht sicher vor allem Sinn, wenn eine solche Umgebung in die eigene Firma integriert wird, um nicht sicherheitsrelevante Daten nach außen weitergeben zu müssen. Dafür wäre eine Portierung/Adaption der Portalumgebung nötig. Die Zugriffe mit Hilfe des Estimation Tools sind bedingt möglich, da die ISBSG für die Nutzung der Daten ein Entgelt erwartet. Wie ein hierfür notwendiger Zahlungs- und Nutzungsmechanismus aussehen könnte, ist derzeit noch in der Diskussion. Möglich wäre, dass man sich (z.B. durch Upload von Projekten) Credits erwirbt, die dann eine bestimmte Anzahl von Zugriffen auf das Estimation Tool ermöglichen oder aber, dass man der ISBSG eine bestimmte Summe zukommen lässt und entsprechend Credits gutgeschrieben bekommt.

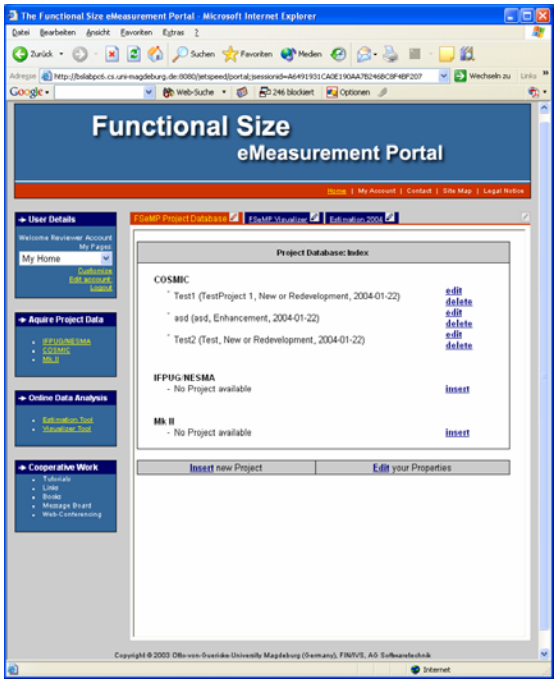


Abbildung 5: Projektverwaltung und -eingabe

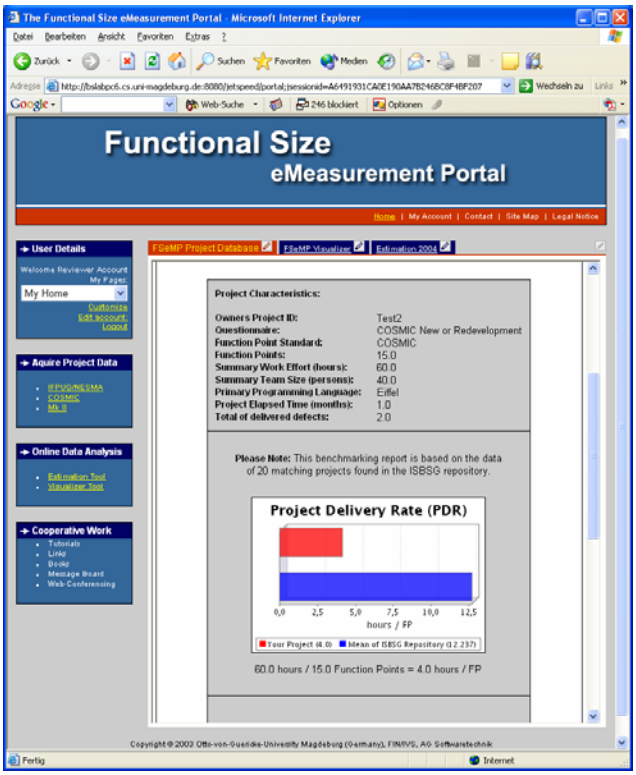


Abbildung 6: Automatisch erzeugter Projekt-Benchmark-Report

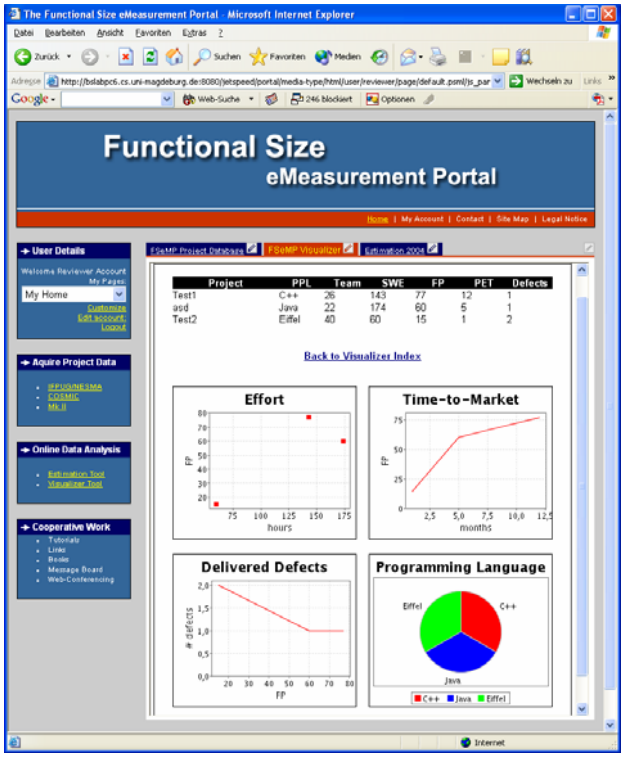


Abbildung 7: Visualisierung von Business-Daten

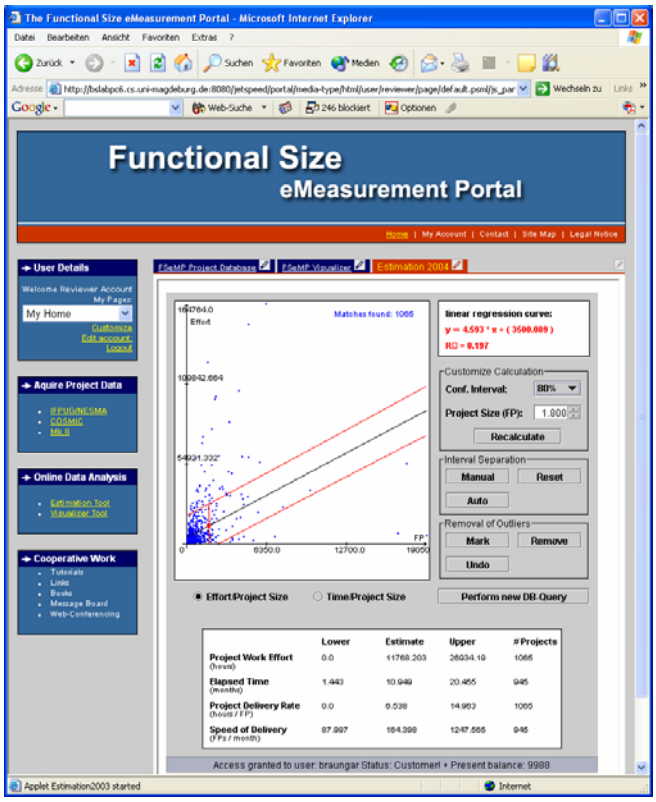


Abbildung 8: Aufwandsschätzung mit Hilfe des Estimation 2004-Tools

FSeM-Portal – Kontaktdaten

Sollten wir Ihr Interesse am FSeM-Portal geweckt und Sie Fragen, Anregungen, Kritik oder möglicherweise Interesse an einer Zusammenarbeit haben, würden wir uns freuen, wenn Sie sich mit dem SML@b-Team in Verbindung setzen würden.

New Books on Software Metrics

Dumke, R.; Abran, A.: (Eds.):

Investigations in Software Measurement

Shaker Publ., Aachen, 2003 (326 pages)
ISBN 3-8322-1880-7

The book includes the proceedings of the 13th International Workshop on Software Measurement (IWSM2003) held in Montreal in September, 2003, which constitute a collection of theoretical studies in the field of software measurement and case reports on the application of software metrics in companies and universities in Argentinia, Canada, Finland, Germany, India, Italy, Japan and the Netherlands.

The contents are described by the listing of the paper abtracts in this Metrcs News.

Albert Endres & Dieter Rombach:

A Handbook of Software and Systems Engineering

Pearson Education Limited, Essex, 2003 (327 pages)
ISBN 0-321-15420-7

Computers are the most pervasive tools of modern society. Their development relies on advanced methods of software and systems engineering. Based on repeated and consistent observations, key lessons of these fields can now be formulated into rules or even laws, providing initial building blocks towards a theoretical foundation that is essential for further research, for teaching and for the practice of software development.

Intended as a handbook for students and professionals alike, this book is the first to identify and discuss such rules and laws. They are largely independent of technologies, and thus form a basis for the principles underlying software and system engineering. Software and system engineers should be aware of this proven body of knowledge, to ensure professionalism and due diligence in their work.

The book is structured around the software development lifecycle. It begins with requirements definition and goes on to maintenance and withdrawal. In different process models, these tasks have different importance or are applied in a different sequence, or even cyclically. The book provides the reader with:

- clear statement of software and systems engineering laws and their applicability
- empirical evidence that proves the usefulness of the material covered
- unique knowledge to apply in an industrial setting.

New Books on Software Metrics

John C. Munson, PH.D.:

Software Engineering Measurement

Auerbach Publications, Boca Raton, Florida , 2003 (443 pages)
ISBN 0-8493-1503-4

The author describes how to manage software development measurement systems, how to build software measurement tools and standards, and how to construct controlled experiments using standardized measurement tools.

The book answers three fundamental questions. First, exactly how do you get the measurement data? Second, how do you convert the data from the measurement process to information that you can use to manage the software development process? Third, how do you manage all of the data?

By demonstrating how to develop simple experiments for the empirical validation of theoretical research and showing how to convert measurement data into meaningful and valuable information, Software Engineering Measurement will show you how to use your measurement information for immediate, software process improvement.

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

Software-Messung in der Praxis

Shaker Publ., Aachen, 2003 (169 pages)
ISBN 3-8322-2146-8

The book includes the proceedings of the DASMA Metric Conference **MetriKon2003** held in Ulm in November, 2003, which constitute a collection of theoretical studies in the field of software measurement and case reports on the application of software metrics in companies and universities.
The contents are described by the listing of the paper abtracts in this Metrcs News.

Pandian, C. R.:

Software Metrics – A Guide to Planning, Analysis, and Application

CRC Press Company, Boca Raton, 2004 (286 pages)
ISBN 0-8493-1661-8

The book simplifies software measurement und explains its value as a pragmatic tool for management. Ideas and techniques presented in this book are derived from best practices. Some of the keywords are fundamentals of software measurement, metrics system architectures, regression models, exploring metrics for defect management, and strategic visions.

Conferences Addressing Metrics Issues

CSMR 2004:

8th European Conference on Software Maintenance and Reengineering
March 24-26, 2004, Tampere, Finland
see: **http://www.cs.tut.fi/csmr2004**

WWW 2004:

International World Wide Web Conference
May 17-22, 2004, New York
see: http://www2004.org/

ASM 2004:

Applications of Software Measurement

2004,
see: **http://www.sqe.com/asm/media.asp**

PE2004:

5. Workshop Software Performance Engineering
14. Mai 2004 in München,
see: http://ivs.cs.uni-magdeburg.de/~schmiete/peak/

Metrics 2004:

10th International Symposium on Software Metrics
September 14-16, 2004, Chicago
see: **http://swmetrics.org/**

CONQUEST 2004:

Conference on Quality Engineering in Software Technology
September 22-24, 2004, Nuremberg, Germany
see: http://www.conquest 2004.de

IWSM2004/Metrikon2004:

14th International Workshop on Software Measurement,
DASMA Metrik Kongress
November 3-5, in Berlin, Germany
see: http://iwsm2004.cs.uni-magdeburg.de

IFPUG 2004:

IFPUG 2004 Annual Conference
September 19-24 , 2004, San Diego
see: http://www.ifpug.org/conferences/

Conferences Addressing Metrics Issues

ISESE 2004:

IEEE International Symposium on Empirical Software Engineering
August 19-20, 2003, Redondo Beach, CA
see: `http://www.cs.umd.edu/~mvz/isese2004/`

UML 2004:

Fourth International Conference on the Unified Modelling Language
October 11-15, 2004, Lisbon, Portugal
see: `http://www.umlconference.org/`

EuroSPI 2004:

European Conference on Software Process Improvement
November 10 - 12, 2004, Trondheim, Norway
see: `http://www.eurospi.net/`

see also: **OOIS, ECOOP** and **ESEC** European Conference

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.aemes.fi.upm.es
  AEMES Association Espanola de Metricas del Software
- http://www.asqf.de
  ASQF Arbeitskreis Software-Qualität Franken e.V., Nuremberg, Germany
- http://www.cosmicon.com
  COSMIC Common Software Measurement International Consortium
- http://www.dasma.org
  DASMA Deutsche Anwendergruppe für SW Metrik und Aufwands-schätzung e.V.
- 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/

Metrics in the World-Wide Web

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/meastl1.htm**
  Maryland Metrics Tools

- **http://cutter.com/itgroup/reports/function.html**
  Function Point Tools by Carol Dekkers

Tool Vendors

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

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

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

Metrics in the World-Wide Web

- **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 8 *2003* *NUMBER 2*

CONTENTS

Announcement **3**

Workshop Report **7**

Position Papers **26**

New Books on Software Metrics **43**

Conferences Addressing Metrics Issues **45**

Metrics in the World-Wide Web **47**

ISSN 1431-8008