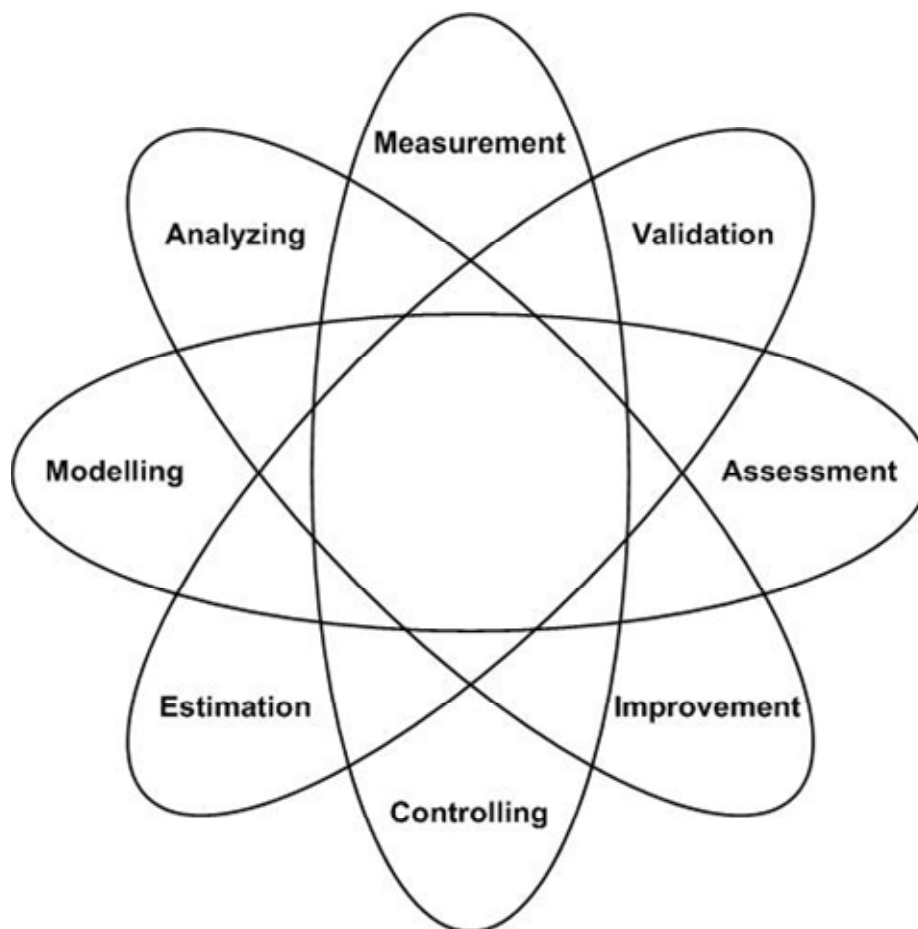


SOFTWARE MEASUREMENT NEWS

Journal of the Software Metrics Community



Editors:

Alain Abran, Günter Büren, Reiner Dumke, Christof Ebert, Cornelius Wille



The *SOFTWARE MEASUREMENT 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 Ouest,
Montréal, Quebec, H3C 1K3, Canada
Tel.: +1-514-396-8632, Fax: +1-514-396-8684
aabran@ele.etsmtl.ca*

Günter Büren

*Vice Chair of the DASMA
Büren & Partner Software-Design GbR
Thurn-und-Taxis-Str. 12, D-90411 Nürnberg, Germany
Tel.: +49-911-5195511, Fax: +49-911-5195555
gb@bup-nbg.de
<http://www.dasma.org>*

Reiner Dumke

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

Christof Ebert

*Dr.-Ing. in Computer Science
Vector Consulting GmbH
Ingersheimer Str. 24, D-70499 Stuttgart, Germany
Tel.: +49-711-80670-175
christof.ebert@vector-consulting.de*

Cornelius Wille

*Professor on Software Engineering
University of Applied Sciences Bingen
Berlinstr. 109, D-55411 Bingen am Rhein, Germany
Tel.: +49-6721-409-257, Fax: +49-6721-409-158
wille@fh-bingen.de*

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

Technical Editor: Dagmar Dörge

The journal is published in one volume per year consisting of two numbers. All rights reserved (including those of translation into foreign languages). No part of this issues may be reproduced in any form, by photoprint, microfilm or any other means, nor transmitted or translated into a machine language, without written permission from the publisher.

© 2011 by Otto-von-Guericke-University of Magdeburg. Printed in Germany

IWSM/MENSURA 2011

The Joint Conference of 21th International Workshop on Software Measurement (IWSM) and 6th International Conference on Software Process and Product Measurement (MENSURA)

November 3-4, 2011, Nara Prefectural New Public Hall, Nara, Japan

Co-located events: International Workshop on Empirical Software Engineering in Practice (November 1) and MSR School in Asia (November 2)



THEME & SCOPE

Software measurement and metrics are key technologies for managing and controlling software development projects. Measurement is essential for any engineering activity and for increasing scientific and technical knowledge regarding both the practice of software development and empirical research in software technology. This conference will facilitate the exchange of software measurement experiences between theory and practice.

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
- Functional size measurement
- Software process improvement
- Product and process metrics, e.g. for security
- Measurement-based best practice evaluation
- Measurement-based project management
- Measurement databases
- Metrics validation
- Measurement services and tools
- Measurement-based business IT alignment
- Theory of measurement
- Measurement paradigms
- Enterprise embedded solutions
- Measurement of agile development
- Measurement of software patterns
- Measurement of software services
- Measurement of service-oriented systems
- Measurement of cloud-based systems
- Software benchmarking
- Software effort estimation
- Software defect prediction
- Software cost models

SUBMISSIONS

The paper types are **full paper** (*maximum 10 pages*), **short paper** (*maximum 6 pages*), and **fast abstract** (*1-4 pages*). Short papers are appropriate for exhibiting late-breaking results and discussions about them. Fast abstracts are appropriate for work that is very new, in a prototype stage. All type papers should not have already been published elsewhere, nor should they have been submitted to a journal or to another conference. Papers which exceed page limitation will be automatically rejected.

Full papers and short papers should be submitted to

<http://www.easychair.org/conferences/?conf=mensura2011>

Fast abstracts should be submitted to mensura2011-papers@is.naist.jp

All types of papers must be formatted to IEEE Conference Publications Manuscript Formatting Guidelines (two-column format). Accepted full papers and short papers will be included in the conference proceedings volume I, **IEEE Computer Society Digital Library**, and **IEEE Xplore**. Accepted fast abstracts will be included in the conference proceedings volume II. At least one of the authors of each paper accepted must register for the conference and ensure paper presentation.

CONFERENCE TIMETABLE

Full papers/short papers

Submission deadline for papers: **June 9, 2011**
 Notification of acceptance: July 28, 2011
 Camera ready due: Aug. 12, 2011

Fast abstracts

Submission deadline for abstracts: **Aug. 28, 2011**
 Notification of acceptance: Sep. 16, 2011
 Camera ready due: Sep. 24, 2011

CONFERENCE ORGANIZATION

General Co-Chairs: Koichi Matsuda (Software Engineering Center, IPA, Japan), and Ken-ichi Matsumoto (Nara Institute of Science and Technology, Japan)

Organizing Chair: Masahide Nakamura (Kobe University, Japan)

Program Chair: Akito Monden (Nara Institute of Science and Technology, Japan)

Fast Abstract Co-Chairs: Yoshiki Mitani (Nara Institute of Science and Technology/MITANI A.R.I., LLC.), and Jens Heidrich (Fraunhofer IESE, Germany)

Steering Committee: Alain Abran (École de technologie supérieure University of Quebec, Canada), Reiner R. Dumke (Magdeburg University, Germany), and Juan J. Cuadrado-Gallego (University of Alcalá, Spain)

CONTACT

IWSM/MENSURA 2011 conference office
 Software Engineering Lab., Graduate School of Information
 Science, Nara Institute of Science and Technology
 8916-5 Takayama, Ikoma, Nara 630 0192, Japan
 e-mail: mensura2011@is.naist.jp

FURTHER INFORMATION

<http://mensura.wordpress.com/>



Deutschsprachige
Anwendergruppe für
Software-
Metrik und
Aufwandschätzung e.V.

MetriKon 2011

DASMA Software Metrik Kongress

16. - 18. November 2011

Fraunhofer IESE Kaiserslautern



GI-Fachgruppe 2.1.10
"Software Messung
und Bewertung"

Aufruf zur Einreichung von Beiträgen

Veranstalter

DASMA e.V. und
GI-Fachgruppe 2.1.10

Kontaktadressen für MetriKon-Beiträge:

metrikon-beitraege@dasma.org

Ausstellungsreservierungen:

DASMA e.V.
E-Mail: info@dasma.org

Programmkomitee

Manfred Bundschuh
DASMA
Günter Büren
Büren & Partner, Nürnberg
Dr. Axel Dold
DaimlerChrysler AG, Ulm
Prof. Dr. Reiner Dumke
Universität Magdeburg
Dr. Christof Ebert
Vector Consulting, Stuttgart
Bernd Gebhard
Bayerische Motoren Werke, München
Prof. Dr. Hans-Georg Hopf
GSO-Fachhochschule Nürnberg
Dr.-Ing. Marek Leszak
Alcatel-Lucent, Nürnberg
Prof. Dr. Claus Lewerentz
Technische Universität Cottbus
Prof. Dr. Peter Liggesmeyer
Fraunhofer IESE, Kaiserslautern
Dr.-Ing. Mathias Lothar
Robert Bosch GmbH, Stuttgart
Dr. Dirk Meyerhoff
Schüco-Service GmbH, Bielefeld
Dr. Jürgen Münch
Fraunhofer IESE, Kaiserslautern
Dr. Frances Paulisch
Siemens AG, München
Prof. Dr. Andreas Schmietendorf
Hochschule für Wirtschaft, Berlin
Harry Sneed
SES, München/Budapest
Prof. Dr. Cornelius Wille
FH Bingen
Prof. Dr.-Ing. Horst Zuse
Technische Universität Berlin

Themenstellung & Abgrenzung

Software-Messverfahren und Metriken sind Schlüsseltechnologien für Controlling und Management von Software-Entwicklungsprozessen und -Produkten und für quantitative Betrachtungen von Produktivität und Qualität. Messen ist eine wichtige Grundlage für die Gewinnung neuer Erkenntnisse in Wissenschaft und Technik. Es ist damit sowohl für die Praxis der Software-Entwicklung als auch für die empirische Forschung zur Software-Technik unverzichtbar. Für den notwendigen Erfahrungsaustausch zwischen Theorie und Praxis bietet die MetriKon eine ideale Plattform. Wir laden Sie herzlich ein, sich aktiv an diesem Austausch zu beteiligen.

MetriKon-Beiträge zur Software-Messung

Gewünscht werden vorwiegend (**Erfahrungs-**)**Berichte aus der industriellen Praxis** und der **praxisrelevanten Forschung**.

Parallel zu den Vorträgen werden **Tutorien oder Workshops** von ca 1,5 Std. Dauer zur Einführung oder Vertiefung komplexer Themen oder Verfahren der Software-Messung und Aufwandschätzung angeboten. Vorschläge dazu mit einer Kurzbeschreibung bitte bis zum 20. Juli 2011 einreichen.

Begleitend zur Tagung wird eine **Ausstellung** organisiert, in der Dienstleister und Werkzeughersteller den Teilnehmern ihre Angebote zur Unterstützung von Softwaremessung und Aufwandschätzung vorstellen. Aussteller sollten Ihren Teilnahmewunsch spätestens bis Ende August beim Veranstalter anmelden.

Themenfelder

Generell sind Beiträge zu allen Themen rund um Software-Metriken und Aufwandschätzverfahren erwünscht. Das Programmkomitee behält sich bei der Auswahl der eingereichten Beiträge vor, auf eine ausgewogene Mischung von Praxis und Theorie zu achten.

Zur Orientierung hier einige Stichworte zu erwünschten Vortragsthemen:

- Software-Metriken, Vergleich von Metriken, Einführung von Metriken etc.
- Zielorientierte Metriken zur Verfolgung und Erreichung von Projekt- und Verbesserungszielen
- Einsatz, Einführung und Erfahrungen mit Aufwandschätzverfahren und -Tools
- Quantitatives Projektmanagement, Projektcontrolling
- CMMI- / SPICE/ ITIL/ CobiT.-konforme Messprogramme und Metriken
- Verwandte Themen wie Benchmarking von Prozessen und Projekten
- Einsatz von Mess- und Aufwandschätzverfahren im Zusammenhang mit Embedded Systems, Cloud Computing, Testen, Anforderungsmanagement, etc.

Weitere Informationen

Zur DASMA e.V. und zur MetriKon 2011-Tagung finden Sie weitere Informationen im Web unter <http://dasma.org> und <http://www.metrikon.de>, zur GI-Fachgruppe 2.1.10 unter <http://ivs.cs.uni-magdeburg.de/sw-eng/us/giak>. Beachten Sie bei der Ausarbeitung von Beiträgen für die MetriKon 2011 bitte die „Richtlinien zur Einreichung von Beiträgen zur MetriKon 2011“. Sie finden diese Richtlinien und auch den Call for Papers im Internet auf den Seiten der MetriKon unter der oben angegebenen Adresse.

Wichtige Termine

22. Juni 2011
20. Juli 2011
bis 10. August 2011
21. September 2011

Abgabeschluss einer aussagekräftigen Kurzfassung
Abgabeschluss für Workshop-/Tutorialvorschläge
Benachrichtigung über die Annahme
Abgabe des druckfertigen Beitrages



*Ankündigung des 6. Workshops
„Bewertungsaspekte serviceorientierter Architekturen“*

BSOA2011

15. November 2011, Köln (Gastgeber: SQS AG Köln)

MOTIVATION

Serviceorientierten Architekturen scheint in jüngerer Zeit ein ähnliches Schicksal wie der Titanic zu widerfahren (vgl. Aussage von Anne Thomas Manes – Jan. 2009): Schon lange untergegangen, aber irgendwie spricht man doch immer wieder darüber. Es stellt sich die Frage nach den Ursachen und den heute benötigten Anforderungen/Veränderungen bei modernen Integrationsarchitekturen:

Ist der „Geist“ der SOA heute unbrauchbar?

Da eine SOA weder als Produkt, Lösung noch als Technologie verstanden werden kann, entpuppt sich eine allzu technologische Betrachtung häufig bereits als potentielle Ursache gescheiterter SOA-Ansätze.

Wenn SOA weiter gefasst wird, vielleicht schwimmt dieses Schiff doch noch?

Die BSOA-Initiative widmet sich diesen empirischen Betrachtungen und führt dazu jährlich einen entsprechenden Workshop durch. Der kommende Workshop wird sich insbesondere mit den folgenden Aspekten auseinandersetzen:

- Agile Vorgehensweise und SOA – ein Oxymoron!?
- Synchronisation zwischen fachlicher und technischer SOA
- SOA ready Services vs. Cloud-Lösungen
- Enterprise Architektur Management und Serviceorientierung
- SOA als Unwort – Bedürfnisse nach neuen Hypes
- Zielstellungen und Eigenschaften moderner Servicemarktplätze

Selbstverständlich geben die dargestellten Themen nur einen ausgewählten Teil möglicher Herausforderungen bei der Bewertung serviceorientierter Architekturen wieder. Dem entsprechend dienen diese der Orientierung und nicht der Einschränkung für potentielle Beiträge.

WORKSHOP-BEITRÄGE

Praktiker und Wissenschaftler, die auf dem Gebiet der Konzeption, Entwicklung und Management moderner Integrationsarchitekturen tätig sind, werden gebeten, Beiträge im doc- oder pdf-Format über die u.g. Webseite einzureichen. Der Umfang der Beiträge sollte 3000 Wörter nicht übersteigen. Die Formatierungsrichtlinien werden auf der unten genannten Webseite veröffentlicht. Angenommene Beiträge werden innerhalb eines 30-minütigen Vortrags präsentiert bzw. in Form eines Posters (innerhalb der Workshoppausen) vorgestellt. Alle angenommenen Beiträge des Workshops erscheinen in einem Tagungsband.

PROGRAMMKOMITEE

S. Aier,
Universität St. Gallen

E. Dimitrov,
T-Systems

T. Grawe,
Advicio

M. Irtmann,
IBM

G. Limböck,
SAP

R. Molle,
ITAB Hamburg

K. Patzer,
T-Systems Dresden

A. Schmietendorf,
HWR Berlin

F. Victor,
FH Köln

F. Balzer,
CA Deutschland

R. Dumke,
Uni Magdeburg

J. M. Gomez,
Uni Oldenburg

B. Klöppel,
T-Systems

M. Lothar,
Robert Bosch GmbH

S. Nakonz,
Bitnologie

H. Pundt,
HS Harz

J. Schuck,
MATERNA GmbH

C. Wille,
FH Bingen

J. vom Brocke,
HS Liechtenstein

M. Fiedler,
Software AG

W. Greis,
TPS Data & CECMG

S. Kusterski,
Toll Collect

M. Mevius,
FZI & HTWG Konstanz

S. Patig,
Universität Bern

M. Rothaut,
Deutsche Telekom

F. Simon,
SQS AG

R. Zarnekow,
TU Berlin

TERMINE

10.08.2011	Einreichung von Abstracts
18.09.2011	Einreichung von Beiträgen
04.10.2011	Annahme/Ablehnung
09.10.2011	finales Workshop-Programm
16.10.2011	Abgabe der druckreifen Beiträge
15.11.2011	Workshop in Köln

WEBSEITE ZUM WORKSHOP

Weitere Informationen: <http://www-ivs.cs.uni-magdeburg.de/~gi-bsoa/2011>

Paper Submission: <https://www.easychair.org/conferences/?conf=bsoa2011>

GASTGEBER

SQS Software Quality Systems AG
Stollwerckstrasse 11, D-51149 Köln
WWW: <http://www.sqs.de>



Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

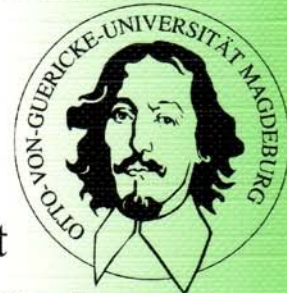
Our International Conference on Software Measurement (IWSM/MetriKon/Mensura 2010) took place in Stuttgart, Germany in November, 10-12, 2010. The following report gives an overview about the presented papers. Furthermore, the papers are published in Shaker-Verlag (ISBN 978-3-8322-9618-6):

Magdeburger Schriften zum Empirischen Software Engineering

Eds.: Prof. Dr. Alain Abran, Université du Québec
Günter Büren, Vorsitzender der DASMA e.V.
Prof. Dr.-Ing. habil. Reiner R. Dumke, Universität Magdeburg
Prof. Juan J. Cuadrado-Gallego, Universidad de Alcalá, Madrid
Dr. Jürgen Münch, Fraunhofer IESE, Kaiserslautern

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

Fakultät für Informatik
Institut für Verteilte Systeme
Arbeitsgruppe Softwaretechnik



Applied Software Measurement

Proceedings of the joined International Conferences
on Software Measurement

IWSM/MetriKon/Mensura 2010

10.-12. November 2010

Vector Consulting Services, Stuttgart, Germany



Deutschsprachige Anwendergruppe für
Software-Metrik und Aufwandsschätzung



GI-Fachgruppe 2.1.10
Software Messung und Bewertung



Otto-von-Guericke-Universität Magdeburg
Software Measurement Laboratory (SMLab)

SHAKER
VERLAG

Keynote I

Lean Development - Measuring and Improving Performance

Dr. Christof Ebert
Vector Consulting Services, Stuttgart

Abstract. To survive in a fast changing environment, we need to continuously improve productivity, reduce rework, and optimize product strategies. Lean development and lean management offer the right ingredients: Eliminating waste, empowering teams, delivering as fast as possible, seeing the whole. But there is a dark side, as recent industry experiences show. Too lean is mean. Lean often fails due to lack of vision, organizational misalignment and insufficient execution. One of the main sources of failure is trying to make a too big step in the ways of working at once. Therefore, the change to lean has to be done in a continuous and incremental way. This keynote will introduce to lean principles and practices in software organizations. It will draw upon experiences from a variety of industries with topics such as lean transition, introducing new tools, improving engineering performance, and setting up a global software organization. Specifically the keynote will highlight how to measure performance and thus to arrive at a more lean software process over time.

Model-based Product Quality Evaluation with Multi-Criteria Decision Analysis

Adam Trendowicz, Michael Kläs, Constanza Lampasona, Jürgen Münch
Fraunhofer Institute for Experimental Software Engineering
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{adam.trendowicz, michael.klaes, constanza.lampasona, juegen.muench}
@iese.fraunhofer.de

Christian Körner, Matthias Saft
Siemens AG, Corporate Technology
Otto-Hahn-Ring 6, 81739 Munich, Germany
{christian.koerner, matthias.saft}@siemens.com

Abstract. *The ability to develop or evolve software or software-based systems/services with defined and guaranteed quality in a predictable way is becoming increasingly important. Essential - though not exclusive - prerequisites for this are the ability to model the relevant quality properties appropriately and the capability to perform reliable quality evaluations. Existing approaches for integrated quality modeling and evaluation are typically either narrowly focused or too generic and have proprietary ways for modeling and evaluating quality. This article sketches an approach for modeling and evaluating quality properties in a uniform way, without losing the ability to build sufficiently detailed customized models for specific quality properties. The focus of this article is on the description of a multi-criteria aggregation mechanism that can be used for the evaluation. In addition, the underlying quality meta-model, an example application scenario, related work, initial application results, and an outlook on future research are presented.*

Keywords: *Software quality model, quality evaluation, product assessment, MCDA, Quamoco.*

Towards a Model-Centric Quality Assessment

Benoît Vanderose, Flora Kamseu, Naji Habra
University of Namur (Belgium)
bva, fka, nha@fundp.ac.be

Abstract. This paper introduces MoCQA, a Model-Centric Quality Assessment framework relying on a quality metamodel and supporting a flexible integration of different types of quantitative quality assessment all along the software development lifecycle. This framework provides a top-down two-level methodology. The first level (conceptual) focuses on the generation of a customised assessment quality model (CAQM) for the software project. The underlying quality metamodel ensures the robustness and coherence of the model generation and its evolution. The second level (operational) implements a concrete measurement plan using the generated CAQM for a systematic and/or semi-automated exploitation of the measurement values. It also allows different backtracking between those steps to support the CAQM “lifecycle”, which is one of the strengths of the framework. So far, early applications highlight the positive impact of the opportunity given by MoCQA to refine and let evolve the CAQM as well as the flexibility allowed by the quality metamodel.

Keywords: Metamodel, model-driven, quality models, continued assessment

Categorical Representation of Software Services Quality and Decision-Making Using Markov Decision Process

O.Ormandjieva, V.Mikhnovsky, E.Yurasovskaya, S.Klasa, J.Bentahar
Concordia University
{ormandj, kiricen, yurasovskaya, klasa, bentahar}@encs.concordia.ca

Abstract. The results described in this paper contribute to the research in the service management and QoS measurement through exploiting the applicability of category theory as a unifying formal language that allows using the same constructs for modeling heterogeneous objects such as service, decision-making, QoS metric values and different types of relations between them. Category Theory can provide abstract basic building blocks for the metamodeling by the categorical notions of object, morphism, functor, as well as natural transformation. Markov Decision Process is employed for establishing the level of confidence in the choice of the decision-making rules in a given context. Putting decision-making on a firm foundation can potentially play a very important role in service orchestration based on judgments on service quality.

Keywords: Quality of Service, Markov Decision Process, Category Theory, decision-making.

Werterhaltung von Software durch evolutionäre Qualitätssicherung

Harry M. Sneed

Fachhochschule Hagenberg, Oberösterreich
Harry.Sneed@T-Online.de

Zusammenfassung: Der folgende Beitrag überträgt die Theorie der Kapitalgütererhaltung des österreichischen Volkswirts Friedrich Hayek auf Software Güter und begründet damit den Aufwand für periodische Reengineering-Projekte. Wie andere Güter auch verliert das Gut Software an Wert wenn es auf sich allein gelassen wird. Die Funktionalität veraltet, die Komplexität steigt und die Qualität sinkt. Wenn Software seinen Wert als Kapitalgut erhalten sollte, dann muss sie ständig erneuert werden und zwar nicht nur funktional sondern auch technisch und qualitativ. Reengineering Projekte umfassen eine Reihe Restrukturierungs- bzw. Refaktorisierung-Maßnahmen, die dazu führen den Wert der Software zu steigern. Der Beitrag schlägt eine Metrik vor um den Grad der Wertsteigerung zu messen und zeigt wie dieser Wert benutzt werden kann die Kosten der Reengineering-Projekte mittels einer Rol Berechnung zu rechtfertigen. Eine Fallstudie aus der ITPraxis illustriert die Anwendung dieser Wirtschaftlichkeitsberechnung.

Schlüsselbegriffe: Software-Reengineering, Restrukturierung, Refaktorisierung, Softwaremetrik, Werterhaltung, Wirtschaftlichkeitsanalyse, Aufwandsschätzung, Rol, Value-driven Softwarewartung.

Entwicklung eines operationalisierten Qualitätsmodells für SAP-Integrationsprojekte

Markus Großmann, Mikolaj Plazewski

Capgemini, Loeffelstr. 46, 70597 Stuttgart
{markus.grossmann, mikolaj.plazewski}@capgemini.com

Zusammenfassung: Im Bereich der Entwicklung betrieblicher Informationssysteme kommen heutzutage vermehrt Kombinationen aus Individual- und Standardsoftware zum Einsatz. Im Rahmen des Forschungsprojekts Quamoco entwickelt Capgemini daher ein Qualitätsmodell für derartige Hybridsoftware im Bereich der ABAP-Entwicklung, welches charakteristische Qualitätseigenschaften beschreibt und Prüfungen dafür anbietet. Dieses Papier beschreibt Ergebnisse der ersten von drei Iterationen zur Entwicklung des Modells: zum einen konnte eine inhaltliche Basis an relevanten Prüfungen erarbeitet werden. Quellen dafür waren ein Qualitätsmodell für Individualsoftware, Richtlinien der SAP und Interviews mit ABAP-Entwicklungsexperten. Zum anderen wurde im Rahmen eines technischen Durchstichs zur Operationalisierung des Modells festgestellt, dass die konkrete Messung von Eigenschaften durch technische Barrieren des SAP-Systems stark erschwert wird. Hier konnten Problemfelder identifiziert und erste Workarounds gefunden werden.

Schlüsselbegriffe: Praktische Software-Messung, Qualitätsmodell, SAP-Integrationsprojekte

Qualitätssicherung in der Agilen Software-Entwicklung

André Janus, Reiner R. Dumke
André Janus - IT Consulting, Karlsruhe
Otto-von-Guericke-Universität Magdeburg
mail@andre-janus.de
dumke@ivs.cs.uni-magdeburg.de

Zusammenfassung: Qualitätssicherung nimmt in der Agilen Software-Entwicklung auf den ersten Blick keine große Rolle ein. Dennoch ist Qualität ein grundlegendes Prinzip von Agilen Entwicklungsmethoden, das implizit von vielen Agilen Praktiken umgesetzt wird. Dieser Beitrag stellt die Grundlagen Agiler Software-Entwicklung dar und gibt einen Überblick über zentrale Agile Praktiken. Es wird der Beitrag Agiler Software-Entwicklung und Agiler Praktiken zu verschiedenen Aspekten der Qualitätssicherung untersucht. Die theoretischen Betrachtungen werden dabei durch Umfrageergebnisse aus Agil durchgeführten Projekten der T-Systems SI ergänzt. Schließlich wird das Agile Maturity Model Integration (AMMI) als Reifegradmodell für Agile Software-Entwicklungsmethoden und -prozesse vorgestellt. Außerdem werden die damit zusammenhängenden offenen Fragen und Herausforderungen für die Forschung beleuchtet, die im Ergebnis eine Bewertung von Agilen Software-Entwicklungsmethoden und -prozessen ermöglichen.

Schlüsselbegriffe: Empirische Fallstudien, Produkt- und Prozessmetriken, Best Practices, Agilität, Qualitätssicherung

Specification and Measurement Derived from System Operations Non Functional Requirements

Alain Abran, Khalid T. Al-Sarayreh
Software Engineering Department, University of Quebec (ETS)
1100 Notre-Dame West Montréal, Québec H3W 1T8, Canada
alain.abran@etsmtl.ca, khalid.al-sarayreh.1@ens.etsmtl.ca

Abstract. An operation is typically described initially as a non functional requirement at the system level. Systems engineers must subsequently apportion these system requirements very carefully as either software or hardware requirements to conform to the operations requirements of the system. A number of concepts are provided in the ECSS and IEEE standards to describe the various types of candidate operations requirements at the system, software, and hardware levels. This paper organizes these concepts into a generic standards-based reference model of the requirements at the software level for system operations. The structure of this reference model is based on the generic model of software requirements proposed in the COSMIC – ISO 19761 model, thereby allowing the measurement of the functional size of such operations requirements implemented through software.

Keywords: Operations Requirements, Non functional requirements – NFR, Functional size, COSMIC – ISO 19761, ECSS International Standards, IEEE-830 standard, Software Operations Measurement

Design Metrics for UML Models

Harry M. Sneed
University of Regensburg
Harry.Sneed@T-Online.de

Abstract. This experience report describes the metrics that were used to measure and evaluate the UML model for a web application development. The metrics were divided into four classes – quantity metrics, complexity metrics, quality metrics and size metrics. The quantity metrics are simple counts of the design entities and their relationships. The complexity metrics measure the relation between the entities, i.e. corners, and their relationships, i.e. edges. The quality metrics measure the relation between the desired and actual characteristics of the system architecture. The size metrics transform the basic quantities into overall size measures which can be used to estimate the costs of development and maintenance. The paper summarizes the types of entities and relationships used in this particular UML model and shows how they were measured using a UML analysis tool. At the end, the measurement results are discussed.

Keywords: UML Models, Metrics, System Design, Design Measurement, System Size, Quality, Complexity,

Specification and Measurement of System Configuration Non Functional Requirements

Khalid T. Al-Sarayreh, Alain Abran
Software Engineering Department, University of Quebec (ETS)
1100 Notre-Dame West Montréal, Québec H3W 1T8, Canada
khalid.al-sarayreh.1@ens.etsmtl.ca, alain.abran@etsmtl.ca

Abstract. The European ECSS-E-40 series of standards for the aerospace industry includes configuration as one of 16 types of non functional requirements (NFR) for embedded and real-time software. Configuration requirements are typically described at the system level as non functional requirements, and a number of concepts and terms are provided in the ECSS series to describe various types of candidate configurations. This paper collects and organizes these configuration related descriptions into a generic model for the specification of software functional user requirements (software FUR) for system configuration NFR, and for measuring their functional size for estimation purposes using the COSMIC ISO 19761 standard.

Keywords: Configuration Requirements, Non functional requirements – NFR, Functional size, COSMIC – ISO 19761, ECSS International standards.

Integration von Prozess-, Modell- und Produktmetriken

Bernhard Peischl

K-net Softnet Austria
Institut für Softwaretechnologie
Technische Universität Graz
peischl@ist.tuGraz.at

Zusammenfassung: Dieser Beitrag beschäftigt sich mit der Integration von Prozess-, Modell- und Produktmetriken. Im Mittelpunkt der Betrachtung steht ein Software Cockpit zur Qualitätssicherung und das zugrundeliegende Datenbankschema zur Integration dieser Metriken. Der Artikel illustriert kurz die Ausgangsbasis – eine Reihe von Innovationsprojekten des K-net Softnet Austria – und stellt den Lösungsansatz zur Integration der Metriken vor. Um die Einstiegshürde in die modell-gestützte Qualitätssicherung vor allem aus praktischer Sicht zu vermindern, wird die Umsetzung eines leichtgewichtigen modellgetriebenen End-to-End Tests für graphische Benutzerschnittstellen bereitgestellt. Die Möglichkeit beliebige (und somit domänenspezifische) Metriken auf UML Modellen zu definieren und zu operationalisieren ist eine Voraussetzung um Modellmetriken in der Praxis zu etablieren. Weiter beleuchtet der Artikel einige praktische Erfahrungen mit Metriken im Allgemeinen.

Schlüsselbegriffe: Integration von Metriken, Modellmetriken, GUI End-to-End Testing, Software Cockpit, Operationalisierung von Metriken, modell-getriebener Test, kennzahlgestützte kontinuierliche Qualitätssicherung und Überwachung

Evidenzbasierte Klassifikation und Bewertung von Aufwandschätzmethoden

Cornelius Wille¹, W. Kuhl², Reiner Dumke³, Martin Kunz⁴

¹Fachhochschule Bingen, wille@fh-bingen.de

²EADS Friedrichshafen, wolfgang.kuhl@eads.com

³Universität Magdeburg, dumke@ivs.cs.uni-magdeburg.de

⁴G+J Electronic Media Service, Hamburg, kunz.martin@guj.de

Zusammenfassung: Bei der Auswahl geeigneter Aufwandschätzmethoden gibt es je nach Ausgangssituation unterschiedliche Fragestellungen. Das Hauptziel besteht natürlich in der bestmöglichen und genauesten Schätzung des Aufwandes für den eigenen IT-Bereich in Verbindung mit geringsten Kosten für die Schätzung selbst. Andererseits sind auch Fragen der Vergleichbarkeit im eigenen Unternehmen oder über (international verteilte) Firmenbereiche hinweg relevant. Des Weiteren kann ein Ziel auch darin bestehen, sich mit Wettbewerbern zu vergleichen bzw. den jeweiligen Aufwand auch nach außen hin transparent zu machen. Der vorliegende Beitrag beschäftigt sich mit diesen Fragestellungen und untersucht dabei die verschiedenen Aufwandschätzmethoden nach ihrer Bedeutung bzw. „Evidenz“ sowie die sich daraus ergebenden Auswahlkriterien für den eigenen Anwendungsbereich.

Schlüsselworte: Aufwandschätzung, Evidenz, Methodenklassifikation, Methodeneignung

Kostenmetriken für virtuelle Rechnerressourcen

Sebastian Breß¹, Robert Neumann², Reiner Dumke³

Otto-von-Guericke-Universität Magdeburg

1)Universität Magdeburg, sebastian.bress@st.ovgu.de,

2)Universität Magdeburg, robert.neumann@ovgu.de,

3)Universität Magdeburg, dumke@ivs.cs.uni-magdeburg.de

Zusammenfassung: Wie können Konzepte des Cloud Computing auf virtuelle Rechnerressourcen angewendet werden? Es wird gezeigt, dass dies mit einigen Einschränkungen und Kompromissen möglich ist. Allerdings ist bei virtuellen Maschinen eine OnDemand-Anpassung etwas "langsamer", da zur Änderung der Hardwareressourcen ein Neustart der betroffenen virtuellen Maschinen erforderlich wird. Diese Unfähigkeit zur Laufzeit OnDemand-Ressourcen zuzuteilen bzw. zu entziehen, kann als großes Handicap angesehen werden. Jedoch kann mit gewissen Einschränkungen eine OnDemand-Nutzung und Konfiguration von virtuellen Maschinen mit einem Pay-as-you-go-Bezahlmodell umgesetzt werden. Hierfür wird eine Kostenmetrik benötigt, die die Hardwarekonfiguration und die Nutzung der (virtualisierten) Hardware auf Kosten abbildet. Eine solche Metrik soll in dieser Arbeit konstruiert werden. Es werden nacheinander Kostenfunktionen für CPU-, Hauptspeicher- und Festplattennutzung diskutiert und die für die Pay-as-you-go-Abrechnung am besten geeignete ausgewählt, um sie in die zu erstellende Kostenmetrik einfließen zu lassen.

Schlüsselbegriffe: Cloud Computing, virtuelle Maschine, Kostenmetrik, OnDemand-Nutzung, Payas-you-go-Bezahlmodell

Which COSMIC Base Functional Components are Significant in Estimating Web Application Development? - A Case Study

*Luigi Buglione¹, Filomena Ferrucci², Cigdem Gencel², Carmine Gravino²,
Federica Sarro²*

¹École de Technologie Supérieure (ETS) - Montréal (Canada);
Engineering.IT – Rome (Italy)

²University of Salerno (Italy)

³Blekinge Institute of Technology, Karlskrona (Sweden)

luigi.buglione@eng.it, {fferrucci, gravino, fsarro}@unisa.it, cigdem.gencel@bth.se

Abstract. Estimation is still a challenging process for planning and managing software projects. Often, estimates are being done on experiential or analogous basis or using effort estimation models. Mostly, these approaches take software size (e.g., Lines of Code, Function Points, Object points) and other cost factors as the main inputs to estimation. This study focuses on functional size based effort estimation for Web application development and investigates the significance of the functional sizes of each of the COSMIC Base Functional Component (BFC) types in explaining the variation in the development effort. A case study was conducted collecting data on 25 Web projects from a software organization. The results show that the size of only one of the BFC Types can explain the variation in the effort nearly as good as the total functional size.

Keywords. Functional Size Measurement, COSMIC, Effort Estimation, Base Functional Component Types, Web Applications, Regression Analysis, Case Study.

Evaluation of Reliability Improvements for COSMIC Size Measurement Results

Erdir Ungan, Özden Özcan Top, Barış Özkan, Onur Demirörs
Department of Information Systems, Informatics Institute,
Middle East Technical University
(erdir, demirors, ozden, bozkan)@ii.metu.edu.tr

Abstract. Functional Size has been favored as a software size attribute that is independent from development method and implementation language and that can be measured in the early project phases. Functional Size Measurement methods are mostly criticized by the variance in measurement results for the same software from different measurers. In this paper, we present the results of an experiment performed as the continuation of a previous experiment. In the first experiment, we identified the common causes of variance and possible means to eliminate those causes. In this study, we evaluate the effect of the means and actions taken to improve the reliability of the measurement results and validate our claims in the first experiment.

Keywords: COSMIC, Functional Size Measurement, Function Points, Estimation, Reliability.

Software Engineering meets Cloud Computing - How Design for Six Sigma can help

Florian Muhß[#], Andreas Schmietendorf⁺

[#] Otto-von-Guericke Universität Magdeburg, FIN-IVS, Department of Distributed Systems,
39106

Magdeburg, Email: florian.muhs@st.ovgu.de

⁺ HWR Berlin – Berlin School of Economics and Law, FB II – Business Information Systems,
Neue

Bahnhofstr. 11-17, 10245 Berlin, Email: andreas.schmietendorf@hwr-berlin.de

Abstract. Developing software solutions using services and software from the cloud requires a flexible engineering process to reach shorter implementation times. To meet cost reduction goals and improve the overall quality, the development has to focus increasingly on the reuse and configuration of available software functionalities that are tried and tested. This leads to a new way of software engineering, a process that is incorporating existing software (e.g. web services) and functionalities that need to be programmed. Therefore, the requirements for project and quality management are much more challenging than before. Within this paper the possibilities of the "Design for Six Sigma" (DFSS) approach will be investigated. DFSS provides a way how this everchanging development process can be managed along five specific steps.

Key Words: Six Sigma techniques, Process improvement, Measuring processes and tools, Cloud Computing, Software Development

Aligning Software-related Strategies in Multi-Organizational Settings

*Martin Kowalczyk^A, Jürgen Münch^A, Masafumi Katahira^B,
Tatsuya Kaneko^B, Yuko Miyamoto^B, Yumi Koishi^B*

^AFraunhofer IESE, ^BJapan Aerospace Exploration Agency (JAXA)
{Martin.Kowalczyk, Juergen.Muench}@iese.fraunhofer.de
{Katahira.Masafumi, Kaneko.Tatsuya, Miyamoto.Yuko, Koishi.Yumi}@jaxa.jp

Abstract. Aligning the activities of an organization with its business goals is a challenging task that is critical for success. Alignment in a multi-organizational setting requires the integration of different internal or external organizational units. The anticipated benefits of multi-organizational alignment consist of clarified contributions and increased transparency of the involved organizational units. The GQM+Strategies approach provides mechanisms for explicitly linking goals and strategies within an organization and is based on goal-oriented measurement. This paper presents the process and first-hand experience of applying GQM+Strategies in a multi-organizational setting from the aerospace industry. Additionally, the resulting GQM+Strategies grid is sketched and selected parts are discussed. Finally, the results are reflected on and an overview of future work is given.

Keywords: Measurement-based business IT alignment, multi-organizational goal alignment, GQM+Strategies deployment, strategic measurement systems

Keynote II The Art of Measurement in Agile Development

Dr. Pekka Abrahamsson
University of Helsinki.

Abstract. Agile software development methods emerged over a decade ago. Many skeptics believed that these methods will fade away similar to many other software engineering innovations over the past 30 years. After ten years of application of agile methods in various industrial settings, it has become apparent that agile methods are being applied in large and small organizations alike almost context independently. While standardization of agile methods has proven to be a challenge it has not hindered the growing adoption rate. This talk will take the measurement viewpoint to agile development, present some concrete numbers arising from agile teams and products, and argue that the transparency inherently built-in to agile system enables new types of measures to improve the ability to coordinate development endeavors.

Functional Size Measurement with Business Process Models: the Business Application Domain

Carlos Monsalve^{1, 2}, Alain Abran², Alain April²

¹CIDIS – FIEC – Escuela Superior Politécnica del Litoral

²GELOG – École de Technologie Supérieure

monsalve@espol.edu.ec, {alain.abran, alain.april}@etsmtl.ca

Abstract. Functional size measurement (FSM) is an important component of a software project, as it provides information for estimating the effort required to develop the measured software. For planning purposes, FSM should be performed during the early stages of the project. Considering that a common use of business process models is to gather requirements from the early stages of the project, this raises the opportunity that a business process model could be a valuable source of information for FSM. This paper analyzes the feasibility of such an approach for the business application domain, including the derivation of the modeling rules for measurement purposes and the rules to map the modeling constructs of a business process modeling language into the COSMIC FSM method concepts. The results are compared with those of a previous FSM case study.

Keywords: Functional size measurement, FSM, COSMIC, ISO 19761, business process model

A Review and Fundamental Issues of Lines of Code Metrics

Capers Jones¹, P. Rodriguez-Soria², J.J. Cuadrado-Gallego³

¹Capers Jones & Associates LLC

²Computer Science Department, University Alcalá, 28805 Madrid, Spain

³École de Technologie Supérieure - ETS, Notre-Dame Quest, Montréal, Canada

CJonesiii@cs.com, pablo.rsoria@uah.es, iicg@uahl.es

Abstract. The oldest metric for software projects is that of "lines of code" (LOC). This metric was first introduced circa 1960 and was used for economic, productivity and quality studies. The economics of software applications were measured using "dollars per LOC". Productivity was measured in terms of "lines of code per time unit". Quality was measured in terms of "defects per KLOC" where "K" was the symbol for 1000 lines of code. The LOC metric was reasonably effective for all three purposes. It is interesting to consider the history of lines of code (LOC) metrics and some of the problems with LOC metrics that led IBM to develop function points metrics. Therefore we will try to describe a brief history of LOC metric from its origins in 1960 through today, with projections to the next decade.

Keywords: Metrics, software measurement, Lines of Code, function points

On Building a Better Program Size Measure

Akito Monden¹, Shinji Uchida², Ken-ichi Matsumoto¹

¹Graduate School of Information Science, Nara Institute of Science and Technology
{akito-m, matumoto}@is.naist.jp

²Department of Information Engineering, Nara National College of Technology
uchida@info.nara-k.ac.jp

Abstract. *Source Lines of Code (SLOC) is a most basic and widely-used program size measure in software project management and/or quality assurance although it greatly depends on a programmer who implemented the program. To build a better (i.e. programmer-independent) program size measure, this paper analyzed 9 independently-built C programs of a same functional specification, and found that 3 base measures (the number of tokens, tokens of code clones, and function parameters) are useful to eliminate programmer-dependent aspects of SLOC. A new size measure called Adjusted Length of Code (ALOC) built upon these 3 base measures showed that variations of size in ALOC was at most 1.22 times difference among 9 programs while SLOC showed 3.16 times difference. Furthermore, ALOC showed at most 1.60 times difference among another 6 independently-built programs of an alternative specification while SLOC showed 4.66 times difference among these programs. These results suggest that the new measure ALOC can reduce the programmer-dependent aspects of program size and can be used as a better size measure in project management.*

Keywords: *Product Metrics, Source Lines of Code, Program Analysis*

Guideline for Sizing Agile Projects with COSMIC

Sylvie Trudel¹ and Luigi Buglione^{2,3}

¹Pyxis Technologies Inc. – Montréal (Canada), ²École de Technologie Supérieure (ETS)
- Montréal (Canada); ³Engineering.IT – Rome (Italy)
strudel@pyxis-tech.com, luigi.buglione@eng.it

Abstract. Agile became one of the most used ‘buzzwords’ in ICT projects in these recent years but the growing diffusion and interest of its related methods and techniques was not accompanied by the same maturity in sizing and estimation practices. In particular the application of a functional size measurement (FSM) method for sizing requirements is not typical to “agilists”, preferring to use mostly estimations based on experience and analogy. In such way, one of several drawbacks is a reduced data gathering from projects, not allowing to do (even at a basic level) statistical analysis, for better estimating the proper efforts value for next user story and – as a whole – of the project. This paper describes the reasons why a guideline for sizing Agile projects with the COSMIC method was required, along with a summary of the resulting guideline. Several agile methods are briefly described, more specifically their expected requirements format and their measurement practices.

Keywords: Functional Size Measurement, Agile Methods, Software Process Improvement, Software Project Estimation, Practical Measurement Application, COSMIC.

COSMIC-based Project Management in Agile Software Development and Mapping onto related CMMI-DEV Process Areas

Enrico Berardi¹, Luca Santillo²

¹Tecnologie nelle Reti e nei Sistemi, ²Agile Metrics
enrico.berardi@trs.it, luca.santillo@gmail.com

Abstract. Functional Size Measurement (FSM) methods have been introduced in the industry since late 70's and have been successfully adopted in software development projects for sizing the amount of functional requirements, often for software estimation purposes. Typically, measurements with FSM are performed in two moments of the project – early design and delivery, respectively supporting project estimation and validation. This approach 'works' for traditional waterfall cycles, typically requires adjustments in real world requirements changes over the projects, but is simply ineffective in modern agile development methodologies, where Evolutionary Requirements Analysis is the rule and the user (providing requirements and feedbacks) is constantly involved in the development process. This work depicts the adoption of the new generation FSM COSMIC method in order to address the agile development process with a standard measurement approach. Such approach results in fine-tuned measures all over the project cycle for agile projects – with a single metrics that holds valid from early design to end phases, while adequately covering all the intermediate iterations and refinements. Project management practices, such as Earned Value assessment, as well as general process improvement practices, such as the CMMI-DEV framework, are easily mapped onto COSMIC measurement and benefit from the proposed approach.

Keywords: Agile, measurement, COSMIC, project management, Capability Maturity Model

From Story Points to COSMIC Function Points in Agile Software Development – A Six Sigma perspective

Thomas Fehlmann¹, Luca Santillo²

¹Euro Project Office AG (Switzerland), ²Agile Metrics (Italy)
thomas.fehlmann@e-p-o.com, luca.santillo@gmail.com

Abstract. The spreading of agile methodologies in software development raises the question of how to measure requirements once more, as it happened in 'traditional' software industry development approaches decades ago. The difference is that requirements are not known in advance but detected as User Stories while iterating and enhancing the software product from one agile 'Sprint' to the other. Some authors – promoting best practices for agile software development – propose Story Points to size User Stories (e.g. Scrum, with Story Cards), yet not combined with base project estimation. Story Points are not standardized, thus leading to eventual misconceptions and quantitative differences among practitioners and domains. The uncertainty implied in such approach can therefore propagate to any estimate based on it, not to mention the difficulty in accurately tracing requirements and their variation over the project and across project iterations. This work investigates benefits from adopting a standardized Functional Size Measurement (FSM) method, such as COSMIC Function Points, in place of Story Points. Using a Transfer Function (from the Six Sigma practice) that transforms size into effort spent within a particular agile team, defect density prediction can be made using sensitivity analysis.

Keywords: Agile software development, Story Points, COSMIC Function Points, Six Sigma, Transfer Function.

Suggestions for Improving Measurement Plans: a BMP application in Italy

Luigi Buglione^{1,2}, *Luigi Lavazza*^{3,4}

¹ École de Technologie Supérieure (ETS) - Montréal (Canada); ² Engineering.IT – Rome (Italy); ³ University of Insubria, Computer Science and Communication Dept., Varese (Italy); ⁴ CEFRIEL, Milan (Italy)
luigi.buglione@eng.it, luigi.lavazza@uninsubria.it

Abstract. Time and Cost are most often in industry the two main (often solely) dimensions of analysis against which a project is monitored and controlled, excluding other possible dimensions such as Quality, Risks, impact on society and Stakeholders' viewpoints in a broader sense. Another issue of interest is the proper amount of measures and indicators to implement in an organization to optimizing the tradeoff between the cost of quality and the cost of non quality. How can multiple concurrent control mechanisms across several dimensions of analysis be balanced? The approach of Balancing Multiple Perspectives (BMP) has been designed to help project managers choose a set of project indicators from several concurrent viewpoints. After gathering experiences from Canada, Germany, Turkey and Spain, this paper presents the results from a new BMP application in Italy, using a list of 14 candidate measures interviewing a double set of respondents from academy. Lessons learned are presented, considering the impact that knowledge from universities newbies can bring into ICT organizations.

Keywords: Balancing Multiple Perspectives (BMP), Balanced Scorecard (BSC), Stakeholders, Measurement Plan, Cost of Measurement, (Practical Software & Systems Measurement) PSM.

Aktives Risikomanagement mit Escrow-Services

Dr. Frank Simon, Sven Euteneuer

SQS Software Quality Systems AG, Köln, Deutschland
(Frank.Simon|Sven.Euteneuer@sqz.de)

Motivation: Im Kontext zunehmender Software-Industrialisierung werden immer weiter reichende Teile der Software-Erstellung und des Software-Betriebs ausgelagert. Hierbei übernehmen Zulieferer wohldefinierte Aufgaben mit wohldefinierten Qualitätsanforderungen. Diese Aufgabendelegation erfolgt immer häufiger als Black-Box, d.h. der gesamte Prozess der Leistungserbringung ist meist weniger vorgegeben als das konkrete abzuliefernde Ergebnis.

Während diese Auslagerung in der Vergangenheit nur in Einzelfällen unternommen wurde, hat sich das Bild in der vergangenen Dekade drastisch gewandelt. Nicht nur, dass nunmehr auch vermeintlich unlukrative Nischen durch OTS-Software (Off-the-Shelf-Software) besetzt werden, die mindestens aus einem standardisierten Kern und an konkrete Anforderungen angepasster Hülle besteht. Vielmehr wird mittlerweile aufgrund erhoffter Kostenvorteile auch die Erstellung ganzer Individualsysteme mittels *offshore outsourcing* als Paket an große asiatische IT-Dienstleister vergeben: ein Markt der trotz Rezession des letzten Jahres im Bereich etwa um 25 Milliarden Euro lag (1).

Für die Software-Erstellung bedeutet diese strikte Aufgabentrennung häufig die präzise und allumfassende Formulierung von Anforderungen auf Seiten des Auftraggebers (evtl. verteilt auf mehrere Iterationen), die der Zulieferer in Form eines zu liefernden IT-Systems anschließend erfüllt. Zulieferer werden in dieser Rolle dann als „Komplettanbieter“ oder „One-Stop-Shop“ bezeichnet (2). Sie beschränken sich dann aufgrund dieser sehr ergebnisorientierten Sichtweise in der Regel darauf, nur das fertige System zu übergeben, da dies schließlich auch Vertragsgegenstand ist. Weiterführende Software-Aspekte wie Source-Code, Dokumentation, Tests, Architekturen oder auch Review-Ergebnisse bleiben in solchen Fällen ausschließlich im Besitz des Zulieferers, der sich damit häufig auch implizit die Übernahme weiterer Wartungsaufträge sichert.

Für den Auftraggeber ist diese Übertragung von Verantwortlichkeit vordergründig ebenfalls wünschenswert: Auf diese Art und Weise kann der Dienstleister als „Black Box“ verstanden werden; eine Sichtweise, die aufgrund des allgemeinen Trends der Business Fokussierung noch an Attraktivität gewinnt. Außerdem könnte aufgrund der Business Fokussierung der Auftraggeber häufig mit über das Executable hinausgehenden Artefakten schlicht nichts anfangen, würde also für als „wertlos“ angesehene Liefergegenstände bezahlen oder zumindest deren Zulieferung steuern müssen.

Embedded Software Memory Size Estimation Using COSMIC: A Case Study

Sophie Stern¹, Cigdem Gencel²,

¹ RENAULT, Embedded software group, Guyancourt Cedex (France)

² Blekinge Institute of Technology (BTH), Karlskrona (Sweden)
sophie.stern@renault.com, cigdem.gencel@bth.se

Abstract. Early and accurate memory size estimation of Electronic Control Units (ECUs), which are dedicated embedded systems providing major software functionalities, is essential for automotive industry. Usually, experts using their expert judgments perform the estimations. Software functional size measures, which attempt at measuring the size by capturing the amount of functionality, have become popular in software industry particularly for effort estimation and project management purposes due to their availability early in the project lifecycle. This study investigates the potential usage of COSMIC Function Points as a predictor of memory size in comparison to expert opinion methods through conducting a case study in Renault.

Keywords: Memory size estimation, Embedded systems, Functional Size Measurement, COSMIC Function Points, Case Study.

An Experience-Based Repository of Reusable Components for an Component-Based Automotive Software System

*Martin Hobelsberger¹, Reiner Dumke², Jürgen Mottok¹, Michael Niemetz³,
Gerhard Wirrer³*

¹University of Applied Sciences Regensburg, LaS3 - Laboratory for Safe and Secure Systems

fmartin.hobelsberger,juergen.mottokg@hs-regensburg.de

²Otto-von-Guericke-University Magdeburg,

Department of Computer Science Institute for Distributed Systems

dumke@ivs.cs.uni-magdeburg.de

³Continental Automotive GmbH Regensburg, Engine Systems
fmichael.niemetz,gerhard.wirrer@continental-corporation.com

Abstract. The automotive industry is facing the challenge of a rapid increase in software functions and software complexity of their applications. Customers demand high quality at low cost, while product development cycles have been shortened considerably. One cornerstone to master these challenges is efficient reuse of proven and well tested components across different projects. Therefore, a component-based software architecture was introduced to the automotive powertrain system. In this paper we introduce a new measurement approach based on network analysis, which allows to classify the components and to optimize their clustering into component groups for increasing reuse efficiency in terms of reuse scope and development effort. A repository of reusable components, using this technique supports the design of new architectures and will lead to a higher quality of the system design.

Keywords: measurement techniques, quality assurance, reuse metrics, component-based automotive systems, network analysis

Automation of Functional Size Measurement of the Model-based Requirements of Real-time Embedded Software

Hassan Soubra^{1, 2, 3}, Sophie Stern¹, Alain Abran², Amar Ramdan-Cherif³

¹Renault SAS, Direction de l'Electronique Avancée, France;

²GÉLOG – Laboratoire de recherche en génie logiciel – ETS, Canada;

³LISV – Laboratoire d'Ingénierie des Systèmes de Versailles – UVSQ, France;
Hassan.soubra@renault.com

Abstract. Functional size measurement (FSM) has become an important task in software development projects for Information Systems and, recently, for real-time embedded systems. Functional size can be used for a number of purposes: to estimate development effort, to manage project scope changes, to measure productivity, to benchmark, and to normalize quality and maintenance ratios. More and more industries have become interested in FSM, and Renault SAS¹, the industrial partner of the research project, has adopted the COSMIC FSM standard, ISO 1976, to measure the model-based requirements of its Real-time embedded software. However, FSM is a time-consuming process if applied manually, and the organizations that have adopted it are interested in automating the measurement of software requirements. This paper discusses some of the issues tackled in connection with the automation of COSMIC FSM in an industrial context, such as at Renault, where modeling

tools are used extensively to document software functional requirements. This paper presents, in particular, overviews of FSM procedures, a common semi-formal representation for the various modeling tools in use at this industry site, a measurement automation prototype, and an assessment protocol for FSM to evaluate the output of the automation tool.

Keywords: Functional size measurement, real-time systems, estimation, measurement procedure, automation tool, COSMIC ISO 19761

How to Evaluate Meta-Models for Software Quality?

Michael Kläs, Constanza Lampasona, Sabine Nunnenmacher

Fraunhofer Institute for Experimental Software Engineering

Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

{michael.klaes, constanza.lampasona, sabine.nunnenmacher}@iese.fraunhofer.de

Stefan Wagner, Markus Herrmannsdörfer, Klaus Lochmann

Institut für Informatik, Technische Universität München

Boltzmannstr. 3, 85748 Garching, Germany

{wagnerst, herrmama, lochmann}@in.tum.de

Abstract. The use of appropriate software quality models is crucial for companies to achieve the product quality required to satisfy customer needs. Most current quality models provide little operationalization and lack adaptation guidelines, which limits their usefulness in practice. It has been proposed to use meta-models to specify an explicit structure in order to ensure that quality models conforming to it can be operationalized and adapted by requiring corresponding model elements and modeling constructs. To be applicable in practice, a meta-model needs to be general enough so that existing quality models can be transferred to the new structure provided by the meta-model while preserving the knowledge they contain. This paper presents an empirical approach for evaluating generality as well as its application to a selected meta-model and six industrial quality models. The results show that (1) the proposed meta-model is general enough to model most contents of the industrial quality models, (2) the generality of a meta-model contributes to its perceived ease of use and usefulness, and (3) the evaluation approach is applicable and reflects the perception of quality model experts well.

Keywords: Software quality assurance, quality model, generality, Quamoco, empirical study

On the validity of benchmarking for evaluating code quality

Harald Gruber, Reinhold Plösch

Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria
{harald.gruber, reinhold.ploesch}@jku.at

Matthias Saft

Corporate Technology – SE 1, Siemens AG, Otto-Hahn-Ring 6, Munich, Germany
matthias.saft@siemens.com

Abstract. Evaluating source code quality is a laborious task when performed by experts. There exist a number of approaches that try to provide an automatic assessment. Absolute quality assessment methods (e.g. based on thresholds) did not yet prove the significance of the results. Software benchmarking is a relative assessment approach based on the general idea of benchmarking from other industries. We developed a benchmarking-oriented code quality assessment method that at least overcomes known technical problems of other benchmark based methods. Nevertheless, the major concern is to validate the significance of benchmarking based results by comparing these results with the quality assessments of experts. For this purpose we conducted two studies (one for Java projects, the other for C# projects) involving a total of 10 open source projects and 22 experts. While the first experiment with Java projects has provided a result that has motivated us to use the benchmarking-oriented assessment more intensively, the experiment with C# showed us that we cannot trust the results of the automatic benchmark assessment blindly.

Keywords: Source code quality, automatic assessment, benchmarking, static code analysis, experimental study

Assessment of the Quality of Functional user requirements documentation using criteria derived from measurement with COSMIC – ISO 19761

Jean-Marc Desharnais¹, Alain Abran²

¹ Senior Research Officer (TAM and Softlab),
Bogaziçi University, 34342 Bebek, Istanbul, Turkey
desharnaisjm@gmail.com

² Ecole de technologie supérieure – U. Québec, Software Engineering Department,
1100 Notre-Dame West, Montreal, Canada
alain.abran@etsmtl.ca

Abstract. The COSMIC measurement method involves applying a set of models, principles, rules, and processes to the Functional User Requirements (FUR) of a piece of software. The result is a numerical ‘value of a quantity’ representing the functional size of that software according to the COSMIC method. The quality of the results depends not only on the COSMIC rules and the measurement expertise of the measurer, but also on the quality of the documentation of the FUR. The data collection form of the International Software Benchmarking Standards Group (ISBSG) asks the data submitter to assign a quality rating to the documentation used for measuring the functional size of a project using a three-category scale. Currently, the measurer assigns such a rating subjectively, and it could vary from one measurer to another. This article proposes a procedure for use by the measurer to assess the quality of the documentation. The procedure is based on criteria defined from lessons learned while measuring functional size using the requirements documentation.

Keywords: COSMIC–ISO 19761, ISBSG, verification process, Function Points, quality of functional size, quality audit, questionnaire documentation.

A Systematic Approach to the Analysis of Function Point - COSMIC Convertibility

Luigi Lavazza

Università degli Studi dell'Insubria
luigi.lavazza@uninsubria.it

Abstract. Objective. After the COSMIC FSM method has been proposed, the issue of convertibility between traditional function points and COSMIC function points has arisen. Several studies have been performed, in order to evaluate whether a correlation between traditional and COSMIC FP measures can be established. Most of these studies applied linear regression analysis to sets of data concerning the size of projects measured in both FP and CFP. These datasets are often characterized by skewness, outliers and heteroscedasticity: it is therefore necessary to evaluate the statistical validity of the studies that used such datasets. Method. Statistical analysis techniques for dealing with the aforementioned problems have been proposed. Here we employ log-log transformations to normalize the datasets, and least median of squares regressions to decrease the dependence on outliers. The resulting models are compared using rigorous statistical methods. Results. The paper shows that different datasets tend to produce different models of the relationship between COSMIC and traditional function points. Moreover, different datasets are best modeled by different types of regressions. Conclusions. On the base of the collected evidence it seems impossible not only to claim that there is a unique formula that can be used to convert traditional function point measures into COSMIC function point (or vice versa), but even that a given type of regression yields the best models for every possible dataset. In many cases, however, robust regression techniques yield reasonably precise conversion formulas.

Keywords: COSMIC Function points, Function point analysis, Functional Size Measurement, Functional size measure convertibility, Robust regression, Data analysis, Outliers, Statistical significance

YES, we measure! A practical metrics experience

Filippo De Carli, Sonia Meocci
Shared Service Center Srl
Filippo.decarli@sscenter.it
Sonia.meocci@sscenter.it

Abstract. Many pages were written to explain to the ICT community why measurement is necessary, what are the tangible benefits of a task that for the industry is a day-byday common practice. The IT world suffers to find a wide agreement on measurement outside a short group of followers. And, when you are called to give an effective contribution to help your organization to obtain the CMMI Level 2 (before) and CMMI Level 3 (after), and without twisting your company metrics system and your counting practice, you realize that you have spent well all the resources your company has made available to your team in these years. To better understand how we reached this uneasy target, is necessary to follow the steps that led us to achieve such a clear result in the measurement field.

Keywords: Cost, Benefits, CMMI, ISO, SLA, Quality, Function Point

Presenting Software Metrics Indicators: A Case Study

Kosta Pandazo, Arisa Shollo, Miroslaw Staron, Wilhelm Meding
IT University of Copenhagen, Copenhagen Business School, Chalmers | University of
Gothenburg, Ericsson SW Research
kopa@itu.dk, as.inf@cbs.dk, miroslaw.staron@ituniv.se

Abstract. Industrial measurement systems in software projects can generate a large number of indicators (main measurements). Having a large number of indicators might result in failing to present an overview of the status of measured entities. In consequence, managers might experience problems when making decisions based on indicators. In essence, visualizing indicators and their dependencies can communicate the information to the stakeholders efficiently if done correctly, or mislead them if not done properly. In this paper we present results of a case study conducted in a unit of Ericsson. During the case study we identified the main requirements for methods for visualizing the indicators, developed these visualizations and conducted a series of interviews evaluating them. The results show that the dashboard presentation is the best solution, but that the simple, tabular visualizations are next best suited for communicating the information to the managers.

Keywords: Software metrics, Information Visualization, Indicators, Case Study

Estimating Software Effort in a Change-intensive Applications Environment: An Experience Report

Baris Ozkan, Onur Demirörs, Ali Ozkaya¹
Informatics Institute, Middle East Technical University
06531, Ankara, Turkey
{bozkan,demirors}@ii.metu.edu.tr
¹Fintek A.S
06520, Ankara, Turkey
aozkaya@fintek.com.tr

Abstract. Even though there exist a significant number of software estimation models and methods, their applicability and effectiveness are questioned from one software organization and environment to another. This paper reports experiences from an exploratory study that was conducted in an IT organization to develop an effort estimation method for software enhancements. The software changes that involve enhancements in many applications in the software environment constituted the major challenge in estimation. With the observation of homogeneity in factors of software development, an estimation approach was developed to overcome this challenge and a functional size based estimation method was developed. Several effort models were developed and evaluated in order to investigate the relation between size and effort. The decisions against the conditions and circumstances in the company are given and the experiences from an estimation method development are explained in order to aid software managers and practioners with a real life case.

Keywords: Software estimation, functional size, software portfolio, software enhancement

COSMIC and SOA Sizing A critical analysis and proposals for improvement potentials

Andreas Schmietendorf¹, Robert Neumann², Reiner Dumke²

¹ HWR Berlin – Berlin School of Economics and Law, FB II – Business Information Systems,
Neue Bahnhofstr. 11-17, 10245 Berlin, Email: andreas.schmietendorf@hwr-berlin.de

² OVG-Universität Magdeburg, FIN-IVS, Department of Distributed Systems,
39106 Magdeburg,

Email: {robneuma, dumke}@ivs.cs.ovgu.de

Abstract. The paper deals with effort estimation possibilities within projects for service oriented architectures. After an overview of related works, the specific requirements of an effort estimation method for SOA projects will be identified. These specific requirements are derived from the literature, as well as from empirical experiences within industrial projects. On this basis, a deeper analysis of the COSMIC approach for SOA projects should be executed. The aim of the analysis is the identification of strength and weaknesses, especially from the SOA communities' point of view. At the end, the realizations gained in this context are represented as suggestions for improvements of the COSMIC approach for SOA related projects.

An Empirical Analysis of Information Technology Operations Cost

*Masateru Tsunoda¹, Akito Monden¹, Ken-ichi Matsumoto¹, Akihiko Takahashi²,
and Tomoki Oshino²*

¹ Graduate School of Information Science, Nara Institute of Science and Technology
{masate-t, akito-m, matumoto}@is.naist.jp

² Economic Research Institute, Economic Research Association
{er102, er352}@zai-keicho.or.jp

Abstract. When outsourcing IT (Information Technology) operations, it is important to estimate operations cost accurately. To help IT operations cost estimation, we analyzed cross-organization data which includes 73 IT operation cases collected from Japanese companies/organizations. Operations cost is roughly decided by the number of operations staff and unit cost of staff. So we clarified attributes which have relationships to the number of staff or unit cost of staff to help analogous estimating of them. The result showed that the number of computer terminals and network range are major factors of the number of staff. Also, the number of UNIX server and process standardization are major factors of unit of staff. In addition, we tried to build estimation models for the number of staff and total cost of operations. The number of staff estimation model is used when IT operations service providers estimate operations cost, and the operations cost estimation model is assumed to be used by IT operations service purchasers. In the experiment, the number of staff estimation model showed moderate accuracy, but the total operations cost estimation model showed low accuracy.

Keywords: Estimation, staff, cost, cross organization data, linear regression

Identification of Quality Issues by systematic quantitative Analysis

Ralf Russ, Frank Rometsch
Siemens AG Corporate Technology
ralf.russ@siemens.com

Abstract. Since about four years Siemens applies more and more a methodology which has its roots in manufacturing to solve non trivial problems in the area of software engineering. When a software engineering department within the Siemens Energy sector was criticized internally for the delivered quality, they decided to follow the Six Sigma methodology for identifying improvement potentials and levers, as former attempts to increase the quality of their application did not get sustainable results in the long run. Instead of directly jumping on best practice solutions, we rigorously addressed the need to define performance indicator measurements to identify the root causes of poor quality.

Keywords: Six Sigma, problem solving, optimization, quality, practical measurement application

The Impact of Design Complexity on Software Quality - A Meta Analysis

Nguyen Duc Anh, Marcus Ciolkowski
Fraunhofer Institute for Experimental Software Engineering
anh.nguyen@iese.fraunhofer.de, marcus.ciolkowski@iese.fraunhofer.de

Abstract. The role of software quality is constantly increasing in industry. As a consequence, many techniques have been applied to assess, predict and improve quality. For example, in early development phases, design complexity metrics are considered useful indicators of software reliability. Although many studies investigate the relationship between complexity metrics and software quality, it is unclear what we have learned from these studies, because no systematic summary exists to date. This paper reports on a meta-analysis on the impact of design complexity on software quality. We aggregated 35 Spearman correlation coefficients from 29 primary studies using a tailored meta-analysis approach. The main goal of the meta-analysis was to investigate the impact of design metrics (CBO, DIT, NOC, WMC, RFC, LCOM) on fault proneness, and to compare them with the impact for LOC. Main results are that metrics of coupling and scale (size) are stronger correlated to fault proneness than cohesion and inheritance metrics, and that LOC is stronger correlated to fault proneness than all investigated design metrics. In addition, the meta-analysis showed a strong inconsistency between the different studies that we were not able to explain satisfactorily. The best explanation (defect collection phase) is able to account for more than 50% of observed variation in 5 out of 7 investigated metrics, but still leaves a significant amount of variation unexplained.

Keywords: Design metrics, complexity, measurement, meta-analysis, systematic review

Applying Human Error Assessment and Reduction Technique (HEART) in the software development process

Konstantina Georgieva, Robert Neumann, Reiner R. Dumke
Institute for Distributed Systems, University of Magdeburg,
P.O. Box 4120, 39106 Magdeburg, Germany
(ina,neumann,dumke)@ivs.cs.uni-magdeburg.de

Abstract. Human factors play a very important role in the software development process. Humans are involved in each and every phase of the software life cycle and they are not perfect but error prone because of this human errors contribute to major risks which results in the partial or complete software failure. This paper addresses the human factors that influence the human performance and in order to make sure that these human factors are critical, they are evaluated using human reliability assessment methods.

Keywords: Software development process, human factors, human reliability assessment

Software Measurement in Software Engineering Education: A Comparative Analysis

Mónica Villavicencio^{1,2}, Alain Abran¹

¹École de technologie supérieure, Montréal, Canada

²Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales, CIDIS-FIEC, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador
mvillavi@espol.edu.ec, alain.abran@etsmtl.ca

Abstract. Measurement and quantitative analysis are fundamental in engineering, and can also be found in the software engineering literature. Are they fundamental in an academic setting as well, at both the undergraduate and graduate levels? This study is aimed at gaining insights into how software measurement is being taught at universities, specifically from a practical perspective in terms of the approach to teaching it. We investigate this question through a literature survey of the works reported between 2000 and 2010 in several institutions of higher learning where software measurement topics are taught. For each publication identified in this survey, a number of topics were looked into, including teaching objectives, course content, types of projects developed by students, and measures collected, as well as the topics from the software engineering body of knowledge, the SWEBOK, that were covered. Preliminary survey results indicate that teaching efforts in universities are only partially aligned with the SWEBOK. Furthermore, the courses analyzed did not cover measurement from a global perspective, and a number of relevant measurement knowledge areas were not included. These findings can provide valuable information to help educators and researchers devise ways to enhance software measurement teaching.

Keywords: Software engineering, software measurement, education

Keynote III

Validating Delivery Performance: Benchmarking Experiences at IBM

Wai F. Hom
IBM Global Business Services

Abstract. After the excitement of implementing a new Metrics program has disappeared within an organization, how does an organization recapture the enthusiasm and vitality of the program after only a few months of data collection? What are some of the common challenges that make the management team impatient and project teams question the value of the program? Many metrics programs fail to gather any traction within an organization and are typically the first to go when budgets are scrutinized. In this presentation, hear what the IBM Corporation has done internally to propel metrics to the forefront of decision making and how the use the business intelligence derived from data has been used to drive business value, implement process improvements, and most importantly drive and sustain new business relationships.

Bewertungsaspekte serviceorientierter Architekturen (BSOA2010) - detaillierter Workshopbericht

Andreas Schmietendorf⁺, Frank Simon[#]

⁺Hochschule für Wirtschaft und Recht Berlin
Email: andreas.schmietendorf@hwr-berlin.de

[#]SQS Software Quality Systems AG
Email: Frank.Simon@sqs.de

1. Historischer Abriss

Seit nunmehr 5 Jahren beschäftigt sich die BSOA-Initiative mit der Bewertung von serviceorientierten Architekturansätzen. Zunächst beschäftigten sich die Teilnehmer im Rahmen der ersten Workshops mit der messtechnischen Erfassung der mit einer SOA einhergehenden Ausprägungen und Merkmale bzw. den involvierten Stakeholdern. Sehr schnell wurde deutlich, dass sich eine SOA weniger auf technologische Sachverhalte bezieht als vielmehr auf die veränderte Sichtweise zur Gestaltung unternehmensweit genutzter IT-Systeme. Erwartete Vorteile einer SOA bezogen sich insbesondere auf die Zielstellungen des Informationsmanagements. In diesem Zusammenhang wurden Mehrwertpotentiale durch eine verbesserte Geschäftsprozessorientierung der IT, reduzierte Daten- und Funktionsredundanzen, verringerte Komplexitäten bei Anwendungen und Schnittstellen, verringerte Kundenbindungen oder auch die Flexibilität mit der eine benötigte IT-Lösung bereitgestellt werden kann, ausgemacht.

Im Kern sollte es möglich sein, geschäftlichen Veränderungen schneller und vor allem kostengünstiger durch eine serviceorientiert aufgebaute IT-Landschaft folgen zu können. Der Übergang von einer Anwendungs- zu einer Servicesicht impliziert allerdings radikale Veränderungen in Bezug auf die Softwareentwicklung und den anschließenden Betrieb. Insbesondere für die Entwicklung bedarf es stringenter Vorgaben zur Implementierung fachlich passfähiger Serviceangebote, welche sich dafür an entsprechenden Referenzmodellen (u.a. Prozesse und Daten) zu orientieren haben. Im Zusammenhang mit der strategischen Ausrichtung und dem notwendigen Management einer SOA hat sich der Begriff der SOA-Governance als Bestandteil der IT-Governance etabliert.

In der jüngeren Vergangenheit mehren sich die Zweifel an den Vorteilen einer SOA; darüber hinaus existieren Beispiele für gescheiterte SOA-Projekte bzw. nicht erreichte Erwartungen. Unter ([Simon 2010] S. 16) wird dafür auf entsprechende Erfahrungen verwiesen. In gewisser Weise wird mit dem Begriff des SOA-Projekts bereits ein grundlegendes Dilemma aufgezeigt, da die Etablierung einer SOA eher nicht als Projektaufgabe zu verstehen ist.

Unter [Marks 2005] wird SOA mit einer strategischen Ausrichtung der Organisation verbunden:

“SOA is not a quick fix for all of your business or IT challenges, but it does provide a clear strategic pathway forward for your organization. Remember, the IT challenges that SOAs address are artefacts of years of corporate decisions and behaviours over time. It will take time to undue the accumulated technological complexity in your IT architecture.”

Gleichzeitig unterstreicht ([Marks 2006] S. 1), dass es sich bei einer SOA um kein Produkt, keine Lösung und keine Technologie handelt! Vielmehr geht es um eine gezielte Restrukturierung gewachsener Anwendungslandschaften. Die Einführung einer SOA kann daher nicht als Projektmanagement-Aufgabe gesehen werden, sondern fällt in die Verantwortung eines entsprechenden Programmmanagements.

Im Zusammenhang mit einer derartigen Zielstellung sind für Projekte der Softwareentwicklung eher Einbußen und eine massive Abnahme an Freiheitsgraden zu erwarten. Wird eine SOA ausschließlich aus einer entwicklungstechnischen Perspektive getrieben, impliziert dieses zumeist eine eingeschränkte Sicht auf die zu entwickelnde Lösung bzw. das zu integrierende System. Der Blick über den Tellerrand endet spätestens bei den eigenen Integrationsbedürfnissen zu bereits existierenden Anwendungen. Die Einbettung in eine SOA-Strategie hat da typischerweise keinen Platz. Zumeist entpuppt sich diese sogar - zumindest innerhalb eines einzelnen Projektes - als kontraproduktiv, da entsprechende Abstimmungen zeit- und kostenintensiv sind und den eigenen Projekterfolg ggf. gefährden. Wenn ein längerfristig orientiertes Programmmanagement nicht gegensteuert, ist SOA für ein solches Projekt gescheitert.

Am „Aufräumen“ über Jahrzehnte gewachsener Anwendungslandschaften bzw. an einer strategischen Ausrichtung der IT führt aus Sicht der Autoren dennoch kein Weg vorbei. Neue Heilsbringer, wie das allgegenwärtige Cloud Computing oder das ebenfalls diskutierte Enterprise Architektur Management, sind ohne die Bewältigung klassischer Integrationsprobleme bzw. die Bereitstellung serviceorientierter Lösungsansätze (impliziert nicht zwingend die Etablierung einer SOA) undenkbar. In diesem Sinne gilt es, die Entwicklung moderner Integrationsarchitekturen durch die BSOA-Initiative zu begleiten und neue Herausforderungen ggf. durch bekannte Lösungsstrategien aus dem SOA-Umfeld zu beantworten bzw. benötigte Forschungsansätze zu forcieren.

2. Beiträge des Workshops

Aus den im Jahr 2010 eingesandten Vorschlägen wurden 6 Paper für eine Präsentation während der Workshopsitzungen und 3 für eine Posterdarstellung während der Pausenzeiten ausgewählt. Mit Herrn Stefan Andres (Telekom Deutschland GmbH) und Herrn Dr. Christian Bartsch (FZI Karlsruhe) konnten darüber hinaus zwei international erfahrene Referenten als eingeladene Sprecher gewonnen werden.

Im Folgenden findet sich eine inhaltliche Zusammenfassung der auf dem Workshop gehaltenen Vorträge, die korrespondierenden Artikel finden sich im Tagungsband zum Workshop [Schmietendorf 2010].

Stefan Andres, Uwe Beyer: Governance als Schlüssel zum Business-Alignment der IT von Telekommunikationsunternehmen

In ihrem Beitrag gehen die Autoren auf die aktuellen Herausforderungen eines IP-basierten Telekommunikationsunternehmens ein. Im Mittelpunkt stehen dabei verschiedene Governance-Modelle und deren Vor- und Nachteile in Bezug auf das Business-Alignment der IT. Mit Hilfe einer Teilprodukt-Governance (Kombination aus Prozess- und Produkt-Governance) und einer geschäftsfallorientierten Harmonisierung involvierter „Fabriken“ soll eine serviceorientierte Ausrichtung in Bezug auf die Business-Sicht forciert angegangen werden.

Martin Fiedler: Beschreibungsmodell des SOA-Entscheidungsprozesses und Hypothesentest der am Entscheidungsprozess beteiligten Konstrukte

Das vorgeschlagene Beschreibungsmodell zeigt die Haupt- und Unterphasen, aber auch potentielle Einflusskriterien eines SOA-Entscheidungsprozesses detailliert auf. Mit Hilfe des Modells sollen Unternehmen in die Lage versetzt werden, eine Entscheidung für oder gegen eine SOA objektiv herzuleiten. Zur Modellverifikation werden Hypothesen hergeleitet, die mit Hilfe einer empirischen Untersuchung (Hypothesentests) bestätigt bzw. verworfen wurden.

Christian Bartsch, Marco Mevius: Analyse der Auswirkungen der Verfügbarkeit von IT-Dienstleistungen auf Geschäftsprozesse

Die modelltechnische und simulative Untersuchung der Verfügbarkeit von IT-Dienstleistungen steht im Mittelpunkt dieses Beitrags. Mit Hilfe von auf höheren Petri-Netzen-basierenden Simulationsexperimenten untersuchen die Autoren, wie sich die Verfügbarkeit von IT-Dienstleistungen auf die Performance von Geschäftsprozessen auswirkt. Die formale Kenntnis der Ausfallwahrscheinlichkeit und daraus folgender Implikationen für die unterstützten Geschäftsprozesse wird insbesondere bei der Gestaltung von IT-Dienstleistungsvereinbarungen benötigt.

Harry M. Sneed: Dynamische Qualitätsüberwachung von Web Services

Die qualitative Überwachung bzw. Kontrolle von vertraglich zugesicherten Web Service Eigenschaften ist der Gegenstand dieses Beitrags. Mit Hilfe einer 3-stufigen Qualitätssicherung geht der Autor auf die statischen Eigenschaften einer wsdl-Spezifikation (u.a. Regeln und Metriken), die funktionale Korrektheit (Zusicherungen) eines Serviceaufrufs und nicht-funktionale Eigenschaften (u.a. Antwortzeit und Verfügbarkeit) eines Web Service ein. Zur Unterstützung der statischen und dynamischen Tests werden korrespondierende Werkzeuge vorgeschlagen.

Sven Liebing, Katrin Patzer: Vorgehen zur Etablierung eines Testverfahrens für die Qualitätssicherung von Services

Das allgemeine Vorgehen zur Etablierung einer testorientierten SOA-Qualitätssicherung im industriellen Kontext wird in diesem Beitrag aufgegriffen. Zunächst wird auf die Entwicklung eines Testverfahrens unter Berücksichtigung von SOA-Prüfkriterien eingegangen und aufwandsseitige Implikationen im Kontext verschiedener Testkomplexitäten diskutiert. Im Weiteren erfolgt eine Zuordnung der Testarten zu den Phasen der Softwareentwicklung (speziell V-Modell) bzw. die Diskussion des benötigten Werkzeugeinsatzes.

Frederick Kramer: Informationsinfrastruktur in kleinen Unternehmen

Die Bedeutung serviceorientierter Architekturen für Kleinunternehmen ist Gegenstand dieses Beitrags. Mit Hilfe zweier Fallbeispiele werden klassische Einstiegshürden in eine SOA herausgearbeitet. Dabei wird insbesondere auf die geringe Bedeutung technischer Aspekte einer SOA verwiesen. Die durchaus benötigte geschäftliche Flexibilität wird typischerweise nicht einer SOA zugeschrieben. Vielmehr gilt es, die Besonderheiten kleiner Unternehmen in Bezug auf einen nachvollziehbaren Nutzen bei einem schnellen ROI und geringen Investitionskosten zu berücksichtigen.

Manfred Bundschuh: Risiken abbauen - mit Methoden aus Studien zum Problemlöseverhalten

Der Umgang mit komplexen und dynamischen Systemen ist im Alltag eines Projektmanagers allgegenwärtig (vom Autoren als „Dynaxity“ bezeichnet). Auch eine serviceorientierte Architektur kann als ein solches System verstanden werden. Der Autor geht im Rahmen seines Beitrags auf Studien zum Problemlöseverhalten im

Kontext dreier Problemarten ein, wobei insbesondere Fragen des Risikomanagements adressiert werden.

Florian Muhß: Bewertung des Integrationsaufwandes von Cloud Services

Die Bestimmung des Wertbeitrags und des Integrationsaufwands von über das Internet angebotenen Services steht im Mittelpunkt dieses Beitrags. Nach einer Analyse der verschiedenen Methoden zur Aufwandsschätzung geht der Autor auf deren Verwendbarkeit im Kontext des Cloud-Paradigmas ein. In diesem Zusammenhang werden verschiedene Cloud-Service-Typen und deren Implikationen zur Aufwandsschätzung untersucht bzw. neue Vorgehensweisen vorgeschlagen.

Neben den dargestellten Vollbeiträgen gab es die folgenden Posterpräsentationen, die insbesondere jungen Absolventen und Doktoranden vorbehalten waren:

Robert Neumann, Konstantina Georgieva, Reiner Dumke: Down-top Enterprise Application Development

Necattin Kalayci: Herausforderungen eines integrierten Change Managements im Umfeld serviceorientierter Architekturen

Ahmed A. Hussein: A Validating Framework for Business Process Based on a Defined Enterprise Security Ontology

Darüber hinaus erfolgte in einem Ad-hoc-Meeting unter Moderation von Dr. Frank Simon (SQS AG Köln) die Diskussion anstehender Herausforderungen unter dem Motto „SOA meets Innovation“.

3. Ergebnisse der Diskussionsrunde

3.1 Diskussionsthemen

Im Rahmen der moderierten Diskussionsrunde wurden aktuelle Trends und Hypes in der IT aufgegriffen und ihre möglichen Implikationen zu serviceorientierten Architekturen herausgearbeitet. Diese Implikationen sollten abschließend jeweils grob klassifiziert werden in:

- „Trend unterstützt SOA“,
- „Trend behindert SOA“ und
- „Trend ist irrelevant für SOA“.

Zur Anregung der Diskussion wurden die folgenden Trends/Hypes den Workshopteilnehmern zur Verfügung gestellt (jeweils bereits angereichert mit einer Implikationsvermutung):

1. Serviceorientierte Architekturen als Enabler für das Cloud Computing. Trend unterstützt SOA.
2. Herausforderungen beim Test von Cloud Services entsprechen denen des SOA-Testens. Trend unterstützt SOA.
3. Agile SOA – ein Widerspruch? Trend behindert SOA.

Durch die Teilnehmer erfolgte eine Priorisierung, wobei sich 33% für das 1. Thema, 27% für das 2. Thema und 40% für das 3. Thema aussprachen. Aufgrund der zeitlichen Restriktionen wurde daher der Fokus auf Thema 1 und 3 gesetzt.

3.2 Ergebnisse der Diskussion

3.2.1 SERVICEORIENTIERTE ARCHITEKTUREN ALS ENABLER FÜR CLOUD COMPUTING

Einführende Thesen des Moderators:

Die Grundfrage lautet: Benötigt die Verwendung von Cloud-Services die gleichen Voraussetzungen, die auch im Zusammenhang mit serviceorientierten Architekturen in der Diskussion stehen? Eine These dazu ist: Die heutigen Factories (sowohl im produzierenden Gewerbe als auch in der IT) setzen auf arbeitsteilige Prozesse – Cloud Services treiben die IT-Industrialisierung weiter!

Im weiteren Verlauf der Diskussion wurde eine Auseinandersetzung mit den Begriffen Cloud und SOA geführt: Cloud-Lösungen können demnach als eine über das Unternehmen hinausgehende SOA verstanden werden, daher lassen sich Methoden und Compliance-Aspekte wechselseitig verwenden.

Geschäftsfallmodell-Management als Teil der Industrialisierung treibt die Etablierung einer serviceorientierten Architektur weiter an. Damit eng verknüpft ist die Beschreibung von Cloud-Services aus geschäftlicher Sicht (Business Impact). Fein granulare Services werden keine Zukunft haben, weder im SOA- noch im Cloud-Bereich.

Cloud-Services bedürfen wie eine SOA einer differenzierten Betrachtung. Der Begriff „Cloud“ als luftige Wolke genügt dabei nicht. So verkommt Cloud als Hype ohne exakte Definition. Eine ähnliche Schwäche der unscharfen Definition hat allerdings auch SOA. Für beide gilt: Entscheidend ist die Schnittstelle, was dahinter steckt, ist eigentlich egal.

Cloud-Services dienen damit, wie SOA auch, als Treiber für die Automatisierung bzw. Industrialisierung (Services als COTS – commercial of the shelf). Trotz der diskutierten inhaltlichen Nähe zwischen SOA und Cloud verkauft sich aktuell der Begriff Cloud deutlich besser.

Neben Gemeinsamkeiten wurden auch Unterschiede zwischen SOA und Cloud diskutiert (auch kontrovers):

- Während sich SOA eher auf die technischen Aspekte bezieht, fokussiert Cloud mehr die betriebswirtschaftlichen Aspekte (Flexibilisierung der IT-Kosten).
- Das Cloud Computing zeigt viele Gemeinsamkeiten mit dem Grid Computing und sollte daher vom SOA-Begriff getrennt werden.
- Von einigen Teilnehmern wurde die Cloud (das Web als Rechner) mit dem Mainframe des 21. Jahrhundert verglichen.

Damit existiert ein weiterer Schnittpunkt zwischen Cloud und SOA: SOA kann als eine technische Instanz einer Cloud angesehen werden. Allerdings können Cloud-Lösungen auch ohne das Vorhandensein einer SOA existieren, insbesondere bei KMUs kann dieses beobachtet werden:

- sie sind offener für Cloud-Lösungen als für SOA-Ansätze,
- das Angebot eines Cloud-Services ist einfacher zu handhaben,
- Cloud-Services bieten einen BWL-seitig nachvollziehbaren (d.h. rechenbaren) Ansatz.

Cloud-Lösungen können ebenfalls als Outsourcing-Modelle verstanden werden. Die Entscheidung dafür ist üblicherweise wiederum inhaltlich getrieben. Das Cloud Computing stellt damit bzgl. der Outsourcing-Modelle „lediglich“ eine Weiterentwicklung bekannter Ansätze dar (existierende Bezüge zum Outsourcing, EAI, SOA, ...).

Für die Klassifikation hat sich die Mehrheit der Teilnehmer für die Bewertung „Cloud unterstützt SOA“ ausgesprochen.

3.2.2 SOA UND AGILE PROZESSMODELLE

Einführende Thesen des Moderators:

Die Grundfrage lautet hier: Umfangreiche Spezifikationen sind für agile Vorgehensweisen eher ungewöhnlich, würden aber bei einer SOA benötigt werden. Wie passt das mit agilen Prozessmodellen zusammen?

Konsens bestand in der Diskussion bzgl. folgender Sichten auf eine SOA und Agilität:

- SOA benötigt stabile Schnittstellenspezifikationen, welche sich an unternehmensinternen und unternehmensübergreifenden Referenzmodellen (Governance) zu orientieren haben.
- SOA bedarf eines „top down“-Ansatzes – mit agilen Vorgehensweisen hat das wenig zu tun. Agile Vorgehensweisen sind eher mit „bottom up“-Ansätzen vergleichbar.

Zu Beginn einer Softwareentwicklung existieren meist nur unscharfe Anforderungen, insbesondere der Umgang mit dieser Tatsache hat zur Entwicklung agilen Vorgehensweisen geführt.

- SOA-Ansätze könnten theoretisch die agile Implementierung dieser unscharfen Anforderungen von benötigten Lösungen unterstützen, allerdings zeigen existierende Ansätze eine ungenügende Reife.
- Aktuell implementierte Systeme werden zumeist unter zeitlichen Restriktionen implementiert (vgl. Projekt- vs. Programmmanagement weiter oben). Dem entsprechend schließen sich agile Vorgehensweisen und SOA gegenseitig aus.

Die Koexistenz agiler Vorgehensweisen und serviceorientierter Ansätze und damit die diskutierte Auswirkung von Agilität auf SOA werden maßgeblich durch die Reife einer SOA beeinflusst.

- Im Kontext einer geringen SOA-Reife behindern agile Vorgehensweisen die serviceorientierte Ausrichtung.
- Bei einer hohen SOA-Reife wären agil implementierte Lösungen zumindest denkbar; die SOA definiert dabei die Leitplanken der agilen Entwicklung.

Im Kontext der Diskussion wurde die Notwendigkeit einer Trennung zwischen Methode und Technik bei der Betrachtung von SOA und agilen Vorgehensweisen postuliert:

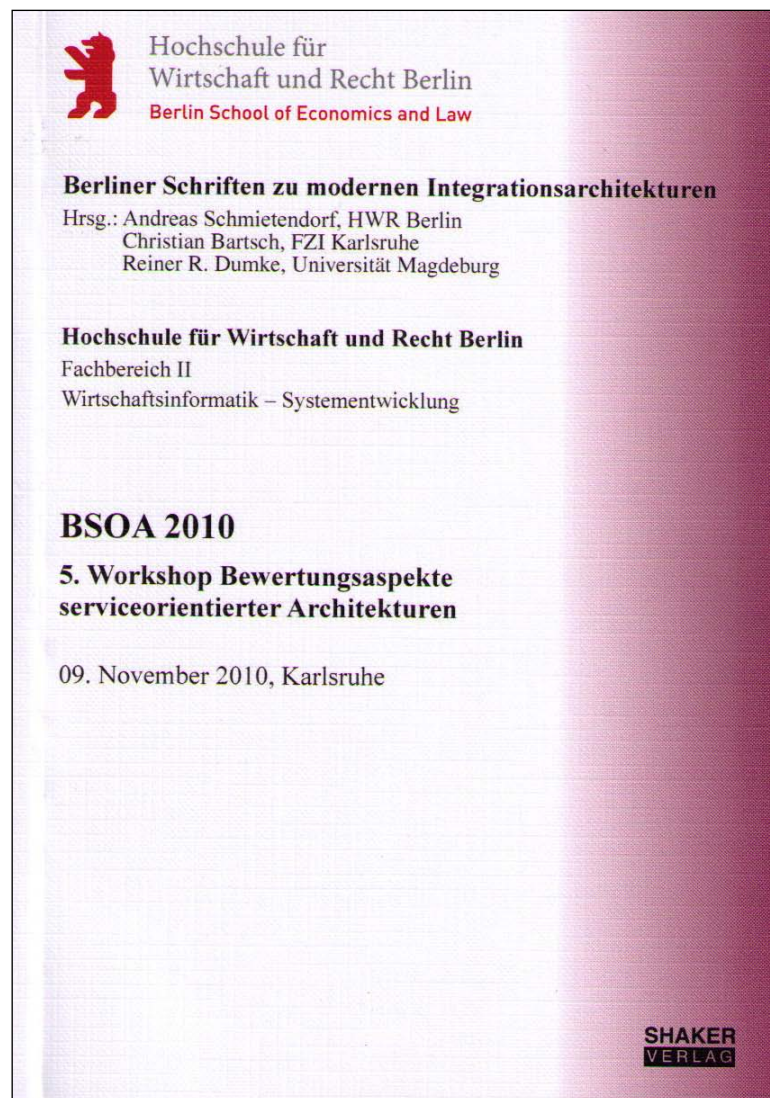
- Agilität ist eine Programmiertechnik, sie hat das Ziel eines schlanken Entwicklungsprozesses. Agilität ist damit eine Methode.
- Eine serviceorientierte Architektur stellt ein Architekturelement dar, ist also ein technisches Artefakt.

Nachträgliche Bemerkung der Autoren:

Wie schwierig der Umgang mit dem Agilitätsbegriff im Zusammenhang mit einer SOA ist, zeigen die Ausführungen von ([Marks 2006] S. 246). Dort wird von „agiler SOA“ als Schlüsselkonzept für die Entwicklung loser gekoppelter Services gesprochen, allerdings nicht zu verwechseln mit agilen Prozessmodellen. Auch ([Erl 2008] S. 87) impliziert im Zusammenhang mit einer erfolgreich implementierten SOA die Unterstützung agiler Organisationen, allerdings erkennt auch er die Problematik des für SOA benötigten „top down“-Ansatzes.

4. Weitere Informationen

Der Tagungsband zum Workshop wurde innerhalb der Schriftenreihe „Berliner Schriften zu modernen Integrationsarchitekturen“ (ISBN 978-3-8322-9313-0) publiziert.



Weitere Informationen zur BSOA-Initiative, wie z.B. der Call for Paper für den kommenden BSOA-Workshop am 15. November 2011 bei der SQS AG in Köln, befinden sich unter folgender URL im Internet:

<http://ivs.cs.uni-magdeburg.de/~gi-bsoa>

5. Quellenverzeichnis

- [Erl 2008] Erl, T.: SOA: Principles of Service Design, Prentice hall, Upper Saddle River, NJ, 2008
- [Marks 2005] Marks, E. A.: Answering the Six Most-Asked SOA Questions – White Paper, AgilePath Corporation, Newburyport/MA, 2005
- [Marks 2006] Marks, E. A.; Bell, M.: Service Oriented Architecture – A Planning and Implementation Guide for Business Technology. John Wiley & Sons, Hoboken/NJ 2006
- [Schmietendorf 2010] Schmietendorf, A.; Bartsch, C.; Dumke, R. (Hrsg.): BSOA 2010 - 5. Workshop Bewertungsaspekte serviceorientierter Architekturen, in Berliner Schriften zu modernen Integrationsarchitekturen, Shaker-Verlag, Aachen – November 2010
- [Simon 2010] Simon, F.; Simon, D.: Qualitäts-Risiko-Management – Ganzheitliche IT-Projektsteuerung, Logos Verlag, Berlin – 2010

Dank

Ohne vielfältige Unterstützung ist die Durchführung eines solchen Workshops nicht denkbar. Ein herzlicher Dank geht an den Gastgeber des 5. BSOA-Workshops 2010, Herrn Dr. Christian Bartsch vom FZI Karlsruhe, für die organisatorische Unterstützung sowie Bereitstellung von Räumen und Infrastruktur. Ein besonderer Dank gilt den Sponsoren der Initiative: die T-Systems Multimedia Solutions GmbH (Dresden), die Software AG (Darmstadt), und die Trilog AG (München). Ein Dank gilt auch allen Mitwirkenden im Programmkomitee und - last but not least – den Autoren und Referenten, die sich mit ihrem Beitrag an der Agenda des Workshops beteiligten und damit wesentlich zum Gelingen beigetragen haben.

Organisation

Veranstaltet wurde der Workshop in Kooperation zwischen der Hochschule für Wirtschaft und Recht Berlin, dem Forschungszentrum Informatik Karlsruhe und der Otto-von-Guericke-Universität Magdeburg (Softwaremesslabor) unter der Schirmherrschaft der ceCMG (Central Europe Computer Measurement Group). Darüber hinaus erfährt die BSOA-Initiative Unterstützung durch die GI (Gesellschaft für Informatik - Fachgruppe Softwaremessung- und Bewertung) und die DASMA (Deutschsprachige Interessensgruppe für Softwaremetrik und Aufwandsschätzung).

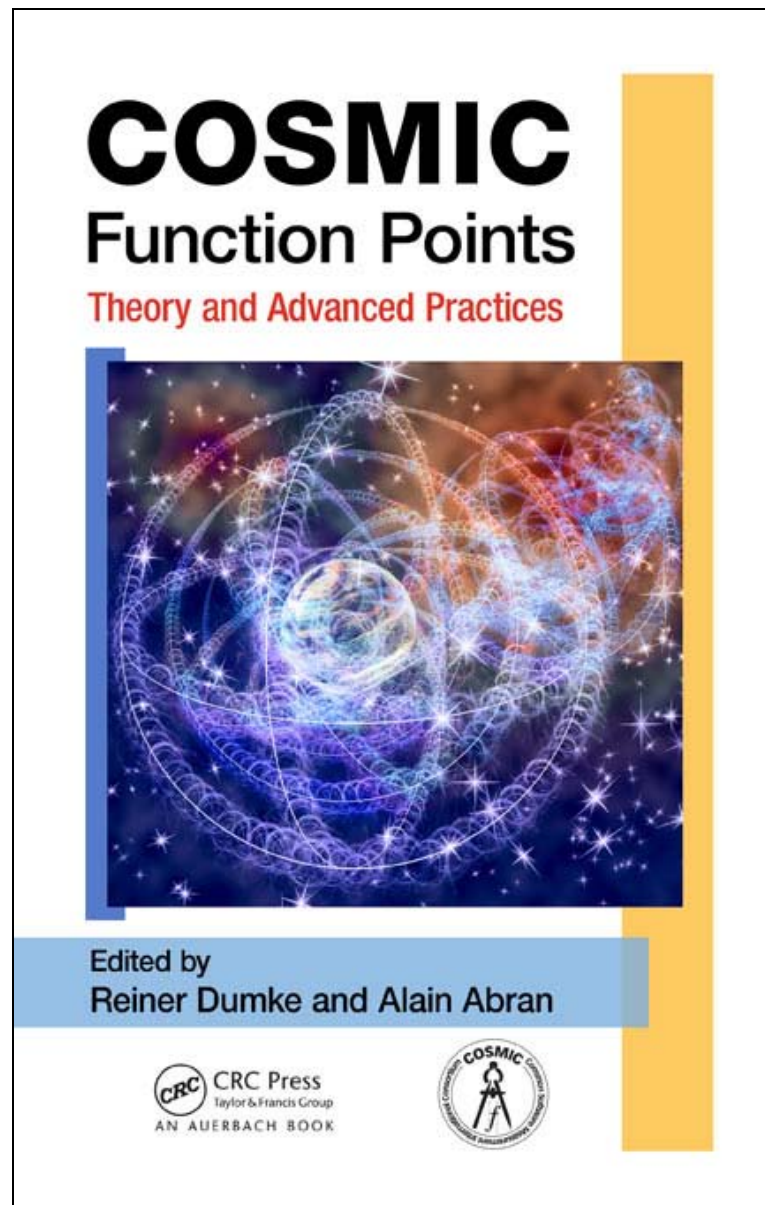
Dumke, R.; Abran, A.:

Cosmic Function Points

Theory and Advanced Practices

CRC Press Taylor & Francis Group, 2011 (334 Seiten)
ISBN: 978-1-4398-4486-1

This book has the following characteristics: the theme is about a new software size estimation method including their scientific and practical background; the chapters are based on papers, that would be published in our conference proceedings during the last six years; the authors are wellknown participants of the international software measurement community (see e. g. COSMIC, IFPUG etc.) and the book content is structured in the main problems of building new measurement or estimation methods in general and should be interesting for the software quality.



Abran, A.; Büren, G.; Dumke, R.R.; Cuadrado-Callego, J.J.; Münch, J.:

Applied Software Measurement

***Proceedings of the joined International Conferences on Software Measurement
IWSM/MetriKon/Mensura 2010, 10.-12. November 2010, Stuttgart***

Shaker Verlag, Aachen, 2010 (646 Seiten)

ISBN 978-3-8322-9618-6

ISSN 1618-7946

The book includes the proceedings of the IWSM/MetriKon/Mensura 2010 held in Stuttgart in November 2010, 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 abstracts in this Measurement News.

Schmietendorf, A.; Bartsch, C.; Dumke, R.R.:

BSOA 2010

5. Workshop Bewertungsaspekte serviceorientierter Architekturen

9. November 2010, Karlsruhe

Shaker Verlag, Aachen, 2010 (122 Seiten)

ISBN 978-3-8322-9313-0

ISSN 1867-7088

Seit nunmehr 5 Jahren beschäftigt sich die BSOA-Initiative mit der Bewertung von serviceorientierten Architekturansätzen. Zunächst beschäftigten sich die Teilnehmer im Rahmen der ersten Workshops mit der messtechnischen Erfassung der mit einer SOA einhergehenden Ausprägungen und Merkmale bzw. den involvierten Stakeholdern. Sehr schnell wurde deutlich, dass sich eine SOA weniger auf technologische Sachverhalte bezieht als vielmehr auf die veränderte Sichtweise zur Gestaltung unternehmensweit genutzter IT-Systeme. Erwartete Vorteile einer SOA bezogen sich insbesondere auf die Zielstellungen des Informationsmanagements. In diesem Zusammenhang wurden Mehrwertpotentiale durch eine verbesserte Geschäftsprozessorientierung der IT, reduzierte Daten- und Funktionsredundanzen, verringerte Komplexitäten bei Anwendungen und Schnittstellen, verringerte Kundenbindungen oder auch die Flexibilität mit der eine benötigte IT-Lösung bereitgestellt werden kann, ausgemacht.

Aus der Vielzahl an eingereichten Beiträgen konnte durch das Programmkomitee eine anspruchsvolle Agenda zusammengestellt werden. Ausgewählt wurden 6 Beiträge für eine Präsentation während der Workshopsitzungen und 3 Beiträge für Posterpräsentationen während der Pausenzeiten.

Schmietendorf, A.:

5. Hochschul-Roundtable der CECMG/DASMA

Industrielle und gesellschaftliche Herausforderungen beim flexiblen Sourcing von IT-Projekten/-Dienstleistungen

Shaker Verlag, Aachen, 2010 (82 Seiten)

ISBN 978-3-8322-8940-9

Das Buch fasst die einzelnen Diskussionsbeiträge vornehmlich aus dem industriellen Umfeld zusammen und stellt auch aktuelle Forschungsansätze zu diesem Themengebiet vor.

Dumke, R.; Mencke, S.; Wille, C.:

Quality Assurance of Agent-Based and Self-Managed Systems

CRC Press Taylor & Francis Group, 2010 (154 Seiten)

ISBN 978-1-4398-1266-2

The challenges in implementing intelligent and autonomous software systems remain to be the development of self-adapting systems, self-healing applications. Corporate global creation, and collaborated robotic teams. With software agent technology widely recognized as a key approach in implementing such global infrastructure, the importance of the role of quality assurance of agent-based systems and system development is growing daily.

Based on the author's more than 15 years of experience in software agent technology, **Quality Assurance of Agent-Based and Self-Managed Systems** presents the basic principles and structures of agent technology. It covers the main quality issues of software system development and provides examples of agent measurement and evaluation. The authors focus on software agent systems and multiagent systems (MAS) and discuss the determination of quality properties. They also explain different techniques and approaches used to evaluate the development of MAS. The final chapter summarizes quality assurance approaches for agent-based systems and discusses some open problems and future directions.

Although often complex and difficult to manage, the applications for software agent systems in essential life systems in crease every day. Since the quality of the agent-based self-managing systems is a central point of software risk; analyzing, evaluating, and improving the quality measurement situation will always be a concern when developing these systems. With more than 60 illustrations and 20 tables, this book builds a foundation in quality and quality for agent-based technology.

Abran, A.:

Software Metrics and Software Metrology

IEEE Computer Society Publ., 2010 (328 Seiten)

ISBN 978-0-470-59720-0

Software Metrics and Software Metrology looks at the fundamentals of the design of a measurement method, which forms the foundation of the measures available in the sciences and in engineering. Alain Abran provides a step-by-step approach to both analyzing the design of current software measures and designing new, robust software measures for a specific business or engineering need. He draws upon years of experience to ensure that software engineers and managers will apply the best practices in software measurement - and therefore be equipped to respond to the most demanding customers and feel supported by senior executives.

- Presents the key concepts that dictate whether a software measure's design is sufficiently strong.
- Features several case studies analyzing strengths and weaknesses in the design of some of the software measures most widely used or quoted.
- Describes how lessons learned led to the design of the COSMIC - ISO 19761 method for the measurement of the functional size of software, from its initial prototype to its adoption by the ISO as an international standard.
- Illustrates day-to-day software measurement issues that have not been seriously addressed, from convertibility across measurement designs to measurement standard etalons.
- Includes chapter exercises for classroom use.

Software Metrics and Software Metrology is meant for software quality specialists and process improvement analysts and managers, in software organizations of all sizes. In addition, this book introduces many of the theoretical concepts and references needed by professionals, managers, and students to help them understand the fundamentals of the identification and evaluation of software development and maintenance processes, as well as improvements to them.

Jones, C.:

Software Engineering Best Practices

Lessons from Successful Projects in the Top Companies

McGraw-Hill Companies, 2010 (660 Seiten)

ISBN 978-0-07-162161-8

This in-depth volume examines software engineering topics that are not covered elsewhere: the question of why software engineering has developed more than 2,500 programming languages; problems with traditional definitions of software quality; and problems with common metrics, "lines of code", and "cost per defect" that violate standard economic assumptions. The book notes that a majority of "new" projects are actually replacements for legacy applications, illustrating that data mining for lost

requirements should be a standard practice. Difficult social engineering issues are also covered, such as how to minimize harm from layoffs and downsizing.

Software Engineering Best Practices explains how to effectively plan, size, schedule, and manage software projects of all types, using solid engineering procedures. It details proven methods, from initial requirements through 20 year of maintenance. Portions of the book have been extensively reviewed by key engineers from top companies, including IBM, Microsoft, Unisys, and Sony.

Humphrey, W.S.; Thomas, W.R.:

Reflections on Management

How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself

Addison-Wesley, 2010 (288 Seiten)

ISBN-10: 0-321-71153-X

ISBN-13: 978-0-321-71153-3

This book, drawn from Humphreys books, articles, and columns, comprises a collection of advice, stories, and hard-earned wisdom, rather than specific instruction on how to implement the PSP or TSP (which are thoroughly covered in Humphreys book on those specific subjects). What emerges for the reader is an understanding that successful software project management is a journey with many obstacles. To succeed, engineers must manage more than their projects. They must use their own experience and that of their teams to first understand and then plan the project ahead. They must influence their teams' attitudes and methods for doing disciplined work. And they must persuade their bosses to set aside ill-informed notions of schedules and resource commitments and look instead at hard, historical data.

The essays in Part I provide insights on types of plans and the planning process. Part II covers team building and motivation. Part III describes how to work with your managers and persuade them to use best practices. And Part IV examines your personal responsibilities, commitments, and processes.

These essays shine a light on the challenges inherent in software development and can set engineers on the road to understanding how to succeed. And while Humphreys particular expertise is software, practitioners in every field of business will benefit from the wisdom and advice contained here.

Chemuturi, M.:

Software Estimation Best Practices, Tools & Techniques

A Complete Guide for Software Project Estimators

J. Ross Publishing, 2009 (298 Seiten)
ISBN 978-1-60427-024-2

Software Estimation Best Practices, Tools & Techniques covers all facets of software estimation. It provides a detailed explanation of the various methods for estimating software size, development effort, cost, and schedule, including a comprehensive explanation of test effort estimation. This unique desk reference, for the novice to expert, also offers direction on which methods are most appropriate for each of the different software project types commonly executed and criteria for selecting software estimation tools.

- Presents software estimation best practices and shows how to avoid common pitfalls.
- Demonstrates a practical methodology with templates for using Delphi estimation and analogy-based estimation for software projects.
- Introduces a new method referred to as software size units for measuring software size that does not make use of the untenable concept of "complexity" for adjusting software size.
- Provides useful methods for converting software size to effort, deriving true productivity, and analyzing variances between actual and estimated values as a tool for productivity improvement.
- WAV offers a free downloadable test effort estimation tool (TPPal), a software size unit estimation tool (SSUPal), and a 180-day demo for a comprehensive estimation tool known as EstimatorPal - available from the Web Added Value™ Download Resource Center at www.jrosspub.com.

Abran, A.; Braungarten, R.; Dumke, R.R.; Cuadrado-Gallego, J.J.; Brunekreef, J.:

Software Process and Product Measurement

International Conferences IWSM 2009 and Mensura 2009
Amsterdam, The Netherlands, November 4-6, 2009

Springer-Verlag, 2009 (346 Seiten)
LNCS 5891, ISBN 978-3-642-05414-3

Since 1990 the International Workshop on Software Measurement (IWSM) has been celebrated annually alternating between Montréal (Canada) and various cities across Germany. The Montréal editions have been organized by the Software Engineering Research Laboratory (GELOG) of the École de technologie supérieure – Université Québec, which is directed by Prof. Alain Abran. The German editions have been organized jointly by the Software Measurement Laboratory (SMLAB) of the Otto von Guericke University Magdeburg (Germany), which is directed by Prof. Reiner R. Dumke.

Since 2006 the Spanish software measurement community directed by Prof. Juan J. Cuadrado-Gallego as MENSURA participate at our conferences also.

This volume comprises the proceedings of IWSM / Mensura 2009 and consists of the final papers presented at these joint events. Each one of these papers has been thoroughly revised and extended in order to be accepted for publication.

Büren, G.; Dumke, R.:

MetriKon 2009 – Praxis der Software-Messung

Tagungsband des DASMA Software Metrik Kongresses

19.-20. November 2009, Kaiserslautern

Shaker Verlag, Aachen, 2009 (320 Seiten)

ISBN 978-3-8322-8649-1

The book includes the proceedings of the DASMA Metric Conference MetriKon 2009 held in Kaiserslautern in November 2009, 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.

Schmietendorf, A.; Fiedler, M.; Dumke, R.R.:

BSOA 2009

4. Workshop Bewertungsaspekte serviceorientierter Architekturen

18. November 2009, Darmstadt

Shaker Verlag, Aachen, 2009 (142 Seiten)

ISBN 978-3-8322-8551-7

Moderne Integrationsarchitekturen bestimmen nach wie vor die Art und Weise, wie softwaretechnische Lösungen im Kontext industrieller Problemstellungen umzusetzen sind. Immer stärker wird den Verantwortlichen dabei bewusst, dass der Erfolg unternehmensweit genutzter Integrationsarchitekturen weniger von einzukaufenden Produkten abhängt, als vielmehr von prozess- und organisationsbezogenen Aspekten beeinflusst wird. Die Implementierung einer serviceorientierten Architektur ist als Strategie zu verstehen, wobei die Vision einer geschäftsprozesskonformen Ausrichtung der IT-Landschaft verfolgt wird. In diesem Kontext werden vielfältige Bewertungsansätze benötigt, die den gesamten Lebenszyklus eines Informationssystems erfassen können. Die BSOA-Initiative widmet sich seit mehr als 3 Jahren diesen Fragen.

Aus der Vielzahl an eingereichten Beiträgen konnte durch das Programmkomitee eine anspruchsvolle Agenda zusammengestellt werden. Ausgewählt wurden 6 Beiträge für eine Präsentation während der Workshopsitzungen und 4 Beiträge für Posterpräsentationen während der Pausenzeiten. Dazu kommen noch die beiden Keynote-Vorträge, die das vorliegende Buch in exzellenter Weise ergänzen und prägen.

Schmietendorf, A.:

***Aufwandsschätzung bei Projekten nach dem eXtreme
Programming-Paradigma***

Shaker Verlag, Aachen, 2009 (643 Seiten)
ISBN 978-3-8322-8560-9

In diesem Buch wird eine monografische Darstellung grundlegender Aufwands-schätzmethoden im Umfeld von Ansätzen der agilen Software-Entwicklung diskutiert.

Schneider, K.:

***Experience and Knowledge Management in
Software Engineering***

Springer-Verlag, 2009, XVI, Hardcover (235 Seiten)
ISBN 978-3-540-95879-6

Nowadays, there is software everywhere in our life. It controls cars, airplanes, factories, medical implants. Without software, banking, logistics and transportation, media, and even scientific research would not function in the accustomed way. Building and maintaining software is a knowledge-intensive endeavour and requires that specific experiences are handled successfully. However, neither knowledge nor experience can be collected, stored, and shipped like physical goods, instead these delicate resources require dedicated techniques. Knowledge and experience are often called company assets, yet this is only part of the truth: it is only software engineers and other creative employees who will effectively exploit an organisation's knowledge and experience.

Kurt Schneider's textbook is written for those who want to make better use of their own knowledge and experience – either personally or within their group or company. Everyone related to software development will benefit from his detailed explanations and case studies: project managers, software engineers, quality assurance responsables, and knowledge managers. His presentation is based on years of both practical experience, with companies such as Boeing, Daimler, and Nokia, and research in renowned environments, such as the Fraunhofer Institute. Each chapter is self-contained, it clearly states its learning objectives, gives in-depth presentations, shows the techniques' practical relevance in application scenarios, lists detailed references for further reading, and is finally completed by exercises that review the material presented and also challenge further, critical examinations. The overall result is a textbook that is equally suitable as a personal resource for self-directed learning and as the basis for a one-semester course on software engineering and knowledge management.

Preprints/Technical Reports:

Farooq, A.; Dumke, R. R.: *Evaluation Approaches in Software Testing*. University of Magdeburg 2008

Richter, K.; Dumke R. R.: *The Causal-Based Software Process Modelling*. University of Magdeburg 2008

Dumke, R.; Kunz, M.; Farooq, A.; Georgieva, K.; Hegewald, H.: *Formal Modelling of Software Measurement Levels of Paradigm-Based Approaches*. University of Magdeburg 2008

see as pdf files:

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

WOSP 2011:

7th International Workshop on Software & Performance

March 14-16, Karlsruhe, Germany, 2011

see: <http://icpe2011.ipd.kit.edu/>

IASTED SE 2011:

IASTED International Conference on Software Engineering 2010

February 15-17, 2011, Innsbruck, Austria

see: <http://www.iasted.org/conferences/ipc-720.html>

SQMB 2011:

4. Workshop zur Software-Qualitätsmodellierung und -Bewertung

February 21, 2011, Karlsruhe, Germany

see: <http://se2011.ipd.kit.edu/programm/>

CSMR 2011:

15th European Conference on Software Maintenance and Reengineering

March 1-4, 2011, Oldenburg, Germany

see: <http://www.se.uni-oldenburg.de/csmr2011/>

SEPG 2011:

23th Software Engineering Process Group Conference

March 21-24, 2011, Portland, Oregon, USA

see: <http://www.sei.cmu.edu/sepg/na/2011/>

EASE 2011:

International Conference on Empirical Assessment in Software Engineering

April 11-12, 2011, Keele University, UK

see: <http://www.scm.keele.ac.uk/ease/>

STAREAST 2011:

Software Testing Analysis & Review Conference

May 1-6, 2011, Orlando, FL, USA

see: <http://www.sqe.com/stareast/>

iqnite 2011:

Software Quality Conference

May 24-26, 2011, Düsseldorf, Germany

see: <http://www.iqnite-conferences.com/de/index.aspx>

ICSE 2011:

International Conference on Software Engineering

May 21-28, 2011, Honolulu, Hawaii, USA

see: <http://2011.icse-conferences.org/>

PSQT 2011:**International Conference on Practical Software Quality & Testing**

West: May 9-13, 2011, Las Vegas;

North: Sept. 12-16, 2011, Minneapolis, USA

see: http://www.psqtconference.com_**XP 2011:****12th International Conference on Agile Software Development**

May 10-13, 2011, Madrid, Spain

see: <http://www.xp2011.org/>**SPICE 2011:****SPICE Conference**

May 30 - June 1, 2011, Dublin, Ireland

see: <http://www.spiceconference.org/>**SMEF 2011:****Software Measurement European Forum**

June 9-10, 2011, Rome, Italy

see: <http://www.iir-italy.it/smef2011eng>**PROFES 2011:****12th International Conference on Product Focused Software Process Improvement**

June 20-22, 2011, Torre Canne, Brasil

see: <http://serlab.di.uniba.it/profes2011/>**ICSECS 2011:****2th International Conference on Software Engineering and Computer Systems**

June 27-29, 2011, Pahang, Malaysia

see: <http://www.sdiwc.net/my/>**SEPG Europe 2011:****Software Engineering Process Group Conference**

June 7-9, 2011, Dublin, Ireland

see: <http://www.sei.cmu.edu/sepg/europe/2011>**ENASE 2011:****6th International Conference on Evaluation of Novel Approaches to Software Engineering**

June 8-11, 2011, Beijing, China

see: <http://www.enase.org/>

REFSQ 2011:

17th International Working Conference on Reuquirements Engineering: Foundation for Software Quality
June 28-30, 2011, Essen, Germany
see: <http://www.sse.uni-due.de/refsq/2011/>

ICPC 2010:

19th International Conference on Program Comprehension
June 22-24, 2011, Kingston, Canada
see: http://icpc2011.cs.usask.ca/conf_site/Dates.html/

ICWE 2011:

International Conference on Web Engineering
June 20-24, 2011, Paphos, Cyprus
see: <http://icwe2011.webengineering.org/>

UKPEW 2011:

24th Annual United Kingdom Workshop on Performance Engineering
July 7-8, 2011, Bradford, UK
see: <http://ukpew.org>

ISSTA 2011:

International Symposium on Software Testing and Analysis
July 17-21, 2011, Toronto, Canada
see: <http://issta11.unl.edu/>

QSIC 2010:

11th International Conference on Software Quality
July 13-14, 2011, Madrid, Spain
see: <http://antares.sip.ucm.es/qsic2011/>

SERA 2011:

9th ACIS Conference on Software Engineering
August 10-12, 2011, Baltimore, USA
see: <http://acis.cps.cmich.edu/SERA2011/>

ICGSE 2010:

6th International Conference on Global Software Engineering
August 15-18, 2011, Helsinki, Finland
see: <http://icgse2011.soberit.hut.fi/>

QEST 2011:

6th International Conference on Quantitative Evaluation of SysTems
September 5-8, 2011, Aachen, Germany
see: <http://www.qest.org/qest2011>

ASQT 2011:**Arbeitskonferenz Softwarequalität und Test**

September 28-30, 2011, Graz, Austria

see: <http://www.asqt.org/>**CONQUEST 2011:****12. International Conference on Software Quality**

September , 2011, Nuremberg, Germany

see: <http://www.conquest-conference.org/>**ESEM 2011:****5th International Symposium on Empirical Software Engineering & Measurement**

September 22-23, 2011, Alberta, Canada

see: <http://esem.cpsc.ucalgary.ca/esem2011/index.html>**UKSMA 2011:****20th Annual UKSMA Conference - Managing your Software (through Measurement)**

October 27-28, 2011, London, UK

see: <http://www.uksma.co.uk/>**IWSM/Mensura 2011:****Common International Conference on Software Measurement**

November 2-4, 2011, Nara, Japan

see: <http://mensura.wordpress.com/>**BSOA 2011:****6. Workshop Bewertungsaspekte service-orientierte Architekturen**

November 15, 2011, SQS, Colonia, Germany

see: <http://www-ivs.cs.uni-magdeburg.de/~gi-bsoa/>**MetriKon 2011:****Common international Conference on Software Measurement**

November 16-18, 2011, IESE, Kaiserslautern, Germany

see: <http://iws2010.cs.uni-magdeburg.de/cfp2011.htm>

SWQD 2012:**Software Quality Days**

January 17-19, 2012, Vienna, Austria

see: <http://www.software-quality-days.at/>

see also: OOIS, ECOOP and ESEC European Conferences

Other Information Sources and Related Topics

- **<http://rbse.jsc.nasa.gov/virt-lib/soft-eng.html>**
Software Engineering Virtual Library in Houston
- **<http://www.mccabe.com/>**
McCabe & Associates. Commercial site offering products and services for software developers (i. e. Y2K, Testing or Quality Assurance)
- **<http://www.sei.cmu.edu/>**
Software Engineering Institute of the U. S. Department of Defence at Carnegie Mellon University. Main objective of the Institute is to identify and promote successful software development practices.
Exhaustive list of publications available for download.
- **<http://dxsting.cern.ch/sting/sting.html>**
Software Technology Interest Group at CERN: their WEB-service is currently limited (due to "various reconfigurations") to a list of links to other information sources.
- **<http://www.spr.com/index.htm>**
Software Productivity Research, Capers Jones. A commercial site offering products and services mainly for software estimation and planning.
- **<http://www.qucis.queensu.ca/Software-Engineering/>**
This site hosts the World-Wide Web archives for the USENET usegroup comp.software-eng. Some links to other information sources are also provided.
- **<http://www.esi.es/>**
The European Software Institute, Spain
- **<http://www.lrgl.uqam.ca/>**
Software Engineering Management Research Laboratory at the University of Quebec, Montreal. Site offers research reports for download. One key focus area is the analysis and extension of the Function Point method.
- **<http://www.SoftwareMetrics.com/>**
Homepage of Longstreet Consulting. Offers products and services and some general information on Function Point Analysis.
- **<http://www.utexas.edu/coe/sqi/>**
Software Quality Institute of the University of Texas at Austin. Offers comprehensive general information sources on software quality issues.
- **<http://www.trese.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>
Annotated bibliography on Object-Oriented Metrics
- <http://www.iso.ch/9000e/forum.html>
The ISO 9000 Forum aims to facilitate communication between newcomers to Quality Management and those who have already made the journey have experience to draw on and advice to share.
- <http://www.qa-inc.com/>
Quality America, Inc's Home Page offers tools and services for quality improvement. Some articles for download are available.
- <http://www.quality.org/qc/>
Exhaustive set of online quality resources, not limited to software quality issues
- <http://freedom.larc.nasa.gov/spqr/spqr.html>
Software Productivity, Quality, and Reliability N-Team
- <http://www.qsm.com/>
Homepage of the Quantitative Software Management (QSM) in the Netherlands
- <http://www.iese.fhg.de/>
Homepage of the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany
- <http://www.highq.be/quality/besma.htm>
Homepage of the Belgian Software Metrics Association (BeSMA) in Keebergen, Belgium
- http://www.cetus-links.org/oo_metrics.html
Homepage of Manfred Schneider on Objects and Components
- <http://dec.bournemouth.ac.uk/ESERG/bibliography.html>
An annotated bibliography of object-oriented metrics of the Empirical Software Engineering Research Group (ESERG) of the Bournemouth University, UK

News Groups

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

Software Measurement Associations

- **<http://www.dasma.org>**
DASMA Deutsche Anwendergruppe für SW Metrik und Aufwands-schätzung e.V.
- **<http://www.aemes.fi.upm.es>**
AEMES Association Espanola de Metricas del Software
- **<http://www.cosmicon.com>**
COSMIC Common Software Measurement International Consortium
- **<http://www.esi.es>**
ESI European Software Engineering Institute in Bilbao, Spain
- **<http://www.mai-net.org/>**
Network (MAIN) Metrics Associations International
- **<http://www.sttf.fi>**
FiSMA Finnish Software Metrics Association
- **<http://www.iese.fhg.de>**
IESE Fraunhofer Einrichtung für Experimentelles Software Engineering
- **<http://www.isbsg.org.au>**
ISBSG International Software Benchmarking Standards Group, Australia
- **<http://www.nesma.nl>**
NESMA Netherlands Software Metrics Association
- **<http://www.sei.cmu.edu/>**
SEI Software Engineering Institute Pittsburgh
- **<http://www.spr.com/>**
SPR Software Productivity Research by Capers Jones
- **<http://fdd.gsfc.nasa.gov/seltext.html>**
SEL Software Engineering Laboratory - NASA-Homepage
- **<http://www.vrz.net/stev>**
STEV Vereinigung für Software-Qualitätsmanagement Österreichs
- **<http://www.sqs.de>**
SQS Gesellschaft für Software-Qualitätssicherung, Germany
- **<http://www.ti.kviv.be>**
TI/KVIV Belgish Genootschap voor Software Metrics
- **<http://www.ukσμα.co.uk>**
UKSMA United Kingdom Software Metrics Association

Software Metrics Tools (Overviews and Vendors)**Tool Listings**

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

Tool Vendors

- <http://www.mccabe.com>
McCabe & Associates
- <http://www.scitools.com>
Scientific Toolworks Inc.
- <http://zing.ncsl.nist.gov/webmet/>
Web Metrics
- <http://www.globalintegrity.com/csheets/metself.html>
Global Integrity
- <http://www.spr.com/>
Software Productivity Research (SPR)
- <http://jmetric.it.swin.edu.au/products/jmetric/>
JMetric
- <http://www.imagix.com/products/metrics.html>
Imagix Power Software
- <http://www.verilogusa.com/home.htm>
VERILOG (LOGISCOPE)
- <http://www.qsm.com/>
QSM

SOFTWARE MEASUREMENT NEWS

VOLUME 16

2011

NUMBER 1

CONTENTS

Announcements	3
Workshop Report	9
New Books on Software Metrics	41
Conferences Addressing Metrics Issues	51
Metrics in the World-Wide Web	55

ISSN 1867-9196