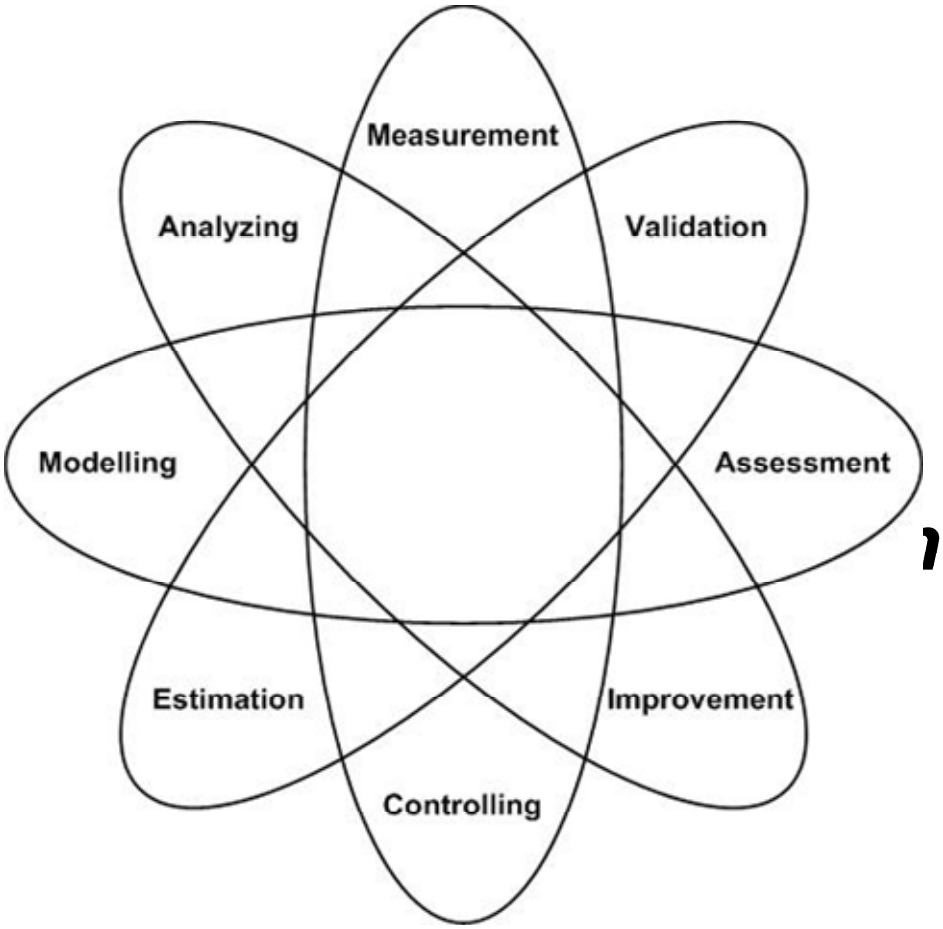
SOFTWARE MEASUREMENT NEWS

Volume 13, Numb



ne Softw

Université du Québec
École de technologie supérieure

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

Manfred Bundschuh
Chair of the DASMA
Sander Höhe 5, 51465 Bergisch Gladbach, Germany
***Tel.:** +49-2202-35719*
manfred.bundschuh@netcologne.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

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

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

Technical Editor: DI Martin Kunz

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

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

Université du Québec
École de technologie supérieure

IWSM2008

18th International Workshop
on Software Measurement

MetriKon2008

DASMA Metrik Kongress

SIEMENS

Mensura2008

3rd International Conference on Software Process and Product Management

18-19 November 2008, Siemens AG, Munich, Germany

THEME & SCOPE:

Software measurement and metrics are key technologies to manage and to control software development projects. Measurement is essential of any engineering activity, by increasing the scientific and technical knowledge for both the practice of software development and for empirical research in software technology. This congress facilitates 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
Measurement acceptance
Functional size measurement
Software estimation
Software process improvement
Metrics for specific areas, e.g. for web services
Metrics for system engineering, integration, and testing
Measurement databases
Metrics validation
Measurement services
Measurement tools
Measurement experience and guidance
Theory of measurement
Measurement paradigms
Enterprise embedded solutions
Software benchmarking

SUBMISSIONS:

Authors should send proposed *short papers (2 to 4 pages)* by e-mail by **June 9, 2008** to

Alain Abran	***Günter Büren***	***Alain Abran***	***Reiner Dumke***
ÉTS, Montreal, Canada	Büren & Partner, Nuremberg, Germany	University of Alcala, Madrid, Spain	University of Magdeburg, Germany
aabran@ele.etsmtl.ca	**gb@bup-nbg.de**	**jjcg@uah.es**	**dumke@ivs.cs.uni-magdeburg.de**

Papers should not have already been published elsewhere. Nor should they have been submitted to a journal or to another conference. At least one among the authors of each paper accepted should register for the conference and ensure paper presentation. Conference languages are English and German. German papers will be presented in a separate track.

CONFERENCE TIMETABLE:

Submission deadline of paper:	June 09, 2008
Notification of acceptance:	August 04, 2008
Final paper deadline:	September 22, 2008
Conference date:	November 18-19, 2008

CONTACT:

DASMA e.V.:

Romy Robra (Tel. +49 - 9126 - 29 79 576, info@dasma.org);

Uni Magdeburg:

Dagmar Dörge (Tel. +49 - 391 - 67 18 664, doerge@ivs.cs.uni-magdeburg.de)

FURTHER INFORMATION *(including author guidelines and templates)*

`http://www.dasma.org`

`http://iwsm2008.cs.uni-magdeburg.de`

`http://www.metrikon.de`

Ankündigung des 3. Workshops „Bewertungsaspekte serviceorientierter Architekturen“

20. November 2008 München (Gastgeber T-Systems)

MOTIVATION

Im Rahmen der von der CECMG, der GI und der DASMA getragenen BSOA-Initiative wird es auch im Jahr 2008 einen entsprechenden Workshop geben. Die ersten beiden Workshops haben das große Interesse an dieser Themenstellung im industriellen und akademischen Kontext verdeutlicht. Der kommende Workshop wird sich insbesondere mit den folgenden Themen auseinandersetzen.

- Wirtschaftlichkeitsbetrachtungen bei der SOA-Einführung,
- Modelle zur Bewertung der SOA-Tauglichkeit einer Organisation,
- Aufwands- und Risikobetrachtungen bei SOA-Entwicklungsprojekten,
- Erarbeitung von Richtlinien zur Serviceentwicklung für eine SOA,
- Qualitäts- und Sicherheitsbewertung angebotener Services und Kompositionen,
- Bewertungsaspekte beim Management serviceorientierter IT-Infrastrukturen,
- Bewertung der Möglichkeiten SOA-basierter Konvergenzangebote.

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 serviceorientierter Architekturen tätig sind, werden gebeten, Beiträge im doc- oder pdf-Format 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 erscheinen in einem Tagungsband.

Bitte senden Sie ihre Beiträge per E-Mail an

`gi-bsoa@ivs.cs.uni-magdeburg.de`

PROGRAMMKOMITEE

S. Aier, Universität St. Gallen	J. vom Brocke, HS Liechtenstein	E. Dimitrov, T-Systems
R. Dumke, Universität Magdeburg	T. Grawe, Advicio	J. M. Gomez, Uni Oldenburg
W. Greis, TPS Data & CECMG	M. Irtmann, IBM	S. Kosterski, Toll Collect
G. Limböck, SAP	M. Lother, Robert Bosch GmbH	M. Mevius, FZI Karlsruhe
R. Molle, ITAB Hamburg	S. Nakonz, Bitnologie	H. Pundt, HS Harz
A. Schmietendorf, FHW Berlin	J. Schuck, MATERNA GmbH	J. Siederleben, T-Systems
F. Victor, FH Köln	C. Wille, FH Bingen	R. Zarnekow, TU Berlin

TERMINE

20.08.2008	Einreichung von Beiträgen
20.09.2008	Annahme/Ablehnung
25.09.2008	finales Workshop-Programm
15.10.2008	Abgabe der druckreifen Beiträge
20.11.2008	Workshop in München

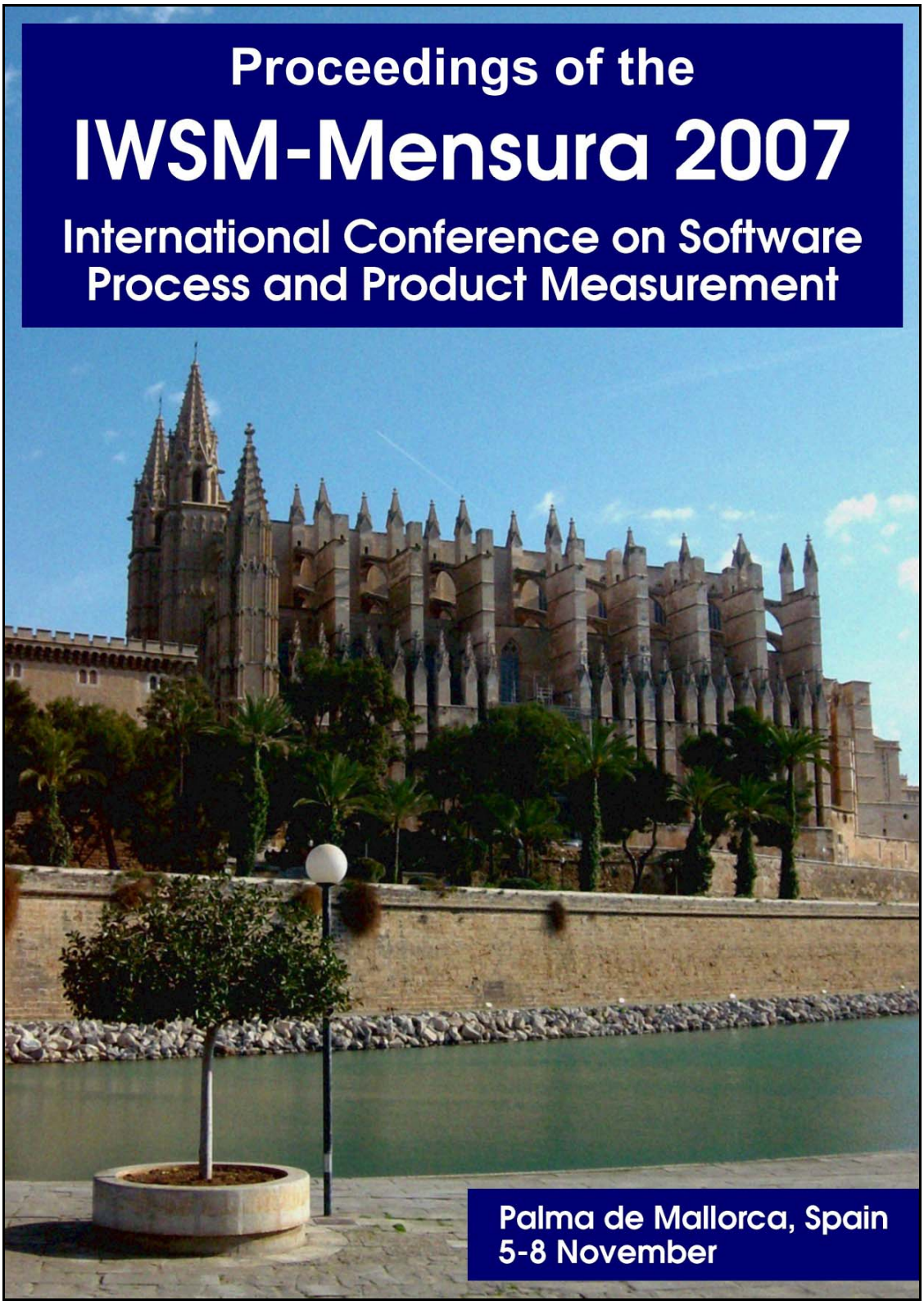
WEBSEITE ZUM WORKSHOP

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

Announcements

Fachhochschule für Wirtschaft Berlin
Berlin School of Economic

Our International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-Mensura 2007) took place in Palma de Mallorca, Spain in November 2007. The following report gives an overview about the presented papers. Furthermore, the papers are published in Edicions UIB (ISBN 978-84-8384-020-7):



Improving the Quality of Information for Software Project Management

Michael Berry and Chris S. Johnson

University of Technology, Sydney

mberry@it.uts.edu.au and chrisj@it.uts.edu.au

Abstract. *Background.* The goal of software measurement and analysis is to produce information for the people performing technical and managerial tasks. Information-centric assessment can identify opportunities to better meet that goal. This paper presents a study into the assessment of Information Quality for Software Project Management using generic and targeted information quality models. The setting for the study is a university software project management subject in which the students must complete a major exercise as part of the subject that attempts to replicate real-world project planning. *Methods.* After the exercise has been completed, the students are surveyed about the quality of the information with which they were supplied to complete the project planning exercise. The survey instrument is based on a generic model of information quality and a targeted model that focuses on the information for specific planning tasks. The study has been run with two cohorts of students and was run with a third cohort at the end of May 2007. *Results.* Improvement opportunities were identified through the 2006 assessment. These improvements were implemented for the third student cohort and the success of these improvements evaluated. The information quality assessment methods used in this study are likely to be of benefit to software project managers by providing the means to identify and remedy deficiencies in the information that they use to carry out their activities.

Developing and Applying a Consolidated Evaluation Framework to Analyze Test Process Improvement Approaches

Ayaz Farooq and Reiner R. Dumke

Institute for Distributed Systems, University of Magdeburg, P.O. Box 4120, 39106 Magdeburg, Germany

farooq@ivs.cs.uni-magdeburg.de, dumke@ivs.cs.uni-magdeburg.de

Abstract. Following the general software process improvement initiatives, improvement models particular to the software test process have also been developed. Although these models have been based on nearly similar principles, yet they differ in a variety of aspects. To compare and critically analyze strengths and weaknesses of these test process models, we need a common analysis framework. Existing SPI evaluation frameworks focus only on some basic and key process characteristics and elements. A number of critical success factors for software process improvement have been identified in literature which impose higher level of requirements on SPI approaches. We present a consolidated evaluation framework derived from critical success factors/literature review and apply it to analyze well known test process improvement approaches.

Keywords: Software process improvement, software test process, test process improvement, Testing Maturity Model, Critical success factors

IT Process Conformance Measurement: A Sarbanes-Oxley Requirement

Rafik Ouanouki[1], Dr. Alain April[2]

[1] RONA, Quality Assurance, 220 Chemin du Tremblay,
Boucherville, Québec, Canada
rafik.ouanouki@rona.ca

[2] École de Technologie Supérieure, 1100 rue Notre-Dame Ouest,
Montréal, Québec, Canada
alain.april@etsmtl.ca

Abstract. In 2006, the most important Canadian distributor and retailer of hardware products was faced with formalizing its internal processes in response to the requirements of the Sarbanes-Oxley Act. This publication is a follow-up information to our first paper on process conformance and audits [7]. The software testing process and how conformance to the documented process was achieved within 3 months is described. This research paper presents an introduction, the testing process itself, the process conformance measurement and the results obtained during the first three months of measurement.

Keywords: Quality Assurance, Testing Process, Process Conformance, Conformance Measurement, and Quality Audit

Assessment of Software Process and Metrics to Support Quantitative Understanding

Dr. Ayça Tarhan[1], and Assoc.Prof. Onur Demirors[2]

[1] Computer Engineering Department, Hacettepe University,
Beytepe Kampusu, 06532 Ankara, Turkey
atarhan@hacettepe.edu.tr

[2] Informatics Institute, Middle East Technical University,
MM Kat:4, 06531 Ankara, Turkey
demirors@ii.metu.edu.tr

Abstract. The use of process metrics and data for quantitative understanding is not very straightforward. If we have an identification of process components and follow a measurement process, we are likely to use process metrics and data effectively. But if we don't have these practices, we can hardly trust on process metrics and data for quantitative understanding. In this paper, we summarize eight case studies that we performed in different industrial contexts. The case studies rely on an assessment approach that investigates suitability of a software process and metrics for quantitative analyses. The approach investigates a process's inner attributes and outer factors as well as a number of usability characteristics for process metrics. We validated the results of the studies via SPC tools. This paper briefs the approach, explains contexts and findings of the case studies, and discusses overall case study results.

Keywords: Software measurement, quantitative management, statistical process control

Web Cost Estimation Models using Radial Basis Function

Neural Networks

Ali Idri[1], Emilia Mendes[2] , Abdelali ZAKRANI[1]

[1]Department of Software Engineering, ENSIAS, Mohamed V University
Rabat, Morocco
idri@ensias.ma

[2]Computer Science Department, The University of Auckland, Private Bag 92019
Auckland, New Zealand
emilia@cs.auckland.ac.nz

Abstract. Radial Basis Function Neural Networks (RBFN) have been recently studied due to their qualification as an universal function approximation. This paper investigates the use of RBF neural networks for web cost estimation. The focus of this study is on the design of these networks, especially their middle layer composed of receptive fields, using two clustering techniques: the C-means and the APC-III algorithms. A comparison between a RBFN using C-means and a RBFN using APCIII, in terms of estimates accuracy, is hence presented. This study uses data on Web applications from the Tukutuku database.

Keywords: Web effort estimation, Neural Networks, predictive accuracy, Radial basis function neural networks

Comparing Machine-learning Techniques for Web Cost Estimation

Emilia Mendes[1], Sergio Di Martino[2], Filomena Ferrucci[2], Carmine Gravino[2], Steve Counsell[3]

[1] The University of Auckland, Private Bag 92019
Auckland, New Zealand, 0064 9 3737599 ext. 86137
emilia@cs.auckland.ac.nz

[2]The University of Salerno, Via Ponte Don Melillo, I-84084 Fisciano (SA)
Italy, 0039 089963374
{sdimartino,fferrucci,gravino}@unisa.it

[3]Brunel University, Uxbridge, UB8 3PH
UK, +44 (0)1895 266740
steve.counsell@brunel.ac.uk

Abstract. The objective of this paper is to partially replicate two previous studies that compared techniques for Web cost estimation in order to identify which machine-learning technique provides best prediction accuracy. We employed the two machine-learning techniques that were mutual to the two studies being replicated, namely Case-Based Reasoning (CBR) and Classification & Regression Trees (CART). We used a cross-company data set of 150 Web projects from the Tukutuku data set. This is the first time such large number of Web projects is used to compare cost estimation techniques. Results showed that all techniques presented similar predictions, and these predictions were significantly better than those using the mean effort. Thus, all the techniques can be employed for Web cost estimation.

Keywords: Cost estimation, effort estimation, case-based reasoning, regression tree, Web projects, Web applications

Towards an Early Usability Evaluation for Web Applications

Jose Ignacio Panach, Nelly Condori-Fernández, Francisco Valverde, Nathalie Aquino, Oscar Pastor

Department of Information Systems and Computation
Technical University of Valencia
Camino de Vera s/n, 46022 Valencia, Spain.
Phone: +34 96 387 7000, Fax: +34 96 3877359

{jpanach, nelly, fvalverde, naquino, opastor}@dsic.upv.es

Abstract. In the Human-Computer Interaction (HCI) community, the usual technique for measuring usability is through a user test. The disadvantage of this approach is that the system must be implemented before carrying out the test. In this paper, a Usability Model is proposed to evaluate early usability from conceptual schemas, allowing the incorporation of improvements before the automatic generation of web applications. We evaluate the usability of artefacts modelled with OOWS, a development method based on model transformations. In addition, two case studies are presented for verifying the instruments used to evaluate our early Usability Model.

Comparing Cross- vs. Within-Company Effort Estimation Models Using Interval Estimates

Stamatia Bibi[1], Nikolaos Mittas[1], Lefteris Angelis[1], Ioannis Stamelos[1], and Emilia Mendes[2]

[1] Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
{sbibi, nmittas, lef, stamelos}@csd.auth.gr

[2] Computer Science Department, The University of Auckland, Private Bag 92019 Auckland, New Zealand
emilia@cs.auckland.ac.nz

Abstract. This paper investigates whether effort predictions for projects from a single company that were obtained using a cross-company (CC) training set can be as accurate as effort predictions obtained using a within-company (WC) training set. We employed five different cost estimation techniques, two providing point estimates (estimation by analogy and stepwise regression) and three providing predefined interval estimates (ordinal regression, classification and regression trees and Bayesian networks). For the development and evaluation of both cross and within company models ISBSG release 9 was utilized. Our results showed no significant differences between CC and WC-based predictions, for all the cost estimation techniques, after comparing the medians of the absolute errors. Other accuracy metrics were also considered, providing in general similar results.

Keywords: Software effort estimation, predefined intervals, cross company estimation models, within company estimation models, regression models, CART, estimation by analogy, Bayesian networks

Industry Case Studies of Estimation Models Using Fuzzy Sets

Francisco Valdés Souto [1], Alain Abran [2]

[1] Mexican Central Bank,Tolona 371, Col Roma Sur, 06760 Mexico D.F.
fvaldes@banxico.org.mx, fvaldes@nrim.com.mx

[2] École de technologie supérieure – Université du Québec
Alain.Abran@etsmtl.ca

Abstract. Today, the duration factor in a software project is as critical as it is strategic, since even a slight delay can lead to missing a market opportunity or to generating significant losses. This paper presents two industry case studies on the use of an estimation model using, as input parameters, fuzzy sets of data proposed by practitioners estimating various project attributes.

Keywords: Estimation Model, Time Estimation, Fuzzy Sets

Analysis of Heterogeneous Software Projects Databases

M. Garre, J.J. Cuadrado, M.A. Sicilia, and I. S_anchez

University of Alcal_a, Ctra. Barcelona km 33.6 - 28871,
Alcal_a de Henares, Madrid, Spain

fmiguel.garre, jjcg, msicilia, isaac.sanchezg@uah.es

Abstract. Parametric software e_ort estimation models rely on the availability of historical project databases from which estimation models are derived. In the case of large project databases with data coming from heterogeneous sources, a single mathematical model cannot properly capture the diverse nature of the projects under consideration. Clustering algorithms can be used to segment the project database, obtaining several segmented models. In this paper, a new tool is presented, Recursive Clustering Tool, which implements the EM algorithm to cluster the projects, and allows use di_erent regression curves to _t the di_erent segmented models. This di_erent approaches will be compared to each other and with respect to the parametric model that is not segmented. The results allows conclude that depending on the arrangement and characteristics of the given clusters, one regression approach or another must be used, and in general, the segmented model improve the unsegmented one.

Keywords. Software Engineering, E_ort estimation, Segmented parametric model, Recursive Clustering Tool (RCT), Clustering, EM algorithm

Using the PR*i*M method to Evaluate Requirements Models with COSMIC-FFP

Gemma Grau, Xavier Franch

Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3, Barcelona E-08034, Spain

{ggrau, franch}@lsi.upc.edu

Abstract. The COSMIC-FFP is a standard method that has been proven effective for measuring the functional size of business applications and realtime software systems from their functional user requirements specification. Despite of this, the methods based on COSMIC-FFP usually require a mapping between the concepts in the requirements specification and their own terms and do not take into account non-functional requirements. On the other hand, PR*i*M is a method that aims at assessing non-functional properties at the early stages of the development process. PR*i*M uses the *i** framework to model the functional and non-functional requirements in terms of actors and dependencies among them. In this paper we present how the *i** constructs proposed in PR*i*M can be adapted to measure the functional size using COSMIC-FFP and, as PR*i*M works with requirements and allows the evaluation of non-functional properties, there is a remarkable benefit when using both methods altogether.

An Empirical Study of Process Management and Metrics based on In-process Measurements of a Standardized Requirements Definition Phase

Yoshiki Mitani[1,2], Tomoko Matsumura[2], Mike Barker[2], Seishiro Tsuruho[1,3], Katsuro Inoue[4], Ken-Ichi Matsumoto[2]

[1] Information Technology Promotion Agency, Japan (IPA)

[2] Nara Institute of Science and Technology (NAIST)

[3] Kochi University of Technology,4 Osaka University

{ymitani|tomoko-m}@empirical.jp, mbarker@MIT.EDU, tsuruho@ipa.go.jp, inoue@ist.osaka-u.ac.jp, matumoto@is.naist.jp

Abstract. This paper focuses on in-process project measurement in the requirements definition phase based on progress with standardization of this phase. The authors have verified the utility of in-process project measurement in a real mid-scale multi-vendor distributed project. This trial was successful, but limited to a part of the total development process. The project measurement target was limited to later processes such as the coding and testing phases where the output products were easy to acquire. The requirements definition phase where process and product were not standardized was difficult to measure. However, a newly provided governmental process guideline standardizes the process and product for the requirements definition phase, and the authors had an opportunity to measure such a requirements definition effort. This paper presents an empirical study of in-process project measurement in the standardized requirements definition phase, verifies the

usefulness of this measurement for project management, and reveals the possibility of creating a new software metrics field using these measurements.

Keywords: Empirical software engineering, Software process measurement, In-process measurement, Enterprise Architecture, Requirements definition phase measurement

A case study using the COSMIC-FFP Measurement Method for Assessing Real-Time System Specifications

Manar Abu Talib[1], Adel Khelifi[2], Alain Abran[3] and Olga Ormandjieva[4]

[1]Zayed University, P.O. Box 4783, Abu Dhabi, UAE

[2]Al Hosn University, P.O. Box: 38772, Abu Dhabi, UAE

[3]École de technologie supérieure, Université du Québec, Montreal, Canada

[4]Concordia University, 1455 de Maisonneuve Blvd. W., Montreal, Canada

manar.talib@zu.ac.ae, a.khelifi@alhosnu.ae, aabran@ele.etsmtl.ca, ormandj@cse.concordia.ca

Abstract. The success of a system development project largely depends on the nonambiguity of its system-level requirements specification document, where the requirements are described at the system level rather than at the software and hardware level. There may be missing details about the allocation of functions between hardware and software, both for the developers who will have to implement such requirements later on, and for the software measurers who have to immediately attempt to measure the software functional size of such requirements.. The result of different interpretations of the specification problem would lead to different software being built, and of different functional size. The research described in this paper is concerned with the challenges inherent in understanding the initial system requirements in textual form and assessing the codesign decisions using the functional size measurement. This paper aimed at understanding the applicability of the COSMIC-FFP functional size measurement method in assessing the hardware-software requirements allocation, and illustrates the approach on a Steam Boiler Controller case study.

Keywords: COSMIC-FFP, ISO 19761, system-level requirements specification, codesign functional size measurement

Speeding up the estimation process with the Estimating Wizard

Harold van Heeringen

Sogeti Nederland B.V., Lange Dreef 17, 4131 NJ Vianen, the Netherlands

harold.van.heeringen@sogeti.nl

Abstract. Nowadays, the division Managed Delivery of Sogeti Nederland gets more and more questions from clients like: “What is your productivity rate for Java projects?”, “What is your duration for building an application of 1000 function points?” and “What is your price per function point for a .Net project?” Literature shows us however, that there is no good answer to these kinds of questions. Putnam shows us that the amount of effort needed highly

depends on the duration chosen. Other factors that influence the answer to these questions might be: size, complexity and the amount of work that is being carried out on an offshore location (like India). It is therefore necessary to consider all the relevant factors when preparing a project estimation. However, if this has to be done on an ad hoc basis (whenever a client asks), it will take a lot of time to analyze the right projects. To make things faster and easier, Sogeti has developed a tool to estimate projects and to answer questions like the questions mentioned above.

Keywords: Project Estimation, Bid Management, Estimation, Uncertainty, Functional Size Measurement, Estimation Tool, Work Breakdown Structure

Scope Management – How Uncertain is Your Certainty

Frank Vogelezang

Sogeti Nederland B.V., Lange Dreef 17, 4131 NJ Vianen, the Netherlands

frank.vogelezang@sogeti.nl

Abstract. Scope management is a project governance approach that makes sensible use of the right software estimation techniques at the right moment in the life cycle of a manageable part of a software project. A lot of software projects make commitments where the right estimation techniques will show that the only certainty those commitments can give is that they can bring this project into the danger zone. With scope management project managers and their project boards can make founded decisions about their software projects, because scope management provides them with solid information about how (un)certain it is that a project will deliver the expected results. The scope manager acts both as a quantity surveyor and an auditor for the project. This article describes the experience of Sogeti with scope management and gives some examples from real projects.

Keywords: Scope Management, Project Management, Estimation, Uncertainty, Functional Size Measurement, Project Governance

Impact of Improvement Actions on Help Desk Costs

Magg. Andrea Raffaelli[1]; Loredana Mancini[2]

[1]Carabinieri,V.le Romania, 45 Rome Italy,

[2]President GUFPI-ISMA (COO Businesse), Via V. Mazzola, 66 Rome Italy

Andrea.Raffaelli@carabinieri.it, Loredana.Mancini@gufpi.org,

Abstract. This paper wants to represent an experience, carried on within an Italian public Government organisation on the analysis of the help desk costs. The analysed help desk, is the internal one of a distributed environment. The help desk acts as the entry point for all the users questions and needs on the usage of the internal systems, the number of users is 30.000 and it has 120/150 calls a week, the users are distributed all over Italy, with different levels of knowledge about the information technology, but the average is quite low. The systems is centralised and there are no major local problems on the application and on the management of the infrastructure, the problems are mainly connected with user local services. The organisation worked with GUFPI-ISMA in order to have input on the methods to measure and analyse the number of calls to the internal help-desk, and the associated costs. The analysis wanted to check the impact on the help-desk costs (i.e. number of calls,

time spent in problem solving, downtime of the systems, time to recover, etc.) of the different improvement actions carried on within the hardware and software systems. In particular each improvement action was planned and deployed one by one, in order to be able to check the costs saving impact. Actions were both on technical aspects as well as organisational ones.

Keywords: Metrics, Help Desk, Improvement

Preliminary Results in a Multi-site Empirical Study on Cross-organizational ERP Size and Effort Estimation

Maya Daneva

Dept of Computer Science, University of Twente, Enschede, The Netherlands

m.daneva@utwente.nl

Abstract. This paper reports on initial findings in an empirical study carried out with representatives of two ERP vendors, six ERP adopting organizations, four ERP implementation consulting companies, and two ERP research and advisory services firms. Our study's goal was to gain understanding of the state-of-the practice in size and effort estimation of cross-organizational ERP projects. Based on key size and effort estimation challenges identified in a previously published literature survey, we explored some difficulties, fallacies and pitfalls these organizations face. We focused on collecting empirical evidence from the participating ERP market players to assess specific facts about the state-of-the-art ERP size and effort estimation practices. Our study adopted a qualitative research method based on an asynchronous online focus group.

Implementing Software Measurement Programs in Non mature Small Settings

María Díaz-Ley[1], *Félix García*[2], *Mario Piattini*[2]

[1] Sistemas Técnicos de Loterías del Estado (STL)
Gaming Systems Development Department 28234 Madrid, Spain
Maria.diaz@stl.es

[2] ALARCOS Research Group
Information Systems and Technologies Department
INDRA-UCLM Research and Development Institute
University of Castilla-La Mancha,
13071 Ciudad Real, Spain
{Felix.Garcia, Mario.Piattini}@uclm.es

Abstract. Although measurement has been successfully applied in various ar-eas, it has proved to be a complex and difficult undertaking in the field of soft-ware and especially in the context of small and medium enterprises (SMEs). Measurement programs in SMEs with a low maturity level should be tailored to their limitations (limited resources, experts, etc.) in order to carry out the meas-urement initiatives efficiently. In this paper we report the method, principles and practices followed in our experience for defining and implementing a measurement program in the development department of Sistemas Técnicos de Loterías del Estado (STL). We also show the characteristics of this company which guided our approach

towards tackling the problem, and the resulting software measurement program. As a result of the application of certain prac-tices in the company, some significant benefits were obtained, which could be replicated in similar environments.

Keywords: Software measurement program, small and medium settings, case study

Software Metrics and Evaluation of Their Usefulness in Finnish Software Companies

Jari Soini[1], Vesa Tenhunen[2] and Markku Tukiainen[2]

[1] Tampere University of Technology – Pori
jari.o.soini@tut.fi

[2] University of Joensuu

vesa.tenhunen@cs.joensuu.fi, markku.tukiainen@joensuu.fi

Abstract. In this study we examine a set of process and product quality metrics used in Finnish software industry. We also look at software measurement practices and experiences related to the software industry in Finland. The results show commonly measured aspects of software production and the emphasis of measurement practices. In addition, we present the users' viewpoints of the characteristics such as usability and effort of the current metrics in use. These results and their interpretation give us unique empirical information in the field of software measurement in practice.

Keywords: Software engineering, software measurement, software metrics

Estimating the Test Volume and Effort for Testing and Verification & Validation

Alain Abran[1], Juan Garbajosa[2], Laila Cheikhi[1]

[1] Ecole de technologie supérieure, Universtité du Québec, Canada;

[2] Universidad Politécnica de Madrid, Spain

alain.abran@etsmtl.ca, jgs@eui.upm.es, laila.cheikhi.1@ens.etsmtl.ca

Abstract. This paper discusses an approach to estimating the test volume and related effort required to perform both testing and verification and validation activities on software projects. The approach uses measures of functional requirements as a basis for quantitatively estimating this volume, followed by effort estimation models based on functional size to prepare a first-degree order, and, finally, an adjustment based on the assessment of non-functional requirements. This estimation approach is illustrated using data from the ISBSG 2006 repository and the set non-functional requirements as categorized in the European standard for the aerospace industry: ECSS-E-40, Part 1B.

Keywords: Estimation, Verification & Validation, ISBSG, Test Volume, Testing effort, Non-functional Requirements

Do Different Functionality Types Affect the Relationship between Software Functional Size and Effort?

Cigdem Gencel[1], Luigi Buglione[2]

[1]Middle East Technical University, Ankara, Turkey
cgencel@metu.edu.tr

[2] École de Technologie Supérieure (ETS) / Atos Origin
Luigi.Buglione@atosorigin.com

Abstract. Effort estimation is a significant practical problem in software engineering, and various cost drivers, including software size, which might have an impact on it have been explored. In many of these studies, total software size (measured in either lines of code or functional size units) is the primary input. However, the relationship between effort and the components of functional size has not yet been fully analyzed. This study explores whether effort estimation models based on the functional size components, that is, Base Functional Component types, rather than those based on a single total value, would improve estimation models. For this empirical study, the project data in the International Software Benchmarking Standards Group (ISBSG) Release 10 dataset, which were sized by the COSMIC FFP method, are used.

Keywords: Functional Size Measurement, Effort Estimation, COSMIC-FFP, Base Functional Component, International Software Benchmarking Standards Group (ISBSG)

Non-Functional Requirements: Size Measurement and Testing with COSMIC-FFP

M. Kassab[1], O. Ormandjieva[2], M. Daneva[3], A. Abran[4]

[1,2]**{moh_kass, ormandj}@cse.concordia.ca**,
[3]**m.daneva@utwente.nl**, [4]**Alain.Abran@etsmtl.ca**

Abstract. The non-functional requirements (NFRs) of software systems are well known to add a degree of uncertainty to the process of estimating the cost of any project. This paper contributes to the achievement of more precise project size measurement through incorporating NFRs into the functional size quantification process. We report on an initial solution proposed to deal with the problem of quantitatively assessing the NFR modeling process early in the project, and of generating test cases for NFR verification purposes. The NFR framework has been chosen for the integration of NFRs into the requirements modeling process and for their quantitative assessment. Our proposal is based on the functional size measurement method, COSMIC-FFP, adopted in 2003 as the ISO/IEC 19761 standard. Also in this paper, we extend the use of COSMICFFP for NFR testing purposes. This is an essential step for improving NFR development and testing effort estimates, and consequently for managing the scope of NFRs. We discuss the merits of the proposed approach and the open questions related to its design.

A method to measure software adoption in organizations: a preliminary study

Bruno Rossi, Barbara Russo, Giancarlo Succi

Free University of Bolzano-Bozen

{Bruno.Rossi,Barbara.Russo,Giancarlo.Succi}@unibz.it

Abstract. The decision about the adoption of Free/Libre/Open Source Software (FLOSS) is a key issue in Small and Medium Enterprises (SMEs). Indeed, very often such organizations don't have the resources needed to fully evaluate the migration from existing legacy systems. To help the decision process of these organizations, we propose a preliminary study about an instrument based on the analysis of files' generation of targeted data standards. We model the file generation process as a self-reinforcing mechanism through the usage of urn models.

By applying the instrument to a large dataset in the office automation field, we found a first confirmation about the importance of network externalities as reported by theory and the importance of past historical file generation for the subsequent file generation process.

Keywords: FLOSS, software adoption, measurement, data standards, path dependent process

Using Controlled Experiments for Validating UML Statechart Diagrams Measures

José A. Cruz-Lemus, Marcela Genero and Mario Piattini

ALARCOS Research Group
Department of Information Technologies and Systems
Indra-UCLM Research and Development Institute
University of Castilla-La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain

{JoseAntonio.Cruz, Marcela.Genero, Mario.Piattini}@uclm.es

Abstract. In this work, we present the main conclusions obtained from the definition and validation of a set of measures for UML statechart diagrams, in a methodological way. The main focus is the empirical validation of the measures as early understandability indicators.

Experiences on Using Software Experiments in the Validation of Industrial Research Questions

Dominik Gessenharter[1], Alexander-Marc Merten[2], Alexander Raschke[1] and Nicolas Fernando Porta[2]

[1] Ulm University, Institute for Software Engineering
and Compiler Construction, D-89069 Ulm, Germany
Dominik.Gessenharter@uni-ulm.de, Alexander.Raschke@uni-ulm.de

[2] DaimlerChrysler, Group Research & Advanced Engineering, Ulm
alexander-marc.merten@daimlerchrysler.com,
nicolas.porta@daimlerchrysler.com

Abstract. Experimentation in software engineering is difficult. One reason is the large number of context variables [1] and the impracticality of experiments in an industrial setting. Considering the budgets of comprehensive projects, it is apparent that a company cannot double its effort executing a project twice, in order to compare two different approaches concerning process or method improvement. Performing experiments on the basis of small projects seldom offers solutions valid for industrial settings. Our commendation is a cooperation between industry and academic education. This approach offers multiple advantages. In this paper, we outline our experiences in experimental software engineering gained in about 20 experiments over the past 10 years by the Ulm University cooperating with DaimlerChrysler Group Research & Advanced Engineering, Ulm. Additionally we provide an insight into a current experiment and present our approach to experimental software engineering in further detail.

An Infrastructure for Empirically-based Software Engineering Technology Selection

Andreas Jedlitschka

Fraunhofer Institute Experimental Software Engineering
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

Andreas.Jedlitschka@iese.fraunhofer.de

Abstract. In today's software development organizations, methods and tools are employed that frequently lack sufficient evidence regarding their suitability, limits, qualities, costs, and associated risks. The need to select the best-suited method, technique or tool in a given business context is becoming more and more important. From a business perspective the trade-off between time-tomarket, quality, and cost is a crucial factor for the decision process. While new findings from research await their transfer into industrial practice, systematic assessment, selection and infusion of these findings with regard to business objectives and context is lacking. This paper presents ongoing research towards the development of a decision support system that aims at improving software engineering technology selection by software managers with regard to business goals. The focus of this paper is on presenting the problems at hand, the idea for a comprehensive decision support, and discussing how the resulting framework could be enacted.

Why software projects fail? Empirical evidence and relevant metrics

Maurizio Morisio, Evgenia Egorova, Marco Torchiano

Politecnico di Torino, Corso Duca degli Abruzzi 24,
10129 Torino, Italy

{maurizio.morisio, eugenia.egorova, marco.torchiano}@polito.it

Abstract. In this paper we present an empirical study on factors that cause the success or failure of a software project. The study deals with the definition and validation of metrics and indicators to make more objective the meaning of success (and conversely failure) of a project, and correlates characteristics of a project (such as experience of the staff and the project manager, techniques used for requirement engineering, and so on) with its result. The study, performed on a number of Italian companies, can be easily replicated in other countries or within a company, in order to assess its software processes and steer its process improvement initiatives.

A Model of Costs and Benefits of Reviews

Tilmann Hampp

Universität Stuttgart, Institut für Softwaretechnologie,
Universitätsstr. 38, 70569 Stuttgart, Germany

tilmann.hampp@informatik.uni-stuttgart.de

Abstract. The benefits of reviews are well-known in software quality assurance. Nevertheless, in industry they are often used in a truncated or shortened way, if at all. The costs are obvious and arise immediately, whereas the benefit of improved quality is hard to measure, and is achieved only in the long run. As decisions often focus on reducing development time and effort, we present an estimation model for costs and benefits of specification and design reviews. The estimation results are discussed in this paper. For in-process benefits, benefits in maintenance and in usage, the results match reported experiences.

How to measure Agile Software Development

Martin Kunz[1], Reiner R. Dumke[1], René Braungarten[2], Andreas Schmietendorf[3]

[1]Software Engineering Group
University of Magdeburg, Germany
{makunz, dumke}@ivs.cs.uni-magdeburg.de

[2] Bosch Rexroth AG, Germany
Rene.Braungarten@boschrexroth.de

[3] Berlin School of Economics, Germany
schmiete@fhw-berlin.de

Abstract. Agile Software Development Methods are nowadays wide spread and accepted. From the Software Measurement point-of-view not all metrics and methods from conventional lifecycle models can be used without adoption. Therefore this paper describes distinct metrics and their implementation into a measurement tool for quality management in agile software development. Taking a closer look to agile methods we observe that some of metrics and the tool itself can be used for measurement in traditional software development as well.

Keywords: Software Measurement, Agile Software Development, Metrics

Towards a Two-dimensional Approach To track Software Degradation

Miguel Lopez[1], Naji Habra[2]

[1] avenue van Volxem 140, B-1190 Bruxelles Belgium
Miguel.lopez@algorismi.com

[2] University of Namur – PRECISE Research Centre
rue de Bruxelles 61, B-5000 Namur Belgium
nha@info.fundp.ac.be

Abstract. A main assumption behind the majority of the refactoring works states roughly that normal evolution of any software deteriorates its structure in such a way that, at a certain point, a refactoring becomes interesting/necessary to recover quality structure.
Of course, this general idea necessitates several precisions about the software itself, the evolution process, the structural properties being deteriorated, etc. According to this fact, an important question is how can we track this degradation in order to better apply refactoring recommendations?

The current paper aims at answering this question by suggesting a twodimensional approach that allows highlighting the level of degradation of a source code within a given period. The paper also provides an empirical case study where we experiment this approach on 8 PHP applications.

The direct finding of this first case study is the possibility of monitoring the evolution of the source code degradation in order to make decision about its refactoring.

Our DASMA Software Metrik Kongress (MetriKon 2007) took place in Kaiserslautern, Germany in November 2007. The following report gives an overview about the presented papers. Furthermore, the papers are published in the following Shaker book (ISBN 978-3-8322-6703-2):

Magdeburger Schriften zum Empirischen Software Engineering

Hrsg: Günter Büren, Büren & Partner Software-Design, Nürnberg
Manfred Bundschuh, AXA AG, Köln, Vorsitzender der DASMA e.V.
Prof. Dr.-Ing. habil. Reiner R. Dumke, Universität Magdeburg

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

Fakultät für Informatik
Institut für Verteilte Systeme
Arbeitsgruppe Softwaretechnik

MetriKon 2007

Praxis der Software-Messung

Tagungsband des DASMA Software Metrik Kongresses

15.-16. November 2007, Kaiserslautern

DASMA 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

Measuring Software Productivity

Barbara Kitchenham

School of Computing and Mathematics
Keel University
Stafforshire ST5 5BG, UK

b.a.kitchenham@cs.keele.ac.uk

Abstract. This Paper discusses a number of problems that occur if productivity is measured as a simple ratio of size and effort. In particular, the simple productivity ratio:

- Assumes that there is a single measure of size.
- Assumes there are no economies or diseconomies of scal.
- Is usually not Normally distributed causing analysis problems.

I introduce two approaches that address all these issues. Both approaches rely on the equivalence between effort-size regression relationships and productivity measures after normalising transformations are applied. One approach leads to a productivity index based on the ratio of estimated effort to actual effort which has an expected value of one with values greater than on eimplying better than expected productivity and values less than one implying worse than expected productivity. The other approach leads to standardized residuals that can be used to construct run charts which plot productivity over time. The theoretical basis of each approach is explained and demonstrated with examples based on a Web application database and a dataset provide by IBM Australia.

Keywords: software productivity measurement; effort estimation; DEA

Konzepte zur Kosten-Nutzen-Messung prozessverbessernder Maßnahmen in der Automotive E/E Entwicklung am Beispiel User Defined Workflow

Nicolas Fernando Porta, Alexander-Marc Merten

DaimlerChrysler Group Research & Advanced Engineering

Nicolas.Porta | Alexander-Marc.Merten@DaimlerChrysler.com

Zusammenfassung: Dieses Papier untersucht Konzepte zur Nutzenmessung prozessverbessernder Maßnahmen zur Anwendung in der Elektrik/Elektronik Entwicklung eines OEMs. Zu diesem Zweck werden ausgewählte etablierte Modelle analysiert und deren Tauglichkeit im gegebenen Kontext überprüft. Anhand eines Experiments im Bereich User Defined Workflow wird gezeigt, wie durch das Aufstellen von Wirkketten und die Dokumentation der dazu getroffenen Annahmen Metriken abgeleitet werden können, um den Nutzen der Maßnahme zu bewerten. Durch zukünftige Experimente und Fallstudien soll ein umfassendes Kosten-Nutzen-Modell erstellt werden, das die Anforderungen aus dem betrachteten Umfeld erfüllt.

Schlüsselbegriffe: Kosten-Nutzen, Quantitatives Projektmanagement, Automotive, Wirkketten, Metriken, User Defined Workflow

Prozessverbesserung über Fehlerstrommessung bei einem mittelständischen Unternehmen

Michael Kläs[1], Marcus Ciolkowski[1], Torsten Schulz[2], Jürgen Goebbels[2]

[1]Fraunhofer IESE, Fraunhofer-Platz 1
67663 Kaiserslautern

{michael.klaes, marcus.ciolkowski}@iese.fraunhofer.de

[2]IBS AG, Rathausstraße 56
56203 Höhr-Grenzhausen

{torsten.schulz, juergen.goebbels}@ibs-ag.de

Zusammenfassung: Eine konstant hohe Produktqualität ist insbesondere für kleine und mittelständische Unternehmen (KMUs) wichtig, um Kundenzufriedenheit gewährleisten zu können. Dabei ist ein guter Entwicklungs- und Qualitätssicherungsprozess maßgeblich für die Erreichung einer hohen Softwarequalität verantwortlich. Die systematische Fehlermessung nimmt dabei eine Schlüsselrolle ein, da nur so empirisch zu ermitteln ist, wie effektiv bestehende Qualitätssicherungs-Prozesse sind, welche Fehlerarten durch welche Prozesse gefunden werden und welches Verbesserungspotential gegeben ist.
In diesem Beitrag werden das Vorgehen und die Erfahrungen bei der Definition und Einführung eines Messprogramms zur Erfassung eines Fehlerstrommodells (FSM) bei einem mittelständischen Unternehmen beschrieben. Das Messprogramm ist eingeführt und wird aktiv betrieben. Dies ist unseres Wissens die erste Dokumentation der Einführung eines FSM bei einer KMU. Durch die gewonnenen Erfahrungen konnte der Definitions- und Einführungsprozess für KMUs verfeinert werden.

Trotz des frühen Zeitpunkts und des durch die statistische Evaluierung aufgezeigten Verbesserungspotentials hinsichtlich der Qualität des Schemas und der Fehlererfassung können schon erste interessante Ergebnisse hinsichtlich der Fehlerkorrekturkosten unterschiedlicher Fehlerarten und Entdeckungszeitpunkte präsentiert werden.

Schlüsselbegriffe: Fehlerstrom, Fehlerklassifikation, Qualitätsmanagement, KMU

Experience-Based Software Measurement and Evaluation Considering Paradigm Evolution

Reiner Dumke, René Braungarten[1], Steffen Mencke, Karsten Richter[2], Hashem Yazbek

Otto-von-Guericke-University of Magdeburg, Germany

[1]Bosch Rexroth Lohr, [2]Bosch Stuttgart,

(dumke,mencke,hashem)@ivs.cs.uni-magdeburg.de

Abstract. This paper introduces an approach for software measurement and evaluation based on the experience of the previous software technology (paradigm). Considering the basic software paradigm we demonstrate the possibilities and meaningfulness by using software measurement results (as experience) for software development in new/inexperienced technologies such as service-oriented architectures, feature programming and agent-based system development.

We show some examples of methods of software measurement that are based on existing experiences for appropriate technologies such as object-oriented metrics thresholds for agent-based systems or function point estimation for service-oriented architectures. Based on simple examples of using existing experience we describe helpful rules for deriving one's own appropriate estimation and evaluation background in the system development in modern technologies.

Keywords: Software development paradigms, metrics, measurement, evaluation

Case Study for using model based strategic Planning of IV&V

Haruka Nakao[I], Shigeo Yoshikawa[I], Daniel Port[II], Yuko Miyamoto[III]

[I]Japan Manned Space Systems (JAMSS)
{haruka, yoshikawa.shigeo}@jamss.co.jp

[II]Department of Information Technology Management University of Hawaii
dport@hawaii.edu

[III]Japan Aerospace Exploration Agency (JAXA)
Yuko Miyamoto@jaxa.jp

Summery. This research presents a grounded empirical case study to objectively compare, through consideration of risk reduction and cost, plans generated by the strategic model to those created by experts. The value of this is that where they are the same, we will have an objective rationalization of that part of the plan. Where they differ, we investigate why and perhaps discover expert bias, error, or a limitation/error in the model. The goal is to increase confidence in our model/expert based planning method. The research is straightforward – use the model to generate plans for several past projects, compare risk reduction and cost profiles with the actual plans implemented, look for significant differences and question experts on these differences (e.g. "Why did you choose activity X over Y?"). In this sense this work is a bit unorthodox in that it is not an empirical study in that it a structured investigation and analysis rather than a series of experiments. However we believe that this approach is appropriate for our purposes and through this research, we found that the model generated plan is a "synthesis" plan that contains the experts' plan. In addition, we found that experts had a noticeable bias their activity selections, but still tended to follow the most cost-effective route.

Keywords: Spacecraft on board software, Independent Verification and Validation (IV&V), model based planning, Quantification, system engineering

Das neue SoftCalc

ein Tool für die differenzierte Kalkulation unterschiedlicher Projektarten

Harry M. Sneed

ANECON GmbH, Wien
Lehrbeauftragter für Software Engineering
an den Universitäten Regensburg, Passau & Budapest

Abstrakt: Der folgende Beitrag beschreibt ein Werkzeug für die automatisierte Schätzung von Software-Projekten unterschiedlicher Art - Prototypprojekte, Entwicklungsprojekte, Evolutionsprojekte, Sanierungsprojekte, Migrationsprojekte, Integrationsprojekte und auch Testprojekte. Geschätzt wird der Aufwand in Personenmonaten und Dauer in Kalendermonaten sowie die zu erwartende Fehleranzahl – Restfehlerwahrscheinlichkeit – und der Aufwand für die Systemwartung. Die Schätzung basiert auf der geplanten oder real existierenden Systemgröße und –qualität, gekoppelt mit der Erfahrung aus bisherigen Projekten gleicher Art. Das Werkzeug bietet die Möglichkeit, mit 8 verschiedenen Schätzmethoden zu arbeiten – COCOMO I, COCOMO II, Function-Point, Data-Point, Object-Point, Use Case-Point, Test-Point und Error Projection. Es besitzt eine umfassende graphische Oberfläche und eine relationale Datenbank mit 12 Tabellen für jedes Projekt.

Keywords: Aufwandsschätzung, Projektkalkulation, Fehlerprojektion, Erfahrungsdatenbank, Risikoanalyse, Einflussfaktoren, Software-Produktivität

Use Case Points Aufwandsschätzung auf Basis unterschiedlicher Spezifikations-Formate

Stephan Frohnhoff, Karsten Kehler

sd&m AG, Berliner Str. 76, D-63065 Offenbach

frohnhoff@sdm.de

Zusammenfassung: Die Use Case Point Methode (UCP-Methode) erlaubt in Software-Entwicklungsprojekten zu einem frühen Zeitpunkt eine einfache Schätzung von zu erwartenden Aufwänden. Basis für eine solche Schätzung sind in der industriellen Praxis häufig Grob-Spezifikationen unterschiedlichen Formats und unterschiedlicher Granularität.

Entscheidend für den Erfolg der UCP-Methode und die Vergleichbarkeit der Ergebnisse ist vor allem, ob und wie es gelingt, die vorliegende Spezifikation auf Use Cases im Sinne der UCP-Methode abzubilden. Hierzu fasst der Artikel einen Vorschlag für einen Leitfaden zusammen, der aus der Spezifikations-Praxis des Softwarehauses sd&m abgeleitet wurde.

Schlüsselbegriffe: Projektschätzung, Top-Down-Schätzung, Aufwandsschätzung, Use Case Points, UCP, Spezifikation

Structuring Test Process Metrics

Ayaz Farooq[1], Reiner R. Dumke[1], Heike Hegewald[2], Cornelius Wille[3]

[1]Institute for Distributed Systems, University of Magdeburg, Germany

[2]CSC Wiesbaden, Germany

[3]University of Applied Sciences, Bingen, Germany

{farooq, dumke}@ivs.cs.uni-magdeburg.de
wille@fh-bingen.de

Abstract. Ability to make informed decisions about progress, effectiveness, and efficiency of the testing process, test adequacy, and fulfillment of testing goals is a prime concern of the test managers. Measurement-based evaluation provides an effective way to answer these questions. However, measurement and evaluation of the test process poses many open problems. What specific kinds of metrics and measurement-based experiences are available and/or needed that aid this process? Which of the existing metrics can be matched to the test process's typical quality rules and improvement goals? How these metrics can be organized to answer different concerns of the test managers? We present a structuring of the test process metrics which can help identify relationships, classification, organization, and application of these measurements to improve the test process. Our purpose is not to define new metrics but to focus our attention to the ways by which existing measurements and experiences can support improvement of the test process.

Keywords: Software metrics, verification & validation, test process improvement

Ein KPI-Konzept für die Steuerung eines QA-Outsourcing

Thomas Fenske, Dr. Ivan Morschel

sd&m AG – Berlin, TÜV Rheinland - Hamburg

Thomas.Fenske@sdm.de, Ivan.Morschel@de.tuv.com

Zusammenfassung: Das hier vorgestellte Konzept umfasst die Bestandsaufnahme, Definition und Messung mithilfe von Key-Performance-Indikatoren (KPIs), wodurch eine quantitative Erhebung der Servicequalität mit einer möglichst geringen Menge von Metriken zu realisieren ist. Eine der großen Herausforderungen bestand darin, aussagekräftige Kennzahlen zu identifizieren, die die unterschiedlichen Aspekte der Software-Entwicklung abdecken sowie eine Extrapolation auf das künftige Governance Model zwischen AOL, HanseNet und TÜV Rheinland ermöglichen.

Die erzielten Ergebnisse umfassen die Definition von Serviceprozessen auf der inhaltlichen Basis eines existenten Servicekatalogs und die Erweiterung um neue Aspekte. Somit decken die aufgestellten KPIs die Bewertung dieser Services in allen wesentlichen Prozessschritten und den Dimensionen Qualität, Zeit und Kosten ab. Die Dimension Qualität sticht dabei besonders hervor. Während die Bewertungskriterien für Zeit und Kosten verhältnismäßig leicht zu finden waren, mussten Qualitätskriterien, deren Aussagekraft über mehrere Services und Projekte vergleichbar sind, iterativ entwickelt werden, ehe ein kompaktes und konsistentes Ergebnis erzielt werden konnte.

Darüber hinaus wurde ein Bewertungssystem aufgebaut, das die Ergebnisse der KPI-Messung über mehrere Ebenen verdichtet. Darauf aufbauend wurde ein Kunden- und QA-Berichtswesen definiert.

Schlüsselbegriffe: KPI-Modell für Software-Entwicklungsprozesse; Messung und Bewertung von Dienstleistungen; Auftraggeber-Auftragnehmer-Beziehung; Outsourcing

Objective-Driven Process Improvement: Experiences with Improving Productivity

Christof Ebert

Vector Consulting Services GmbH

Christof.Ebert@vector-consulting.de

Abstract. In order to achieve a maximum business benefit and get buy-in by various stakeholders,process improvement must be managed as a project with tangible shortterm results. This article will underline how to set up and drive a process improvement program based upon explicit business objectives and how to deliver tangible value. Despite an increasing body of knowledge with improvement frameworks such as CMMI or SPICE, many organizations still struggle in practice. This article goes beyond such method frameworks. Objective-driven process improvement (ODPI) underlines the need to start with clear business objectives and from those derive a specific and tailored approach towards achieving engineering excellence. The E4-measurement process is applied to support objective-driven process improvement and show how to measure throughout an improvement project. Improving productivity and efficiency is selected as a hands-on example how to practically implement objective-driven process improvement.

Keywords: Efficiency, E4-measurement process, improvement project, objective-driven process improvement (ODPI), productivity, process improvement

Bewerten der Produktivität in der Entwicklung – eine Vision oder ein Mysterium?

Melanie Ruhe

SIEMENS AG, CT SE3, Munich, Germany

melanie.ruhe@siemens.com

Zusammenfassung: Die Produktivität ist eine Kennzahl für die Leistungsfähigkeit einer Organisation. Steigende (Arbeits-)Produktivität ist ein Zeichen dafür, dass die Effizienz und Effektivität des Faktors Arbeit verbessert wird [11]. Die bekannte Definition der Messgröße (Relation von Ausbringungsmenge zur Einsatzmenge) ist in der Welt von Produktentwicklungsorganisationen (Software-, Hardware-, System-Produkte) stark umstritten. In Deutschland gibt es derzeit nur ganz wenige Unternehmen, in denen versucht wird, die Produktivität mit Hilfe einer Metrik zu bestimmen. So lange jedoch kein einheitliches, praktikables Messinstrument vorliegt, kann das Thema Produktivitätssteigerung in der Entwicklung nicht sinnvoll angegangen werden.

In der Forschungsabteilung der Siemens AG wird derzeit an einer neuen Vorgehensweise zur Bewertung der Produktivität gearbeitet. Bisherige Studien gehen oft von fixen, typischen Einflussgrößen in Entwicklungsorganisationen aus und versuchen quantitativ die Stärke der Relation zwischen Produktivität und Einflussgröße mit Hilfe von Regressionsmodellen zu

analysieren. Dieses Vorgehen setzt voraus, dass die Produktivität auch maßgeblich von diesen Einflussgrößen bestimmt wird und dass diese von der Organisation beeinflussbar sind.

Wir entfernen uns vom Ansatz eines fixen Modells, das im Zweifelsfalle nicht zu den Gegebenheiten der Organisation passt. Stattdessen erarbeiten wir zusammen mit den Experten der Organisation ein organisationsspezifisches Modell mit den wesentlichen Einflussgrößen der Produktivität, die sowohl beeinflussbar als auch messbar sind. Die in Interviews identifizierten Einflussgrößen werden hinsichtlich Bedeutung und Wirktiefe geordnet und anschließend ein geeignetes Mess- bzw. Bewertungsinstrument für diese Größen definiert. Somit kann man sowohl den aktuellen Status der einzelnen Größen bewerten als auch den relativen Fortschritt aufzeigen. Aufgrund der Erfahrung, dass solche Einflussgrößen oft miteinander in Beziehung stehen, streben wir eine Balance der Top-3 oder Top-5 Einflussgrößen an, d.h. einen ausgewogenen Zustand der wichtigsten Einflussgrößen der Produktivität.

Schlüsselbegriffe: Produktentwicklung, Produktivitätsmessung, Produktivitätsmanagement, Modell

Prozessqualität bei dem Übergang zur serviceorientierten Architektur

Dmytro Rud[1], Andreas Schmietendorf[1,2], Martin Kunz[1], Reiner R. Dumke[1]

[1] Arbeitsgruppe Softwaretechnik, Institut für Verteilte Systeme, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg

{rud, schmiete, makunz, dumke}@ivs.cs.uni-magdeburg.de

[2] Fachhochshule für Wirtschaft Berlin, Fachbereich II – Wirtschaftsinformatik
Neue Bahnhofstr. 11-17, 10245 Berlin

schmiete@fhw-berlin.de

Zusammenfassung: Dieser Beitrag befasst sich mit Reifegradmodellen im Bereich der serviceorientierten Architektur (SOA). Es wird die Frage diskutiert, unter welchen Bedingungen die Reifegradmodelle sich für die Qualitätsbewertung des Prozesses der SOA-Einführung in Unternehmen eignen und der erreichte Reifegrad sich für sinnvolle Prozessmetrik ansehen lässt. SOA-Reifegradmodelle werden also sowohl als Qualitätsbewertungsmittel als auch als Qualitätsbewertungsobjekte betrachtet. Anhand formulierter Kriterien wird eine Vergleichanalyse existierender Reifegradmodellen durchgeführt.

Schlüsselbegriffe: Serviceorientierte Architektur, Reifegradmodelle, Prozessqualität

Von gefährlich bis gesteuert: Reifestufen der Open-Source-Verwendung

Sven Euteneuer, Dr. Frank Simon

Software Quality Systems AG, Deutschland

sven.euteneuer@sqs.de, frank.simon@sqs.de

Zusammenfassung: Aufgrund der Anforderungen an die IT, in immer kürzerer Zeit immer komplexer werdende Systeme effizient und zuverlässig zu erstellen, gewinnt die Verwendung fertiger Open-Source-(Teil-)Systeme zunehmend an Bedeutung. Allerdings ist die ungesteuerte Wiederverwendung von Open-Source-Software (OSS) durchaus risikobehaftet. Es existiert eine Vielzahl vorhandener Lizenzmodelle mit jeweils unterschiedlichen, teils widersprüchlichen Anforderungen, die auf Kompatibilität miteinander und mit der gewünschten Verwendung untersucht werden müssen.

Auch wenn die Antwort auf die Frage, ob ein OSS juristisch korrekt innerhalb einer kommerziellen Anwendung wiederverwendet wurde, nur zwei Ausprägungen besitzt, können innerhalb der Unternehmen unterschiedliche Reifestufen bzgl. der systematischen OSS-Verwendung festgestellt werden. Im folgenden Papier werden diese Reifestufen auf konkrete 5 Reifegrade abgebildet: Diese ermöglichen den jeweiligen Unternehmen ein Rahmenwerk, mit dem eine genaue Positionsbestimmung bzgl. der OSS-Wiederverwendung möglich ist. Hierbei ist es ähnlich den Prozessmodellen (z.B. CMM(I) oder Spice) nicht in jedem Fall sinnvoll, die höchste Reifestufen erreichen zu wollen; es ist jedoch in jedem Fall notwendig, ein Bewusstsein und eine Einschätzung für das eigene Risiko zu entwickeln.

Management von SW-Zuverlässigkeit in der Produktentwicklung: ein Erfahrungsbericht

Axel Dold

DaimlerChrysler AG
Group Research & Advanced Engineering

axel.dold@daimlerchrysler.com

Zusammenfassung: Die Benutzerakzeptanz eines Systems hängt entscheidend von seiner Zuverlässigkeit und Verfügbarkeit ab. Angesichts des Vormarschs der eingebetteten Systeme und da gleichzeitig immer mehr Funktionalität im Fahrzeug in Software realisiert wird, gilt es neben der Hardware-Zuverlässigkeit auch Aspekte der Software-Zuverlässigkeit zu berücksichtigen. Im Rahmen einer internen Studie wurden die derzeitigen Ansätze aus der Literatur zum SW Reliability Engineering im Hinblick auf deren Relevanz und Anwendbarkeit für die Entwicklungsdomäne Automotive bewertet, Nutzungsszenarien hergeleitet und erste Evaluierungen anhand dieser Nutzungsszenarien durchgeführt. Dieser Artikel beschreibt die Erfahrungen mit Zuverlässigkeitswachstumsmodellen zur Prognose des Ausfallverhaltens während der Elektrik/Elektronik-(E/E-)Systemintegration, innerhalb derer die Komponenten der Zulieferer zusammengeführt und in ihrem Zusammenspiel abgesichert werden. Auf Basis der bisher durchgeführten Analysen ergab sich insgesamt ein zufrieden stellendes Prognoseverhalten. Entwickler können somit bereits während der ersten Hardware-in-the-Loop (HiL)-Tests Prognosen über das weitere Ausfallverhalten der Komponente durchführen und bei Abweichungen vom erwarteten Ausfallverhalten rechtzeitig reagieren und gegensteuern.

Schlüsselbegriffe: SW Metriken, Software Reliability Engineering, Automotive Embedded Systems

Von Risiken zu Nutzen: Vorhersage von Qualitätsrisiken

Andrea Herrmann

Universität Heidelberg

herrmann@informatik.uni-heidelberg.de

Zusammenfassung: Qualität kann man definieren als das Nicht-Vorhandensein oder die Verringerung von Risiken. In diesem Beitrag wird gezeigt, dass diese Konzentration auf Risiken es aber auch erleichtert, Abhängigkeiten zwischen Qualitätsattributen zu modellieren und Ähnlichkeiten zwischen ihnen zu finden.

Es wird zunächst definiert, was Risiko in Zusammenhang mit verschiedenen Qualitätsattributen – wie Sicherheit, Zuverlässigkeit – bedeutet, d.h. auf welchen unerwünschten Zustand jedes Qualitätsattribut sich jeweils bezieht und wie Qualitätsmetriken Risiken messen.

Die Vorhersage von Qualitätsrisiken, d.h. solchen, die mit der Systemqualität verbunden sind, unterstützt die Quantifizierung des Nutzens von Qualitätsanforderungen/-maßnahmen, die dieses Risiko verringern oder verhindern. Diese Abschätzungen will man oft bereits vor der Implementierung durchführen. Viele der gängigen Vorhersagemodelle sind allerdings nicht zu einer derart frühzeitigen und quantitativen Vorhersage geeignet. Weitere Arbeit hierzu wird nötig sein, wobei die in dieser Arbeit herausgearbeiteten Ähnlichkeiten und Abhängigkeiten zwischen den Qualitätsattributen vermutlich hilfreich sein werden.

Schlüsselbegriffe: Anforderungen, Qualität, Risiko, Vorhersagemodelle

BugzillaMetrics - Design of an adaptable tool for evaluating user-defined metric specifications on change requests

Lars Grammel, Holger Schackmann, Horst Lichter

RWTH Aachen University – Research Group Software Construction

lars.grammel@googlemail.com,
{schackmann, lichter}@cs.rwth-aachen.de

Abstract. The evaluation of metrics on the data available in change request management (CRM) systems can give valuable information for the management of software development. It can for example be helpful in assessing the current workload, product quality or development process weaknesses.

Metrics and charts on change requests are already available in current CRM systems. They provide information about common metrics, but their adaptability is limited with respect to the specification of metrics customized to organizationspecific needs.

This paper describes a more flexible approach for the evaluation of metrics on change requests. The core part of the presented tool is an event driven evaluation algorithm for the calculation of time series data. It is parametrized with user defined metric specifications. This enables a separation between metric specification and information retrieval. Further design decisions enable a transparent execution optimization and an abstraction from the data sources of the underlying CRM database.

Keywords: Process metrics, change request management, metrics specification

Fallen und Verräter –

der Einsatz von Metriken in kommerziellen Softwareprojekten

Dipl.-Math. Dirk Malzahn

OrgaTech GmbH, Zum Pier 73-75, 44536 Lünen

dm@orgatech.org

Zusammenfassung: Der Artikel widmet sich im Kern den Erfahrungen, die innerhalb der letzten drei Jahre bei der qualitativen Begleitung eines Softwareprojekts gemacht wurden. Zu Beginn wurden Metriken lediglich als „Managementbefriedigung“ betrachtet. Der Einsatz der CMMI brachte den weiteren Zwang, sich auch auf Projektebene mit dem Thema Metriken auseinanderzusetzen. Um den für Metriken getriebenen Aufwand zu rechtfertigen, wurde seitens der Projektleitung die Forderung aufgestellt, auf Basis der Metriken auch Potenziale im eigenen Projekt zu ermitteln. Hier wurde schnell festgestellt, dass die bereits vorhandenen Management-Metriken nur einen geringen Nutzwert für die Optimierung des Projekts brachten. Um ein Metriksystem für das Projekt aufzubauen, wurde ein CMMI konformer Prozess aufgesetzt, der insbesondere weiche Faktoren wie Verantwortung, Identifikation mit dem Projekt, Schulungsstand, etc. umfasst. Durch die Beschäftigung mit diesem Thema schaffte es das Projektteam, sich von einem Team unter vielen zum weltweit besten Team innerhalb des international operierenden Konzerns zu entwickeln.

Im Rahmen dieses Artikels wird die Entwicklung des Metriksystems dargestellt. Hier liegt der Schwerpunkt auf der Darstellung, welche Fallen es bei der Definition eines Metriksystems gibt und wie sichergestellt wird, dass Metriken ihr Projekt nicht dadurch verraten, dass diese inkonsistent ermittelt und vom Team nicht akzeptiert sind.

Schlüsselbegriffe: Weiche Metriken, kommerzielle Softwareentwicklung, CMMI, SPICE

Erfassung und Auswertung von Metriken über Internetpräsenzen

*Cornelius Wille, Reiner Dumke**

Fachhochschule Bingen
Berlinstraße 109, 55411 Bingen, Germany,
Tel.: +49-6721-409-257, Fax: +49-6721-409-158,

wille|@fh-bingen.de

*Otto-von-Guericke-Universität Magdeburg. Institut für Verteilte Systeme
Universitätsplatz 2, 39106 Magdeburg, Germany,
Tel.: +49-391-67-18664, Fax: +49-391-67-12810,

dumke@ivs.cs.uni-magdeburg.de

Zusammenfassung: Kaum ein großes Unternehmen nutzt sein Internetportal nur zur räsentation nach Außen. Neben Geschäftsmodellen werden im zunehmenden Maße auch die Support- und Servicedienstleistungen über das Internet angeboten. Der folgende Beitrag präsentiert einen Ansatz wie der gezielte Einsatz von Metriken in diesem Anwendungsgebiet zu einer Erhöhung der Kundenzufriedenheit beitragen kann. Ausgehend von dem Ziel der Kunden, die gesuchte Information, Anleitung oder Treiber möglichst schnell zu finden, wird dabei die Länge des Klickpfades im Navigationsbaum als Indikator der zu erwartenden Kundenzufriedenheit betrachtet. Im Beitrag wird ein Bewertungsmodell vorgestellt, um die in den Logfiles der Webserver enthaltenden Informationen in einen Kundenzufriedenheitsindex

umzusetzen. Die Ergebnisse lassen sich zu einer zielgerichteten Verbesserung des Aufbaus und der Struktur der Internetportale nutzen.

Schlüsselbegriffe: Webmetrik, Internetportal, Kundenzufriedenheit, Messung

GQM+Strategies®: A Comprehensive Methodology for Aligning Business Strategies with Software Measurement

Victor R. Basili [A, D], Jens Heidrich [B, C], Mikael Lindvall [A], Jürgen Münch [B], Myrna Regardie [A], Dieter Rombach [B, C], Carolyn Seaman [A, D], and Adam Trendowicz [B]

[A] Fraunhofer CESE, [B] Fraunhofer IESE, [C] University of Kaiserslautern, and [D] University of Maryland

basili@fc-md.umd.edu, jens.heidrich@iese.fraunhofer.de, mikli@fc-md.umd.edu, juergen.muench@iese.fraunhofer.de, mregardie@fc-md.umd.edu, dieter.rombach@iese.fraunhofer.de, cseaman@fc-md.umd.edu, adam.trendowicz@iese.fraunhofer.de

Abstract. In software-intensive organizations, an organizational management system will not guarantee organizational success unless the business strategy can be translated into a set of operational software goals. The Goal Question Metric (GQM) approach has proven itself useful in a variety of industrial settings to support quantitative software project management. However, it does not address linking software measurement goals to higher-level goals of the organization in which the software is being developed. This linkage is important, as it helps to justify software measurement efforts and allows measurement data to contribute to higher-level decisions. In this paper, we propose a GQM^{+}Strategies® measurement approach that builds on the GQM approach to plan and implement software measurement. GQM+Strategies® provides mechanisms for explicitly linking software measurement goals to higher-level goals for the software organization, and further to goals and strategies at the level of the entire business. The proposed method is illustrated in the context of an example application of the method.

Keywords: Software measurement, goal question metric approach, project management, strategic management, business strategy, software strategy, IT strategy

Einsatz von Metriken zur Vorbereitung einer Sprachkonvertierung

Friedrich Volmering[1], Georgios Lajios[2], Doris Schmedding[1]

[1] Universität Dortmund, Informatik (Software-Technologie)

doris.schmedding@udo.edu, friedrich.volmering@gmx.de

[2] Die Continentale, Dortmund

georgios.lajios@continentale.de

Zusammenfassung: Wir setzen eine Werkzeugkette aus Syntaxanalyse und Metriken zur automatisierten Generierung von Berichten ein, die bei der Durchführung einer Sprachkonvertierung von C++ nach Java mögliche Problemstellen und Konvertierungshindernisse aufzudecken helfen. So wird die gezielte Vor- und Nachbereitung des Quellcodes und die Aufwandsschätzung für die Konvertierung unterstützt.

Schlüsselbegriffe: Metriken und Tools, Aufwandschätzung für Sprachkonvertierung, Kombination von Syntaxanalyse mit Metriken, Zusammenarbeit von Universität und Industrie

Fehlerverursachende Strukturen in objektorientierter Software

Roland Neumann, Gerrit Hanselmann

Technische Universität Kaiserslautern, Fraunhofer IESE Kaiserslautern

roland.neumann | gerrit.hanselmann @informatik.uni-kl.de

Zusammenfassung: Softwarefehler sind ein großes Qualitätsrisiko und verursachen Ausfälle. Die statische Analyse erlaubt mithilfe von Softwaremaßen Strukturen zu identifizieren, deren Zusammenhang mit aufgetretenen Fehlern mithilfe eines Prognosemodells erfasst werden kann. Fehlermodelle von genügend vielen Studien erlauben eine Generalisierung nach möglichen Fehlerursachen und eine Prognose wahrscheinlicher Fehler. Dieser Überblick soll in dieser Untersuchung dargestellt werden.

Schlüsselbegriffe: Praktische Messverfahren, Empirische Fallstudien, Hauptkomponenten-analyse, Statische Fehlerprognose

Granularitätsmetriken für serviceorientierte Architekturen

Dmytro Rud[1], Steffen Mencke[1], Andreas Schmietendorf[1,2], Reiner R. Dumke[1]

[1] Arbeitsgruppe Softwaretechnik, Institut für Verteilte Systeme, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg

{rud, mencke, schmiete, dumke}@ivs.cs.uni-magdeburg.de

[2] Fachhochshule für Wirtschaft Berlin, Fachbereich II – Wirtschaftsinformatik
Neue Bahnhofstr. 11-17, 10245 Berlin

schmiete@fhw-berlin.de

Zusammenfassung: In diesem Beitrag werden verschiedene Arten der Granularität von Services in serviceorientierten Architekturen (SOA) diskutiert. Zur Bewertung dieser Eigenschaften werden entsprechende Metriken vorgeschlagen.

Schlüsselbegriffe: Serviceorientierte Architektur, SOA, Service, Schnittstelle, Granularität, Qualität

Möglichkeiten der Aufwandsschätzung im Zeitalter agiler Methoden zur Softwareentwicklung

Andreas Schmietendorf [1, 2], Reiner Dumke [2]

FHW Berlin (Berlin School of Economics), Fachbereich II
Neue Bahnhofstrasse 11-17, D-10245 Berlin

schmiete@fhw-berlin.de

Otto-von-Guericke-Universität Magdeburg, FIN/IVS (Softwaretechnik)
Universitätsplatz 2, D-39106 Magdeburg

schmiete|dumke@ivs.cs.uni-magdeburg.de

Zusammenfassung: Die kommerzielle Entwicklung von Software bewegt sich grundsätzlich im Spannungsfeld zwischen Kosten, Qualität, Funktionalität und Zeit. Dementsprechend gilt es den funktionalen Umfang einer zu entwickelnden Lösung bereits zu Beginn der Entwicklung einschätzen zu können, um auf dieser Grundlage auf korrespondierende Aufwände zu schließen. Methoden, wie z.B. das Function-Point-Verfahren (durch A. J. Albrecht 1979 bei IBM entwickelt), unterstützen diese Aufgabenstellung. Wenngleich das Function-Point-Verfahren bzw. die in Anlehnung an diese Idee entwickelten Derivate eine methodische Grundlage bieten, kann deren Verwendung im industriellen Kontext nicht befriedigen. Zumeist stellt sich mit jedem neuen technologischen Hype auch die Frage, wie dieser im Rahmen einer Aufwandsschätzung berücksichtigt werden kann bzw. muss. Im Rahmen dieses Beitrags soll der Aspekt der Aufwandsschätzung im Rahmen agil durchgeführter Entwicklungsprojekte betrachtet werden.

SOA & Metriken – Die Aufwandsschätzung – Einsatz der Function-Point-Methode in einem durchgängigen SOA-Entwicklungsprozess

Dierk Pingel

Management-, Architektur-, Prozess- und Methodenberater

Pingel@t-online.de oder **Pingel@SE-Vision.de**

Zusammenfassung: Aufwandsschätzungen sind immer noch ungenau, obwohl es bereits verschiedene Methoden gibt, den Projektumfang und die Systemkomplexität zu bestimmen. Der folgende Beitrag wird zeigen, wie mit Hilfe eines wohl definierten Entwicklungsprozesses Aufwandsschätzungen verbessert werden können. Da SOA (Service-Orientierte Architekturen) in aller Munde ist, soll als Entwicklungsprozess beispielhaft der in [2] beschriebene SOA-Entwicklungsprozess verwendet werden. Als Methode zur Bestimmung des Projektumfangs soll die seit Jahren etablierte Function-Point-Methode eingesetzt werden. Die Methode werden wir leicht modifizieren, um eine noch höhere Praxisrelevanz zu erhalten.

Schlüsselbegriffe: SOA, Entwicklungsprozess, Function-Point-Methode, Aufwandsschätzung, Prozessmodellierung

TestASS: Eine Methode zur Testaufwandsschätzung

Ulrike Dowie, Georg Herzwurm

Universität Stuttgart, Lehrstuhl für allg. BWL und Wirtschaftsinformatik II

ulrike.dowie@gmx.de, herzwurm@wi.uni-stuttgart.de

Zusammenfassung: Im Rahmen einer Forschungsarbeit zur Ermittlung der Einflussfaktoren des Testaufwands wurde eine Methode entwickelt und erprobt, mit Hilfe derer aus der Menge potenzieller Einflussfaktoren die in einer bestimmten Organisation relevanten Einflussfaktoren identifiziert werden können. Zusammen mit der Auswahl und Integration geeigneter Schätzmethoden kann so ein organisationsspezifisches Vorgehen zur Schätzung des Testaufwands hergeleitet werden. Dieser Beitrag stellt die Methode TestASS (Test-Aufwands-Schätzung für Standardsoftware) vor.

Schlüsselbegriffe: Testaufwand, Aufwandsschätzung, organisationsspezifisches Modell

BSOA07 – Workshopbericht

Andreas Schmietendorf

Fachhochschule für Wirtschaft Berlin
(Ansprechpartner der BSOA-Initiative)

schmiete@fhw-berlin.de

1. Hintergrund zur BSOA-Initiative

Serviceorientierte Architekturen (kurz SOA) beeinflussen die Vorgehensweise zur Implementierung neuer Softwareanwendungen erheblich. IT-Analysten sehen mit SOA die Möglichkeit, monolithische Softwarearchitekturen aufzubrechen. Darüber hinaus verspricht dieser Trend eine stärkere Ausrichtung der Informationstechnologie an den Geschäftsprozessen. Neue Anwendungen sollen primär auf der Basis bereits existierender fachlicher Serviceangebote erstellt werden können. Um solche Kompositionen von Serviceangeboten schaffen zu können, redundante Systementwicklungen zu vermeiden und entsprechend Kosten einzusparen, werden neue Bewertungsmodelle benötigt, die prozess-, produkt- und ressourcenbezogene Aspekte im Kontext von SOA berücksichtigen. Die BSOA-Initiative greift diese Herausforderung auf und widmet sich unter anderem den folgenden Themenstellungen:

- Bewertung der Nutzungspotenziale einer SOA,
- Erarbeitung von Richtlinien zur Serviceentwicklung für eine SOA,
- Qualitätsbewertung angebotener Services und Kompositionen,
- Mess- und Bewertungsansätze zum Reifegrad einer SOA,
- Services Level Agreements (SLAs) und Verhandlungsaspekte.

Am 13. November 2007 fand dazu der 2. Workshop am Forschungszentrum Informatik in Karlsruhe (FZI) statt. Wie bereits auf dem 1. Workshop im Jahr 2006 an der Fachhochschule für Wirtschaft Berlin konnte auch dieses Mal ein ausgewogenes Teilnehmerfeld aus dem industriellen und akademischen Umfeld beobachtet werden.

Der diesjährige Workshop wurde in Kooperation zwischen der FHW Berlin, dem FZI Karlsruhe und der Otto-von-Guericke-Universität Magdeburg (Softwaremesslabor) unter der Schirmherrschaft der CECMG (Central Europe Computer Measurement Group) veranstaltet. Darüber hinaus erfährt die BSOA-Initiative Unterstützung durch die GI (Gesellschaft für Informatik) und die DASMA (Deutschsprachige Interessensgruppe für Softwaremetrik und Aufwandsschätzung).

In seiner Grußnote unterstrich der Direktor des Forschungsbereichs Programmstrukturen am FZI Karlsruhe und Vizepräsident der Gesellschaft für Informatik Herr Prof. Dr. Andreas Oberweis den aktuellen Bedarf an nachhaltigen

und wissenschaftlich fundierten Bewertungsansätzen im Kontext der vielfältigen SOA-Ansätze.

2. Inhalte des Workshops

Aus den eingereichten Beiträgen wurden im Rahmen eines bundesweit zusammengesetzten Programmkomitees 6 Beiträge für einen Vortrag bzw. 3 Beiträge für eine Posterpräsentation ausgewählt. Darüber hinaus konnten mit Herrn Harry Sneed und Herrn Rüdiger Molle zwei industriell erfahrene Gastredner gewonnen werden.

Im Folgenden findet sich eine zusammenfassende Darstellung der auf dem Workshop gehaltenen Vorträge:

Harry Sneed (Anecon GmbH / Österreich): Ein Qualitätsnachweis für Web Services (eingeladener Beitrag)

In diesem Beitrag geht der Autor auf die Frage der Zuverlässigkeit von Web Services ein. Dafür werden der mögliche Aufbau eines Qualitätsnachweises, der Prozess des Testens von Services und benötigte Werkzeuge für die Qualitätssicherung erörtert. Sehr treffend formuliert der Autor im Rahmen seiner Zusammenfassung: „Erst wenn nachgewiesen ist, dass Web Services wirklich im Umfeld des Anwenders korrekt und performant funktionieren wird der Weg zur SOA frei sein“.

Rüdiger Molle (ITAB Hamburg): Bewertung von Service-Kompositionen mittels Simulation (eingeladener Beitrag)

Der ehemalige Director Gartner Consulting (IT-Architekturen) geht in seinem Beitrag auf die Wechselwirkungen zwischen Geschäftsprozessen und serviceorientierten Architekturen ein. Dabei beschäftigt er sich insbesondere mit der simulativen Analyse von mit Hilfe von BPMN modellierten Geschäftsprozessen.

Christoph Rathfelder, Henning Groenda (FZI Karlsruhe): Geschäftsprozessorientierte Kategorisierung von SOA

Im Mittelpunkt dieses Beitrags steht die geschäftsprozessorientierte Kategorisierung verschiedener SOA-Typen. Die Autoren identifizieren und beschreiben dafür die folgenden SOA-Typen: Integration SOA, Business Function SOA, Business Process SOA und On Demand SOA.

Niko Zenker, Martin Kunz, Claus Rautenstrauch (Otto-von-Guericke-Universität Magdeburg): Service Oriented Architecture: Resource Based Evaluation of a SOA

Die durch einen Service belegten Ressourcen werden im Rahmen dieses Beitrags untersucht. Mit dem rModel wird eine Erweiterung des UDDI-inhärenten tModels

vorgeschlagen, so dass auch korrespondierende Ressourcenverbräuche bei der Serviceauswahl berücksichtigt werden können.

Matthias Stutz, Stephan Aier (Universität St. Gallen /Schweiz): Szenariobasierte Architekturbewertung für serviceorientierte Architekturen

Die Autoren schlagen eine fachliche und architekturspezifische Bewertung für serviceorientierte Architekturen vor, um so die Wirkung auf die Flexibilisierung von Geschäftsprozessen nachweisen zu können. Dabei gehen sie von einem szenariobasierten Bewertungsansatz aus, um auf diese Weise Architekturalternativen vor der Implementierung bewerten zu können.

Evgeni Dimitrov, Marco Kuhnert, Andreas Schmietendorf (T-Systems, FHW Berlin): Bewertungsansätze im Kontext der SOA-Governance

Mit diesem Impulsbeitrag wird ein generischer Ansatz für die Etablierung einer SOA-Governance im industriellen Umfeld vorgestellt und hinsichtlich möglicher Bewertungsansätze untersucht. Grundlage dafür bildet das SOA-Governance-Model, welches die zu berücksichtigenden Aspekte (Strategie, Management, Organisation, Architektur und Operationalisierung) als Ordnungsrahmen aufzeigt.

Jan vom Brocke, Christian Sonnenberg (Hochschule Liechtenstein /FL Liechtenstein): Serviceorientiertes Prozesscontrolling

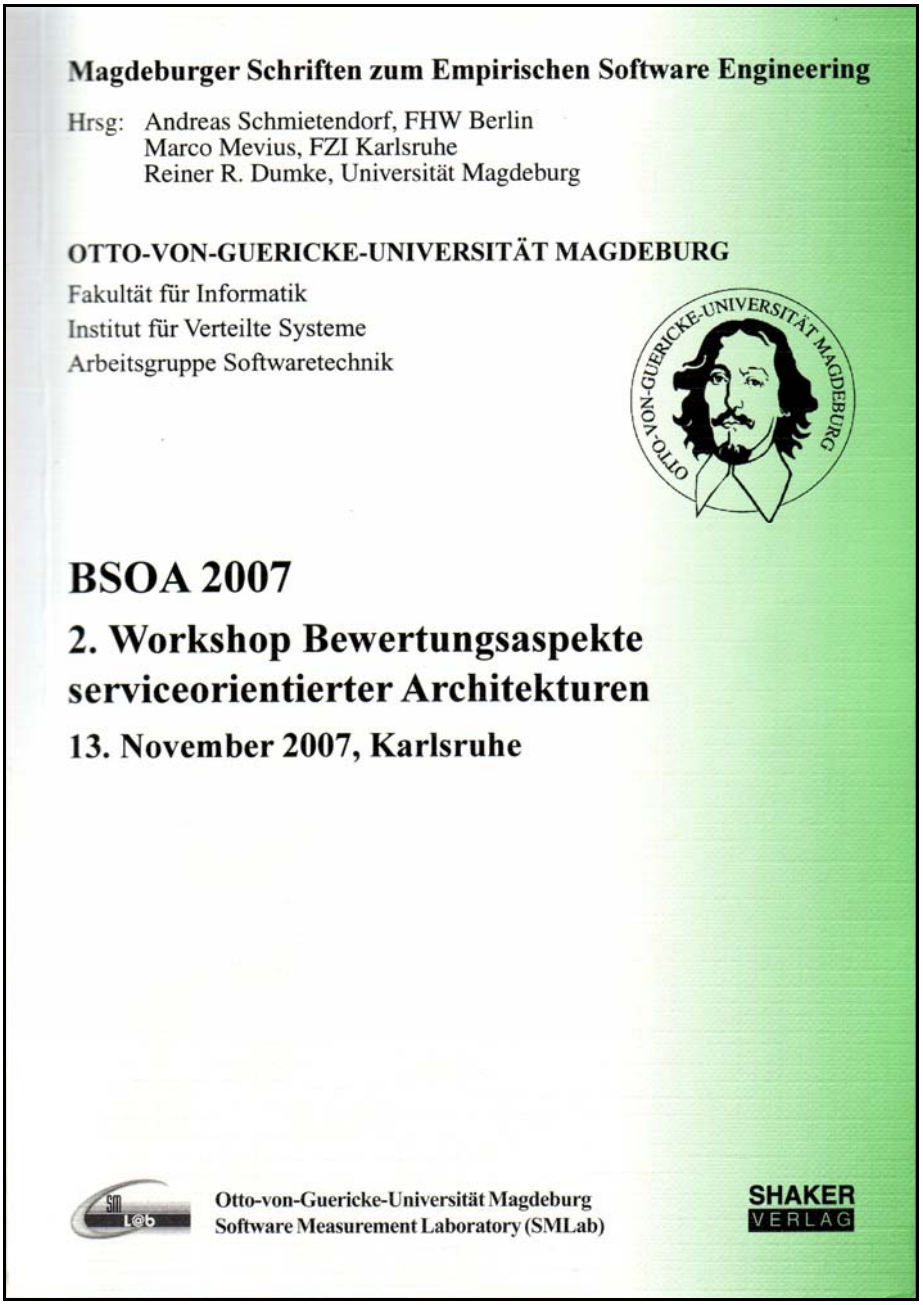
Dieser Beitrag greift die monetäre Bewertung der Nutzungspotenziale einer Serviceorientierten Architektur aus unternehmensindividueller Sicht auf. Mit Hilfe der Methoden des serviceorientierten Prozesscontrollings erfolgt die beispielhafte Analyse zur Wirtschaftlichkeit eines SOA-Einsatzes.

Florian Dreifus, Peter Loos (Universität Saarbrücken): SOA-Reifegrad – Eine konzeptionelle Darlegung relevanter Erhebungsaspekte

Der Beitrag geht neben der konzeptionellen Darlegung einer geeigneten Vorgehensweise zur Erhebung des SOA-Reifegrades auf die Herleitung eines entsprechenden SOA-Reife-gradmodells ein. Das vorgeschlagene Modell berücksichtigt vier Ebenen – Add-Hoc Support, Integration Platform, Collaboration Platform und Business Process Platform.

3. Weitere Informationen

Sämtliche Beiträge zum Workshop (inklusive die der Gastredner) wurden in einem Tagungsband beim Shaker-Verlag, in der Reihe „Magdeburger Schriften zum Empirischen Software Engineering“, publiziert. (ISBN 978-3-8322-6716-2)



Weiterführende Informationen zur BSOA-Initiative, wie z.B. der *Call for Paper* für den kommenden BSOA-Workshop im Jahr 2008, finden sich unter folgender URL:

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

4. Dank

Ohne vielfältige Unterstützung ist die Durchführung eines solchen Workshops nicht denkbar. Ein herzlicher Dank geht dem- entsprechend an die Gastgeber des Workshops Herrn Prof. Dr. Andreas Oberweis und Herrn Dr. Marco Mevius vom FZI Karlsruhe für die Bereitstellung von Räumen und die Infrastruktur. Besonderer Dank gilt der Fa. Trilog AG für die Übernahme der Catering-Kosten.

DSML Success Factors and Their Assessment Criteria

Abdelilah Kahlaoui, Alain Abran, Éric Lefebvre

Abstract. *Over the past few years, a number of Domain Specific Modeling Languages (DSMLs) have been developed, and their use has increased in approaches such as Model Driven Engineering (MDE), software factories and even MDA (Model Driven Architecture). However, developing a DSML is still a challenging and time-consuming task. Issues to tackle include the DSML development process, DSML quality and DSML model verification and validation (V&V). Therefore, techniques and solutions are needed to make DSML development easier and more accessible to software developers and domain experts. This paper recommends a list of success factors to consider when developing or choosing a DSML for those developing it, and for software developers and domain experts interested in using it. The paper then maps these success factors to a set of assessment criteria that can be used to assess DSML quality.*

1 Introduction

Models play a central role in the Model Driven Engineering (MDE) approach, and they constitute the main artifacts to develop in the software development life cycle. While they have traditionally been used mainly for documentation purposes, models are considered in MDE as first-class entities that can (and should) be used for code generation.

This use of models as inputs to code generation increasingly demands high-quality domain-specific modeling languages capable of producing formal models that can be processed by tools (i.e. generators, interpreters, compilers, etc.) [1]. Examples of the quality characteristics required include formality, domain specificity and expressiveness, among others.

Most of the existing modeling languages lack such characteristics. To help developers have a clear idea of what makes a good DSML, and to help deciders choose the right DSML to meet their needs, a set of assessment criteria for both functional and quality attributes, as well as their related measures, should be set up.

In this paper, we provide a list of the success factors we consider important for domain-specific modeling languages and propose a technique for converting them into assessment criteria. The technique was designed to be generic, so that it can be used for domains other than DSMLs.

This paper is organized as follows. Section 2 summarizes related work on the quality of models and modeling languages. Section 3 identifies a set of success factors that should be considered when building a DSML. Section 4 describes a technique for converting success factors into assessment criteria. Finally, section 5 concludes the paper with a discussion.

2 Related work

The subject of domain-specific modeling languages has been studied from a variety of perspectives, among them DSML design, the DSML definition process, DSML building tools and DSML quality. Three of these topics constitute a good starting point for the study of DSML success factors. They are:

a. **Quality of models:** The effort in this domain has been focused on finding solutions to improve the quality of models by proposing methods and techniques which help build better-quality models. Here, a distinction is made between studies which have focused on models built using the Unified Modeling Language (UML) [2;3] and those which have extended their scope to cover conceptual models in general, regardless of the modeling language used to build them.

   Quality characteristics that have been found to be essential in the case of UML can be directly applied to DSML. However, it is to be noted that these are not enough, and do not take into account some of the specific aspects of DSMLs, namely those characteristics related to domain specificity, models transformation and code generation.

   Similarly, it has been noted that the studies that have examined conceptual models in general usually focus on specific categories of models, such as process models [4;5], requirements models, data models [6;7], etc., and also that there is a need for research to investigate the quality of models from a domain-specific perspective.

b. **Quality of modeling languages:** Authors in this field have looked at the issue of modeling language quality and assessment from a variety of perspectives. These studies cover modeling language evaluation [8-10], the development of evaluation methodologies [11] [12;13] and design principles for modeling languages [14].

   Similarly, it has been noted that aspects related to the nature of domain-specific modeling languages are missing.

c. **DSML design experiences:** In the last fifty years or so, hundreds of DMSLs have been built. The experience accumulated in developing these languages can serve as a good resource for identifying success factors. For example, lessons reported by Wile [15] can be very easily transformed into success factors.

3 DSML Success Factors

3.1 Identification of success factors

The following success factors have been identified by combining the results of work carried out in the three dimensions described in the previous section:

- ***Domain expertise***: DSML development requires an in-depth knowledge of the domain of interest, and domain knowledge facilitates the identification of domain concepts, terminology, rules and constraints. This can be achieved using domain analysis methods.
- ***Domain scoping***: Defining the appropriate scope for the domain is a critical task, as it determines the utility and usefulness of the DSML. If the scope is too broad, DSMLs will be less specific and less expressive; if it is too narrow, the return on investment might be low.
- ***Effective support tools***: DSML development is difficult. It needs to be supported by a set of tools that can automate some of the more tedious tasks in the DSML development process (i.e. analysis, verification, validation, code generation, etc.).
- ***Effective meta-model***: Developers should choose the meta-models used to define their DSML carefully. An effective meta-model will make it easier to define formal, unambiguous and expressive DSMLs. By contrast, an inappropriate meta-model may have a negative impact on a DMSL's quality.
- ***Effective underlying generator***: Since the aim of domain-specific modeling is to increase productivity by eliminating, or at least reducing, manual coding, generators capable of transforming DSML models into code are required. Without these transformation tools, DSML models will only be used for documentation purposes.
- ***High level of abstraction***: For a DSML to be effective and useful, it should define abstractions that use domain experts' vocabulary; in other words, it should raise the level of abstraction to bring the implementation world closer to the specification world. This can be done by defining languages based on domain concepts rather than on code concepts.
- ***Domain engineering environment (DEE)***: Ideally, DSML development should occur within a DEE. This is where all the core assets (i.e. reusable components, architectures, patterns, design, etc.) should be developed. DEE include, among others, domain engineers, domain experts, domain developers and DSML designers. Their primary goal is to collect, organize and model domain knowledge. The availability of rich domain knowledge is critical in identifying concepts and defining DSML elements.
- ***Language development expertise*:** Defining a domain-specific language is not an easy task. Skilled specialists in language development are required to define convenient DSMLs, and a lack of expertise may lead to some awkward and unfitted DSMLs. Any organization that decides in favor of in-house DSML development should consider assigning (or possibly hiring) the appropriate staff to accomplish the job;
- ***Viewpoint orientation*:** Viewpoints are a great way to separate and organize stakeholder concerns, and a viewpoint-oriented DSML is most likely to fit the needs of its users. Focusing on one perspective of the system at a time makes DSML models more specialized and useful;

- ***Purpose-orientation*:** A DSML is a specialized language designed to deal with a particular problem within a single domain;
- ***Domain expert support***: Domain experts are the primary users of DSMLs. Their praise for the DSML and their approval of it are critical to its adoption. Developers of these languages should make sure that they provide a DSML that fills domain experts' needs.
- ***Effective DSML definition process*:** As with any engineering activity, DSML development should be based on a set of well-defined processes, practices and tools. The process describes the activities to perform and the artifact to deliver when developing a DSML.

3.2 Categorization of success factors

In this section, a categorization scheme similar to that proposed by Wile [15] is given to help differentiate among the success factors listed above (see Table 1).

- **Organizational**: related to the organizational culture (i.e. the organization's mission, values, beliefs, norms, etc.);
- **Personal**: related to human resources' capabilities (i.e. competencies, experiences, expertise, etc.);
- **Social**: related to social behaviors and relationships;
- **Technical**: related to technical issues, such as tools and technologies.

Table 1. Success Factor Classification

Success Factor	Technical	Organizational	Social	Personal
Domain expertise				**X**
Domain scoping		**X**		
Effective supporting tools	**X**			
Effective meta-model	**X**			
Effective underlying generator	**X**			
Domain engineering environment		**X**		
High level of abstraction	**X**			
Language expertise				**X**
Viewpoint orientation	**X**			
Purpose-orientation		**X**		
Domain expert support			**X**	
Effective DSML definition process		**X**		

4 Transformation Technique

To transform DSML success factors into criteria for DSML assessment, a four-step technique has been designed (see Figure 1). The technique is presented below, and the way in which it can be used to extract DSML assessment criteria from the above DSML success factors is explained:

1. **Identifying success factor impact**: the existence (or non-existence) of a success factor has a direct (or indirect) impact on the quality of the DSML. The existence/absence of a success factor has a positive/negative impact and, usually, affects one or more elements of the subject (here, the subject is a DMSL).
2. **Identification of the EAI (*Elements Affected by the Impact*):** identification of the elements (i.e. some aspects related to the subject) affected by the success factor impacts.
3. **EAI attribute identification**: For each EAI, we determine the features and properties that characterize it and assist in its assessment. These features and properties are then organized into categories to facilitate the selection of those that affect the quality of the subject;
4. **Selecting attributes that have a direct effect on quality**: selection of the attributes that affect the quality of the subject (i.e. internal quality, external quality and quality in use as defined in the ISO/IEC 9126 models of the quality of software products [16]).

Table 2 illustrates an application of this technique to derive assessment criteria for DMSLs:

- From the list of DSML success factors identified in section 2, we identified a set of positive and negative impacts related to the existence or the absence of these factors;
- Then, we identified the elements affected by this impact for each success factor. For a DMSL, we have identified four elements:
    a. ***Abstract Syntax:*** defines the essential concepts and structures to be modeled in a DSML;
    b. ***Concrete Syntax:*** defines a notation (i.e. visual appearance) for the concepts defined by the abstract syntax and how these abstract concepts are realized in a concrete notation such as text or graphics;
    c. ***Semantics:*** gives a meaning to the abstract concepts and their relationships;
    d. ***Views:*** defines a perspective from which a given aspect of a software product can be described [1].

  Finally, we extracted a list of assessment criteria for each element, each time taking into account the impact at hand

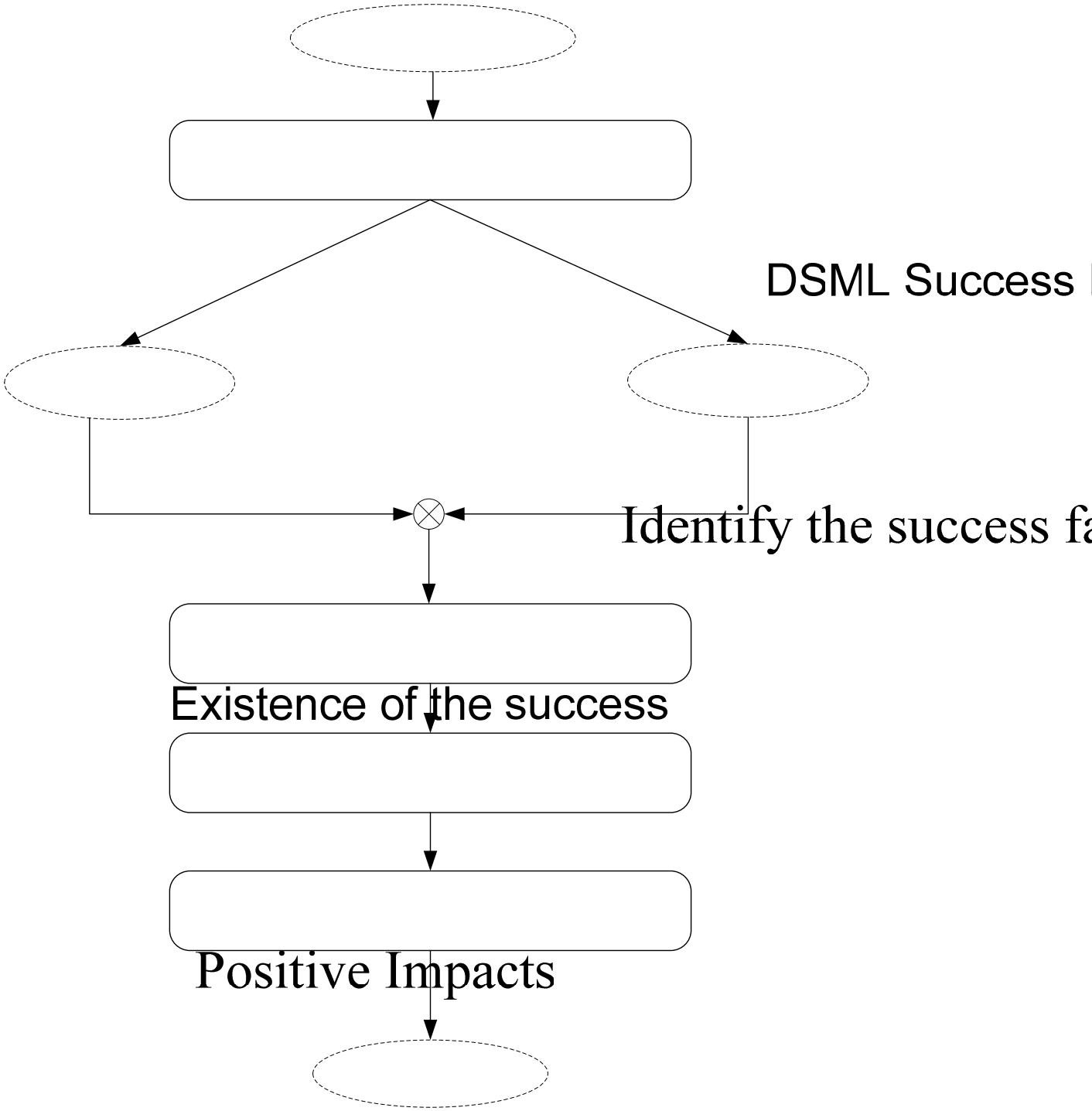


Figure 1. Process of transformation of success factors into assessment criteria

Identification of the elemen
impact (FA

5 Discussion

While in the past few years, a number of DSMLs have been developed and their use has increased in approaches, such as Model Driven Engineering (MDE) and MDA (Model Driven Architecture), there has been little work done on the quality of such languages. The motivation for the work reported here was to identify DSML assessment criteria that should be built in by those developing these languages, and that should be looked for by those software developers and domain experts interested in using them.

On the basis of success factors documented in the literature for DSMLs, we have proposed a technique to convert them into assessment criteria.

The list of DSML success factors and their corresponding assessment criteria is aimed at helping evaluators and decision makers assess these languages. Evaluators in other areas of knowledge may also use the success factor assessment criteria conversion technique to identify criteria that help them assess their products.

We do not claim that the list of success factors and assessment criteria presented in this paper is exhaustive. More effort is needed to investigate its completeness. Case studies are also required to verify and validate the relevance of these factors and assessment criteria in industrial contexts.

References

[1] Greenfield, J., & Short, K. (2004). *Software factories: assembling applications with patterns, models, frameworks, and tools*: Wiley Pub.

[2] Lange, C. F. J. (2006). Improving the quality of UML models in practice **http://doi.acm.org/10.1145/1134285.1134472** in *Proceeding of the 28th International Conference on Software Engineering* (pp. 993-996). Shanghai, China ACM Press.

[3] Unhelkar, B. (2005). *Verification and validation for quality of UML 2.0 models*. Hoboken, N.J.: John Wiley.

[4] List, B., & Korherr, B. (2006). An evaluation of conceptual business process modelling languages. **http://doi.acm.org/10.1145/1141277.1141633** in *Proceedings of the 2006 ACM Symposium on Applied Computing* (pp. 1532-1539). Dijon, France ACM Press.

[5] Krogstie, J., Sindre, G., & Jørgensen, H. (2006). Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems, 15*, 91-102.

[6] Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. **http://dx.doi.org/10.1016/j.datak.2004.12.005** *Data Knowl. Eng. , 55* (3), 243-276

[7] Moody, D. L., Sindre, G., Brasethvik, T., & S\&\#248;lvberg, A. (2003). Evaluating the quality of information models: empirical testing of a conceptual model quality framework, in *Proceedings of the 25th International Conference on Software Engineering* (pp. 295-305). Portland, Oregon IEEE Computer Society.

[8] Nysetvold, A., & Krogstie, J. Assessing Business Processing Modeling Languages Using a Generic Quality Framework. In *Proceedings of the CAiSE '05 Workshops, Vol. 1, Tenth International Workshop on Exploring Modeling Methods in Systems Analysis and Design, J. Castro and E. Teniente (eds.), FEUP Edições, Porto, 2005, pp. 545-556.*

[9] Krogstie, J. (2003). Evaluating UML using a generic quality framework In *UML and the unified process* (pp. 1-22): Idea Group Publishing.

[10] Krogstie, J., & Arnesen, S. Assessing Enterprise Modeling Languages using a Generic Quality Framework. *Information Modeling Methods and Methodologies*, 63-79.

[11] Morris, S., & Spanoudakis, G. (2001). *UML: an evaluation of the visual syntax of the language.* Paper presented at the System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34), ed. R.H. Sprague, Jr. (IEEE Computer Society, 2001)..

[12] Bobkowska, A. (2005). Modeling pragmatics for visual modeling language evaluation. **`http://doi.acm.org/10.1145/1122935.1122950`** in *Proceedings of the 4th International Workshop on Task Models and Diagrams* (pp. 75-78). Gdansk, Poland, ACM Press.

[13] Bobkowska, A. (2005). A Methodology of Visual Modeling Language Evaluation. In *SOFSEM 2005: Theory and Practice of Computer Systems, LNCS 3381*: Springer.

[14] Paige, R. F., Ostroff, J. S., & Brooke, P. J. (2000). Principles for modeling language design. *Information and Software Technology, 42*(10), 665-675.

[15] Wile, D. (2004). Lessons learned from real DSL experiments. *Science of Computer Programming, 51*(3), 265-290.

[16] ISO/IEC 9126:1999. Software Engineering – Product quality. Int. Org. for Standardization, ISO 9126, 1999.

Reiner R. Dumke

1 Empirical Software Engineering

We already mentioned that *"software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software"* (IEEE definition, in [1]).

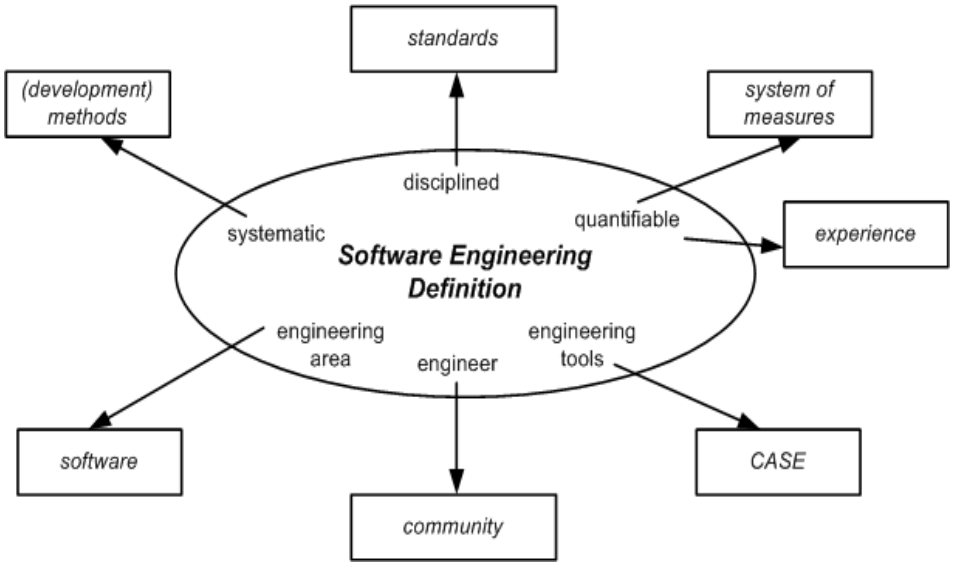


Figure 2. Software Engineering Aspects

Empirical software engineering motivates to consider all relevant aspects of software engineering in order to understand, explain, improve and control software systems, processes and resources. Essential approaches are emprical validation (based on controlled experiments, case studies and assessments) and exploration leading to a confirmation (rule of thumb, analogies, laws, priniples, lemmas, formulas and calculus) in the manner of scientific evidence [1]. "The goal of empirical software engineering is to influence the practice of software engineering" (B. Kitchenham, in [2]).

2 The Dagstuhl Workshop Initiative

In the Dagstuhl castle in Germany a group of software engineering researchers discussed in order to provide many insightful perspectives and suggestions for defining a roadmap for Empirical Software Engineering (ESE) research [2]. "This summary attempts to consolidate these ideas into an overall roadmap. As emphasized in the roadmapping introduction, defining a roadmap is an ongoing process and the resulting roadmap needs to be considered a 'living document'. New ideas and changing environments will continue to influence the roadmap, and

consequently, the roadmap will need to be updated periodically to incorporate these new ideas and environments. Each roadmap dimension defines one important aspect of ESE research. The overall ESE roadmap consists of four categories that organize and group together nine different dimensions. The ESE roadmap categories and dimensions are as follows with dimensions indented below categories:

- Maturity
  - *Cohesiveness of field*
  - *Research methodology*
- Coverage
  - *Process/technique/phase*
  - *Problem domain*
  - *Artifact scale*
  - *Subject expertise level*
- Understanding
  - *Evidence*
- Impact
  - *Science/engineering*
  - *Industry"*

The following short description gives a short overview about the next steps of empirical software engineering in the future:

- Category: **Maturity**
  - Dimension: **Cohesiveness of field**
    - **Today(2006):** *Individual research plan, common terminology*
    - **+5 Years:** *Guidelines, Standards*
    - **+10 Years:** *Common research plan, Handbook*
  - Dimension: **Research methodology**
    - **Today(2006):** *Subjecttive views, Isolated techniques, Understand techniques tradeoffs*
    - **+5 Years:** *Integrated techniques, Repeatable methods*
    - **+10 Years:** *Objective framework, standards*
- Category: **Coverage**
  - Dimension: **Process/technique/phase**
    - **Today(2006):** *Single technique/phase, Multiple techniques of phases*
    - **+5 Years:** *Multiple techniques and phases*
    - **+10 Years:** *Comprehensive processes, techniques, and phases*
  - Dimension: **Problem domain**

    - **Today(2006):** *Single artefact, Single project, Single domain*
    - **+5 Years:** *Multiple domains*
    - **+10 Years:** *Comprehensive domain coverage*
  - Dimension: **Artifact scale**
    - **Today(2006):** *Units/components, Subsystems, Small-scale systems*
    - **+5 Years:** *Large-scale systems*
    - **+10 Years:** *System-of-systems*
  - Dimension: **Subject expertise level**
    - **Today(2006):** *Junior expertise, Intermediate expertise*
    - **+5 Years:** *Advanced expertise*
- Category: **Understanding**
  - Dimension: **Evidence**
    - **Today(2006):** *Research conjectures Supportive observations, Common patterns/similarities*
    - **+5 Years:** *Replicated results*
    - **+10 Years:** *Validated theories*
- Category: **Impact**
  - Dimension: **Science/engineering**
    - **Today(2006):** *Subset of Empirical Software Engineering, Empirical Software Engineering*
    - **+5 Years:** *Software Engineering, Computer Science*
    - **+10 Years:** *Sytems Engineering and/or Information Technology*
  - Dimension: **Industry**
    - **Today(2006):** *Isolated examples, Organizational adoption*
    - **+5 Years:** *Multi-organizational adoption*
    - **+10 Years:** *Industry-wide shift*

References

[1] Ebert, C.; Dumke, R. (2007): *Software Measurement: Establish, Extract, Evaluate, Execute.* Springer Pub.

[2] Basili, V. R. et al. (2007). *Emprical Software Engineering Issues – Critcal Assessent and Future Directions*. Springer Publ.

Improving Estimations by Effort Type Proportions

Luigi Buglione

École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest,
Montréal, Canada H3C 1K3

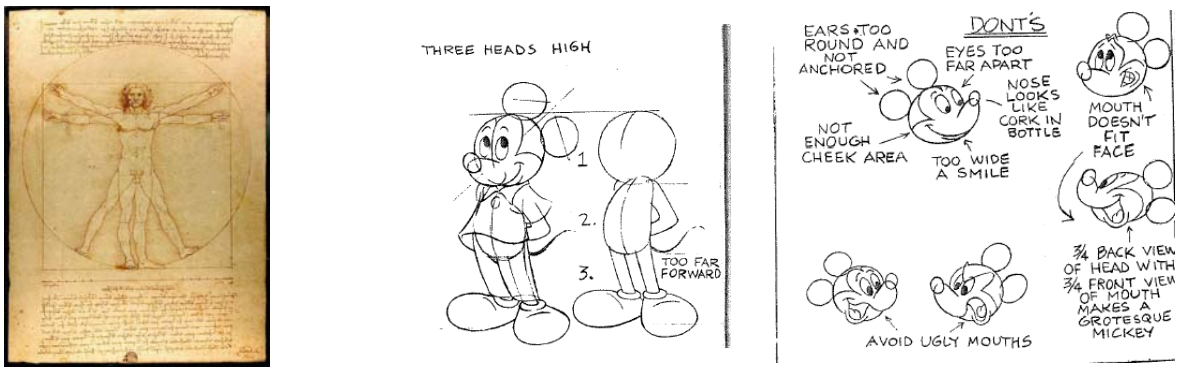
Luigi.Buglione@computer.org

Abstract. *One of the main goals in each human activity, from a cooking recipe up to the project for a building, is to make affordable and reliable estimates. But before being able to count and compute with accuracy the basic entities for obtaining a result, at least estimators have to known the proportions among the resources to use in order to obtain an approximated value to be refined as soon as new and improved information will be available.*
When dealing with software projects, the nature of project requirements from which determining the possible size measures to be used in an estimation process (whatever the technique-approach used) and their approximated percentage in terms of effort can be a significant information in order to establish assumptions for determining the proper estimated value, also by analogy browsing a project historical database or repository, before knowing with a affordable level of certainty the different sizes from which these efforts are derived.
The aim of this paper is to discuss the relevance of proportions across the different requirements classifications and their possible impacts on project productivity and performance at the aim to obtain better estimations.

Keywords – *Proportionality; ISO/IEC 14143-1; F/Q/T; Non-Functional Requirements (NFR); Functional Size Measurement (FSM); Productivity, ISBSG.*

1 Introduction

Estimation is one of the main and more difficult issues to deal with in any field, from a cooking recipe up to the building of a house, due to missing or incomplete information. Thus, estimators at least have to know which could be the series of proportions among the different resources to use for achieving the goal for such estimate.
In the case of a cooking recipe, usually cooking books propose the list of ingredients for four people. Suppose that three more people will have the dinner with us this night: how much more ingredients should be added for obtaining always the expected result? And what about if we would have to cook only for three people? Asking to a very proved and experienced chef, probably the answer would be – more than applying an arithmetical calculation – to determine the right proportions among ingredients. Another example (more obvious) in Visual Arts. The Vitruvian Man (Figure 1a) by Leonardo da Vinci is probably the best and most known expression of what we mean. The harmony for drawing a human body is expressed by a series of proportions among legs, arms and so on. Passing to *cartoons*, the same lesson was observed also for drawing a well-known character as Mickey Mouse (Figure 1b).



(a) (b)

Fig. 1: Lessons learned about proportions: (a) Classic art: the Vitruvian Man (Leonardo da Vinci) [2]; (b) Cartoons: the case for Mickey Mouse (© Disney)

Passing from Visual Arts to Software Engineering, looking that estimation practices are often not so mature, a first-level question – whatever the estimation method used – would be to know at least the proportions among the main resources composing the projects. So, pertaining questions would be firstly: *how much effort should I need for my next project*? Secondly, *how much effort by each possible requirement type should I need for my next project*? This second question is becoming more and more relevant, because the growing complexity of software systems makes more evident the impact of non functional requirements and their related efforts and costs.

This paper aims to discuss this issue, with a particular attention to possible impacts on project productivity and performance, since the inclusion/exclusion of such requirements in a bi-directional traceability chain from requirements to delivery can produce a different level of agreement between the two parties of a contract.

2 Project Effort Estimation: levels of granularity

2.1 Parametric models

About the first question (*how much effort should I need for my next project?*), there are ways to estimate project effort [15]. The typical regression models based on project data are called here as OLS (Ordinary Least Squares Regression) with several possible types of regression. For instance, the generic formula for a linear regression model is:

$$Effort = a + (b * Size)$$

with a <u>product</u> functional size unit used as the independent variable and the <u>project</u> overall effort as the obtained dependent variable. Typical parameters to control the estimation reliability at the project level can be R^2, (Mean Relative Error (MRE) and Standard Error.

2.2 Project Effort Profiles by SLC phase

From the Project Manager's viewpoint, after determining the estimated project effort, next step is to create the WBS and schedule tasks according to project's

requirements, distributing the overall effort across the different Software Life Cycle (SLC) phases in order to choose the right type and amount of people to hire and staff for the new project.

Browsing a project repository and filtering data by one or more criteria can lead to the determination of a cluster of projects where the distribution of data points can present a sufficiently large variability. Even if right now any of the listed repositories proposes the classification effort data by requirement type, other possible classification are yet done. For instance, Déry and Abran [13] analyzed the ISBSG r9 repository [14] and found 32 possible effort *profiles*[1] considering the 2562 projects sized against the IFPUG FSM method[2] and 3 profiles presented more than 100 projects each, identified by the initials of the SLC phases considered: **PSBT** (profile 19, 405 projects, 15.8% of the selected projects), **PSBTI** (profile 20, 350 projects, 13.7% of the selected projects) and **SBTI** (profile 30, 349 projects, 13.6% of the selected projects). For each profile, the average effort distribution per phase against the total effort was presented, providing useful indication about the distribution to be adopted in a next project similar to that one after a regression analysis based on the number of functional size units as the independent variable. Re-applying those filters to ISBSG r10 dataset and eliminating those projects with no phase effort details or with a negative unphased effort, the PSBT profile would include 93 projects, with the following detail[3]:

	UFP	Tot.Eff (hrs)	Plan (abs)	Spec (abs)	Build (abs)	Test (abs)	Unphased (abs)	Plan (%)	Spec (%)	Build (%)	Test (%)	Unphased (%)
Max	16148	73920	17668	13600	35520	16000	19894	70,0%	67,0%	93,0%	74,8%	61,9%
Median	**346,0**	**1350,0**	**172,0**	**140,0**	**813,0**	**250,0**	**39,0**	**7,9%**	**7,3%**	**55,0%**	**18,4%**	**1,8%**
Avg	**816,7**	**6806,2**	**842,6**	**837,6**	**3077,3**	**1220,2**	**828,4**	**11,7%**	**10,3%**	**51,6%**	**18,3%**	**8,2%**
Min	24	280	7	1	30	6	0	0,6%	0,2%	3,2%	1,2%	0,0%

Table 2. Distribution of effort (absolute and %) for IFPUG projects (n=93) – PSBT profile

The first element to note about UFP is the large size range between the largest and the lowest value in the dataset that would suggest to use Pareto analysis for determining more granular sub-clusters. After deriving five sub-clusters, the results – taking care of all the projects – would be these ones (in bold the highest value per SLC phase):

Cluster	UFP	N	Plan (%)	Spec (%)	Build (%)	Test (%)	Unphased (%)
All	24-16148	93	7,9%	7,3%	55,0%	18,4%	1,8%
A	1-150	26	7,1%	7,2%	58,7%	14,8%	2,8%
B	151-300	15	**11,2%**	7,2%	51,6%	18,7%	**10,6%**
C	301-600	21	7,4%	6,9%	55,5%	18,1%	1,8%
D	601-900	11	7,3%	6,7%	**62,2%**	10,6%	1,7%
E	901-	20	10,7%	**10,6%**	35,3%	**21,7%**	1,4%

Table 3. Distribution of effort (median values) for IFPUG projects (all Dev.Types) – PSBT profile

[1] 31 combinations against the five main SLC phases and a further group for those projects having no indication of project phases.
[2] A further filter was the Data Quality Rating (A|B).
[3] The summation of median values could be not always equal to 100%.

where it is possible to observe how the effort devoted to the planning (P) phase is close to 7-8%, as well as the specification (S) phase. The build (B) phase is always the one absorbing more resources (more than 50%) and the testing (T) phase seems to absorb a quite stable amount of effort with a growing demand for largest projects (cluster E) and with less attention to those from cluster D, more concentrated on the build phase.

Another possible question to pose is if there is a difference in the distribution of effort phases according to the project development type (e.g. new development or enhancement). Splitting the original dataset into two ones, Table 3 shows the results for the “new development” projects, while Table 4 for the “enhancement” ones.

Cluster	UFP	N	Plan (%)	Spec (%)	Build (%)	Test (%)	Unphased (%)
All	28-16148	49	7,6%	7,9%	54,5%	18,1%	1,7%
A	1-150	10	6,9%	**12,2%**	46,5%	14,8%	2,0%
B	151-300	10	8,5%	7,3%	53,3%	18,9%	**6,1%**
C	301-500	5	4,2%	7,0%	**81,5%**	5,6%	1,8%
D	501-800	9	5,9%	7,0%	63,2%	12,6%	1,6%
E	801-	7	**10,4%**	10,5%	38,5%	**21,8%**	1,2%

Table 4. Distribution of effort (median values) for IFPUG projects (New Dev. projects) – PSBT profile

Cluster	UFP	N	Plan (%)	Spec (%)	Build (%)	Test (%)	Unphased (%)
All	24-2391	44	8,9%	6,7%	55,7%	18,4%	2,5%
A	1-75	7	5,2%	6,9%	63,0%	**18,9%**	0,0%
B	76-250	12	9,0%	6,0%	58,0%	16,5%	4,0%
C	251-400	6	6,4%	6,1%	**64,2%**	18,5%	**8,9%**
D	401-900	14	10,9%	8,2%	53,6%	18,6%	1,9%
E	901-	5	**25,4%**	**10,8%**	32,2%	10,8%	1,8%

Table 5. Distribution of effort (median values) for IFPUG projects (Enhancement. projects) – PSBT profile

Comparing the two tables, it is possible to observe some differences between new development and enhancement projects. In the first group there is less effort spent on the planning phase, a bit more effort for specifications, more on the build one (with the exception of the Enhancement project for cluster B) and more attention is paid to large new development projects for the testing phase.

But the “Design” or “Specification” SLC phases cover both the functional analysis and the non-functional analysis (i.e. about architectural issues), not necessarily performed by the same analyst in the project team, with the same productivity levels and costs, needed in order to determine the final estimated cost for a project.

Thus, a further question for an estimator could be how to properly determine the right amount of resources to staff in a new project according to effort distribution by SLC phase but also taking care of the nature of the effort type, crossing therefore the two information.

3 Classifying Project Requirements

3.1 The ISO 14143-1 taxonomy (F/Q/T)

The spread and growing diffusion and usage of Functional Size Measurement Methods (FSMM) generated a family of ISO standards – the 14143 series, managed by JTC1/SC7/WG12 – affirming and stating the common underlying principles for a method to be recognized as a *de jure* FSMM. This series comprises 6 parts and Part 1 was about main basic concepts [16]. One of these was that a FSMM necessarily has to consider as input document the so-called FUR (Functional User Requirements), defined as "*a subset of User Requirements (that) describe what the software shall do, in terms of tasks and services*". Such norm initially classified UR into three main types:

- F – Functional: a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirements;

- Q – Quality: any requirements relating to software quality as defined in ISO 9126;

- T – Technical: requirements relating to the technology and environment, for the development, maintenance, support and execution of the software (note: examples of Technical Requirements include programming language, testing tools, operating systems, database technology and user interface technologies)[4].

In the latest update of this standard (February 2007) [3], such taxonomy was simplified taking care only of Functional (**FUR**) and Non-Functional (**NFR**) requirements[5]. As clearly stated both in the ISO 14143 family as well as recognized by IFPUG [5], functional size units (**fsu**) – whatever the method used – represent the expression only of FUR.

3.2 A possible extension (F/Q/T/O)

It is clear that such classification refers basically to *explicit* requirements communicated by the customer. But looking at ISO 9000:2005 standard [6], *quality* is defined as "*all the stated and implicit needs…*". This poses a question about the exhaustivity of the ISO/IEC 14143-1 definition: is it complete or not? What about a difficulty in classifying a requirement? Again, which are approximated proportions among the different efforts related to those main types? This is a very relevant question for those ones involved in estimations, because the quantitative approximation of those quantities represent the step before arriving to a more precise measurement.

[4] A further acknowledgement in such direction is the new upcoming initiative have been started on February 2008 by the IFPUG's IT Performance Committee (ITPC) for a study group on determining the *Technical Size*, whose results will be presented by the end of the year.

[5] NFR therefore correspond to what is not a FUR (the Q/T/O requirement types discussed above).

From a *project* viewpoint, the effort to estimate is included in the WBS boundary, including therefore all the possible activities (tasks) to be performed. Going back from tasks to their related processes and observing those processes to which process group/category they can pertain according to ISO 12207 or CMMI taxonomies, it is possible to observe this kind of mapping, to be refined process by process at the light of their own purposes in case of doubt:

ISO 12207 process groups	Examples (processes)
Primary	Acquisition, Supply, Development, Operation, Maintenance.
Support	Documentation, Configuration Management, Quality Assurance, Verification, Validation, Joint Review, Audit, Problem Resolution.
Organizational	Management, Infrastructure, Improvement, Training.

According to previous definitions, the main group out of the FQT taxonomy would be mainly referred to the ISO 12207's organizational and support processes (and derived tasks in a project WBS). A proposal is to add a fourth category in this requirement taxonomy, referred to the *project* entity and not strictly to the *product* one [1], that could be named "O" (Other/Organizational).

4 Functional vs Non-Functional effort

4.1 Measurable Entities and Impacts on Productivity

As discussed in [8], productivity – usually defined as "*the amount of output created-in terms of goods produced or services rendered - per unit input used*" – that is a CMMI ML2 concept – is included in the more comprehensive concept of performance – definable as "*the degree to which a system or a component accomplishes its designated functions within given constraints*" [9] – that is a CMMI ML4 concept. The reason is due to the inclusion of all the 'real', intangible assets/outcomes produced/generated in the project, not countable and valuable applying the productivity definition. An example of such elements can be easily derived also by the COCOMO cost/scale drivers list (i.e. how to measure team cohesion?), having an indirect impact on productivity, but that – more important – have an impact on the overall project performances. In the current definition of productivity, a series of countable outputs (work products) referable to the software (*product*) with the overall (*project*) effort is related. The result is a not homogeneous ratio, because the two parts of the formula refer to different (even related) entities.

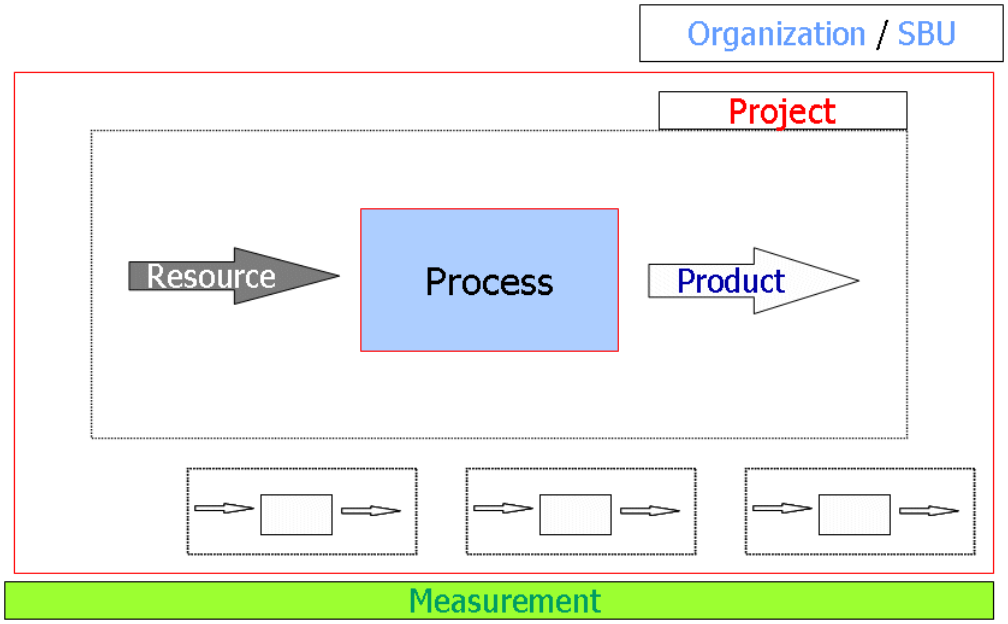


Fig. 2 – STAR taxonomy [7]

So, looking at the bottom part of the formula (worked effort) it would be interesting to understand how it is composed. And retrieving the new ISO 14143-1 requirement taxonomy, at least it would be possible to split the overall project effort among at least functional vs non-functional parts. For instance, how much functional effort does a SAP customization project typically express? And what about the production of a web portal? In such case effort profiles by SLC phase could be not sufficient or provide complete information, where the information need is about the distribution of the project effort by requirement type, as discussed in Section 3. A more complete discussion of possible impacts on schedule and costs in ICT projects using different possible definitions of productivity is discussed in [1].

4.2 Project Effort Profiles by Requirement Types

In order to obtain proper estimations, it is fundamental to filter homogeneous data points from a project repository. And the deletion of some outliers from a first-level query is a usual practice for improving the prediction capability of an estimation model. A low R^2 from a regression analysis could be also due to projects with very different effort types distribution between its functional and non-functional parts. If a project receives some change requests related classifiable as NFR, it will increase the overall project effort, but not increase the number of 'fsu', with the result of an *apparent* reduced productivity. In fact, the new, added effort is just a different type of effort to be spent within the project scope and the number of 'fsu' can be increased only against new functional requirements to be worked.

Again, if a new project will take into account SOA (Service-Oriented Architecture), it would be useful to know how much effort from previous projects was assigned to the 'technical' part, in order to understand the right level of staffing (and related costs) of a software architect with such skills, probably different from the involvement of a generic resource or a functional resource as a programmer. And the same would

happen with the deployment of any other type of NFR (e.g. quality assurance people, configuration manager, etc.)

Therefore, the same kind of analysis presented in Section 2.2 could be replicated using the desired requirement type classification, whatever it is (F/Q/T, F/Q/T/O, F/NF).

4.3 Main External Project Repositories and Effort Data

The following list is about some external project repositories and initiatives for stimulating ICT organizations to gather data for estimation purposes:

- ISBSG [4] for ICT projects, yet introduced in Section 2 for the effort profile by SLC phase. ISBSG is a not-for-profit organization created in 1997 with the mission to help improve the management of IT resources by both business and government through the provision and exploitation of public repositories of software engineering knowledge that are standardized, verified, recent and representative of current technologies. ISBSG aims to fulfill its mission by growing, maintaining and exploiting three repositories of software metrics: (a) Software Development and Enhancement; (b) Software Maintenance and Support; (c) Software Package Acquisition and Implementation.

- PROMISE [10] is a research project for retrieving and redistributing to the ICT community a series of “open sourced” data sets useful for estimation purposes. Currently there are 57 data sets in the following areas: defect predictions (27), effort prediction (9), general purpose (7), model-based software engineering (5) and text mining (9). In order to stimulate more and more the practice of a consistent and focus project data gathering, from 2006 there is also a related conference on such issue.

- TukuTuku [11] is another research project from the University of Auckland with a repository for hypermedia web projects.

Looking at effort data, the maximum detail that usually is gathered and implicitly suggested to be also internally gathered is a classification by SLC phase, adopting a general-purpose classification into three or four phases. This information can help in refining estimates, in particular proving tips about the proper number of FTE (Full Time Equivalent) to staff in a certain SLC phase by project role. A suggestion is to insert in the data gathering questionnaires of such project repositories and in the project’s closure practice specific fields for classifying project’s effort not only by SLC phase, but also by Requirement Type.

4.4 Other Possible Project Profiles

After considering some possible effort distributions, other possible filters that can be adopted, trying to present other project *profiles*. An example could be a size distribution among BFC (Base Functional Components) by FSM method [12][14], that seems to have a better approximation in regression models more than using the single ‘fsu’ value.

As explained, proportions could be useful in order to take decisions, and should be driven by the organization's information needs.

5 Conclusions & Prospects

In every domain, from cooking to visual arts, proportions help people doing estimations in understanding the right amount of resources to consider for a new activity, whatever it is. And the same mechanism should be applied also in software projects. Effort distribution by SLC phase is yet used and considered in most known repositories as the ISBSG one: it helps project managers to know the typical effort profiles according to some projects' characteristics, as well as functional profiles by BFC that allow to obtain more performant estimation models using such details.

The same lesson could be applied when analyzing the different types of requirements composing a software project: it would be sufficient to gather (even if approximately and not with the highest level of precision) not only the classification of project effort by SLC phases but also by requirement type. According to the desired level of granularity, this classification will consider at least two types (functional vs. non-functional), or can be more detailed including a large number of requirement types (e.g. the ISO's F/Q/T taxonomy [3] or the F/Q/T/O one [1]).

The consistent application of such practice in ICT organization can help estimators in having other useful filtering criteria for determining clusters of projects closer and closer to the next project to estimate, whatever the estimation technique adopted, and improving therefore estimates by reducing the probability to obtain high MRE from the start of the estimation process.

References

[1] Buglione, L.: *Some thoughts on Productivity in ICT Projects,* version 1.0, WP-2007-01, White Paper, July 2007, URL: **http://www.geocities.com/lbu_measure/fpa/fsm-prod-100e.pdf**

[2] ---, *Vitruvian Man,* Wikipedia, URL: **http://en.wikipedia.org/wiki/Vitruvian_Man**

[3] ISO/IEC, IS 14143-1:2007, *Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*, February 2007

[4] ISBSG, *Repository r10, January 2007*, URL: **www.isbsg.org**

[5] IFPUG, *Framework for Functional Sizing*, Version 1.0, International Function Point Users Group, White Paper, September 2003

[6] ISO, *IS 9000:2005 – Quality Management Systems – Fundamentals and Vocabulary*, December 2005

[7] Buglione, L.; Abran, A.: *ICEBERG: a different look at Software Project Management*, IWSM2002 in "Software Measurement and Estimation", Proceedings of the 12th International Workshop on Software Measurement (IWSM2002), October 7-9, 2002, Magdeburg (Germany), Shaker Verlag, ISBN 3-8322-0765-1, pp. 153-167, URL: **http://www.lrgl.uqam.ca/publications/pdf/757.pdf**

[8] Buglione, L.; Abran, A.: *A Model for Performance Management & Estimation*, Proceedings of METRICS 2005, 11th IEEE International Software Metrics Symposium, 19-22 September 2005, Como(Italy), ISBN 0-7695-2371-4

[9] IEEE, Standard Glossary of Software Engineering Terminology, IEEE Std 610.12, 1990 (Rev: 2002)

[10] Boetticher, G.; Menzies, T.; Ostrand, T.: *PROMISE Repository of empirical software engineering data,*, West Virginia University, Department of Computer Science, 2007, URL: **http://promisedata.org/**

[11] Mendes, E.; Mosley, N.; Counsell, S.: *Investigating Early Web Size Measures for Web Cost Estimation*, in Proceedings of EASE'2003 Conference, Keele, April, 2003, 1-22, URL: **http://www.cs.auckland.ac.nz/~emilia/publications/EASE2003.pdf**

[12] Abran, A.; Panteliuc, A.: *Estimation Models Based on Functional Profiles*. III Taller Internacional de Calidad en Technologias de Information et de Communications, Cuba, 15-16 February 2007, URL: **http://www.gelog.etsmtl.ca/publications/pdf/831.pdf**

[13] Déry, D.; Abran, A.: *Investigation of the Effort Data Consistency in the ISBSG Repository*, Proceedings of the 15th International Workshop on Software Measurement – IWSM 2005, Montréal (Canada), Shaker Verlag, Sept. 12-14, 2005, pp. 123-136. URL: **http://www.gelog.etsmtl.ca/publications/pdf-presentations/909.pdf**

[14] Gencel C.; Buglione L.: *Do Different Functionality Types Affect the Relationship between Software Functional Size and Effort?*, Proceedings of IWSM/MENSURA 2007, Palma de Mallorca (Spain), November 5-8 2007, pp. 235-246, URL: **http://www.gelog.etsmtl.ca/publications/pdf/1090.pdf**

[15] Briand L.; Wieczorek I.: *Resource Estimation in Software Engineering*, Technical Report, ISERN-00-05, Int. Software Engineering Research Network, 2000, URL: **http://isern.iese.de/network/ISERN/pub/technical_reports/isern-00-05.pdf**

[16] ISO/IEC, IS 14143-1:1998, *Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*, 1998

Büren, G.; Bundschuh, M.; Dumke, R.:

MetriKon 2007 – Praxis der Software-Messung

Shaker Verlag, Aachen, November 2007 (350 Seiten)
ISBN 978-3-8322-6703-2

The book includes the proceedings of the DASMA Metric Conference **MetriKon 2007** held in Kaiserslautern in November, 2007, 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 Metrics News.

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

2. Workshop Bewertungsaspekte serviceorientierter Architekturen (BSOA 2007)

Shaker Verlag, Aachen, November 2007 (132 Seiten)
ISBN 978-3-8322-6716-2

The book includes the proceedings of the 2. Workshop Bewertungsaspekte serviceorientierter Architekturen **BSOA 2007** held in Karlsruhe in November, 2007, 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 Metrics News.

Ebert, C.; Dumke, R.:

Software Measurement

Establish - Extract - Evaluate - Execute

Springer-Verlag Berlin Heidelberg, 2007 (561 Seiten)
ISBN 978-3-540-71648-8

Our world is shaped by software. Since software is so ubiquitous, we need to stay in control. Software measurement is the discipline that assures that we stay in control. In this volume, Ebert and Dumke provide a comprehensive introduction to software measurement. They detail knowledge and experiences about software measurement in an easily understood, hands-on presentation.

Brief references are embedded from world-renown experts such as Alain Abran, David Card, Robert Glass, Peter Liggesmeyer, Charles Symons, and many more. Examples and case studies are provided from Global 100 companies such as Alcatel-Lucent, Atos Origin, Axa, Bosch, Deloitte, Deutsche Telekom, Shell and Siemens.

This combination of methodologies and applications makes the book ideally suited for both professionals in the software industry and for scientists looking for benchmarks and experiences. Besides the many practical hints and checklists readers will also appreciate the comprehensive reference list. Further information, continuously updated, can be found on the book's Web site: **`http://metrics.cs.uni-magdeburg.de/`**.

Jones, C.:

Estimating Software Costs: Bringing Realism to Estimating, Second Edition

Mc Graw Hill Publ., 2007 (644 Seiten)
ISBN: 978-0-07-148300-1

Get a clear, complete understanding of how to estimate software costs, schedules, and quality using the real-world information contained in this comprehensive volume. Find out how to choose the correct hardware and software tools, develop an appraisal strategy, deploy tests and prototypes, and produce accurate software cost estimates. Plus, you'll get full coverage of cutting-edge estimating approaches using Java, object-oriented methods, and reusable components.

- Plan for and execute project-, phase-, and activity-level cost estimations
- Estimate regression, component, integration, and stress tests
- Compensate for inaccuracies in data collection, calculation, and analysis
- Assess software deliverables and data complexity
- Test design principles and operational characteristics using software prototyping
- Handle configuration change, research, quality control, and documentation costs

Selby, R.W.:

Software Engineering

Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research

IEEE Computer Society Publ., 2007 (818 Seiten)
ISBN: 978-0-470-14873-0

This book presents forty-two of Barry W. Boehm's best articles on software engineering, organizes them into nine chapters with newly written summaries by nine of his colleagues, and concludes with a new chapter on Barry's "thoughts for the future." The book chapters address:

- Software Architecture and Quality
- Software Economics

- Software Tools
- Software Process: Early Spiral Model
- Software Risk Management
- Software Process: Emerging Extensions
- Software and Systems Management
- Software Engineering State of the Art and Practice
- Value-Based Software Engineering
- A Software Engineer in the Software Century

This book is recommended as a guide and resource for software engineers, project managers, and technology executives as well as a textbook for advanced undergraduate and graduate courses.

Basili, V.R.; Rombach, D.; Schneider, K.; Kitchenham, B.; Pfahl, D.; Selby, R.W.:

Empirical Software Engineering Issues

Critical Assessment and Future Directions

Springer-Verlag Berlin Heidelberg, 2007 (192 Seiten)
ISBN-10: 3-540-71300X, ISBN-13: 978-3-540-71300-5

This book constitutes the thoroughly refereed post-proceedings of the International Dagstuhl-Seminar on Empirical Software Engineering, held in Dagstuhl Castle, Germany in June 2006. The purpose of this workshop was to identify the progress of empirical software engineering since 1992, to summarize that state-of-the-art in ESE, to summarize the state-of-the-practice in ESE in industry, and to develop an ESE roadmap for research, practice, education and training.

Rud, D.:

Qualität von Web Services - Messung und Sicherung der Performance

VDM Verlag Dr. Müller Saarbrücken, 2006 (201 Seiten)
ISBN-10: 3-86550-666-6, ISBN-13: 978-3-86550-666-5

Web Services stellen eine neue Technologie verteilter Anwendungen dar, welche – dank der Benutzung standardisierter Formate und Protokolle – viele Interoperabilitäts- und Kompatibilitätsprobleme lösen soll, die bei der Verwendung früherer Technologien vorkamen.

Die Dienstgüte (quality of service, QoS) von Web Services hat viele Aspekte, wie z.B. Performance, Skalierbarkeit, Sicherheit usw. In dieser Arbeit steht die Performance im Mittelpunkt.

Im ersten Teil des Buches werden grundlegende Web Service-Technologien sowie ihre Bedeutung im Kontext der Dienstgüte untersucht. Danach folgt eine Analyse von

existierenden mathematischen Modellen, komplexen Managementinfrastrukturen und praktischen Fallstudien auf dem Gebiet der Qualitätssicherung von Web Services.

Der zweite Teil beschreibt einen Web Service-Messservice. Es werden theoretische und technische Aspekte der Messung der Performance von Web Services sowie der Aufbau der wichtigsten Komponente des Messservice – des Lasttreibers – diskutiert. Schließlich werden interne Abläufe und die webbasierte Benutzerschnittstelle des Messservice beschrieben.

Das Buch richtet sich an Praktiker, die sich mit Web Service befassen.

McConnell, S.:

Software Estimation: Demystifying the Black Art

Microsoft Publ., 2006 (308 Seiten)
ISBN: 978-0-7356-0535-0

Often referred to as the “black art” because of its complexity and uncertainty, software estimation is not as difficult or puzzling as people think. In fact, generating accurate estimates is straightforward – once you understand the art of creating them.

In his highly anticipated book, acclaimed author Steve McConnell unravels the mystery to successful software estimation – distilling academic information and real-world experience into a practical guide for working software professionals. Instead of arcane treatises and rigid modelling techniques, this guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and development teams can apply to their projects to help achieve estimation proficiency.

Discover how to:

- Estimate schedule and cost – or estimate the functionality that can be delivered within a given time frame
- Avoid common software estimation mistakes
- Learn estimation techniques for you, your team, and your organization
- Estimate specific project activities – including development, management, and defect correction
- Apply estimation approaches to any type of project – small or large, agile or traditional
- Navigate the shark-infested political waters that surround project estimates

Kandt, R.K.:

Software Engineering Quality Practices

Auerbach Publications, 2006 (256 Seiten)
ISBN 3-8493-4633-9

Software Engineering Quality Practices describes how software engineers and the managers who supervise them can develop quality software in an effective, efficient, and professional manner. This volume conveys practical advice quickly and clearly while avoiding the dogma that surrounds the software profession. It concentrates on what the real requirements of a system are, what constitutes an appropriate solution, and how you can ensure that the realized solution fulfils the desired qualities of relevant stakeholders. The book also discusses how successful organizations attract and keep people who are capable of building high-quality systems.

The author succinctly describes the nature and fundamental principles of design and incorporates them into an architectural framework, enabling you to apply the framework to the development of quality software for most applications. The text also analyzes engineering requirements, identifies poor requirements, and demonstrates how bad requirements can be transformed via several important quality practices.

Lanza, M.; Marinescu, R.:

Object-Oriented Metrics in Practice

Springer-Verlag Berlin Heidelberg, 2006 (205 Seiten)
ISBN-10 3-540-24429-8
ISBN-13 978-3-540-24429-5

Metrics are paramount in every engineering discipline. However, due to its lack of rigor and its intrinsic complexity, software engineering is not considered a classical engineering activity. Moreover, defining, understanding and applying software metrics often looks like an overly complex activity, recommended only to ‘trained professionals’. In general, if a software system is delivering the expected functionality, only few people – if any – care about measuring the quality of its internal structure. Consequently, software metrics are still regarded rather circumspectly by most software developers.

Lanza and Marinescu demystify the design metrics used to assess the size, quality and complexity of object-oriented software systems. Based on a novel approach, backed by generally accepted semantics for metrics and by statistical information from many industrial projects, they deduce a suite of metrics-based patterns for assessing the design of object-oriented software systems. They show in detail how to identify design disharmonies in code, and how to devise and apply remedies.

The combination of theoretically sound results and practically tested procedures and solution paths makes this book an ideal companion for professional software architects, developers and quality engineers. The pattern-oriented description of disharmonies offers easy access to detecting shortcomings and applying solutions to real problems.

Laird, L.M.; Brennan, M.C.:

Software Measurement and Estimation: A Practical Approach

IEEE Computer Society, Wiley Interscience, 2006 (257 Seiten)
ISBN 3-471-67622-5

The text begins with the foundations of measurement, identifies the appropriate metrics, and then focuses on techniques and tools for estimating the effort needed to reach a given level of quality and performance for a software project. All the factors that impact estimations are thoroughly examined, giving you the tools needed to regularly adjust and improve your estimations to complete a project on time, within budget, and at an expected level of quality.

This text includes several features that have proven to be successful in making the material accessible and easy to master:

- Simple, straightforward style and logical presentation and organization enables you to build a solid foundation of theory and techniques to tackle complex estimations
- Examples, provided throughout the text, illustrate how to use theory to solve real-world problems
- Projects, included in each chapter, enable you to apply your newfound knowledge and skills
- Techniques for effective communication of quantitative data help you convey your findings and recommendations to peers and management

Software Measurement and Estimations: A Practical Approach allows practicing software engineers and managers to better estimate, manage, and effectively communicate the plans and progress of their software projects. With its classroom-tested features, this is an excellent textbook for advanced undergraduate-level and graduate students in computer science and software engineering.

Preprints/Technical Reports:

Kunz, M.; Dumke, R.: *Empirische Grundlagen zur COSMIC-FFP-Anwendung für die Aufwandsschätzung,* University of Magdeburg 2007

Mencke, S.; Dumke R.: *Agent-Supported e-Learning*, University of Magdeburg 2007

Frohnhoff, S.; Kehler, K.; Dumke, R.: *Modellbezogene Use-Case-Identifikation für die UCP-basierte Aufwandsschätzung.* University of Magdeburg 2007

see as pdf files:

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

IASTED SE 2008:

IASTED International Conference on Software Engineering 2007
February 12-14, 2008, Innsbruck, Austria
see: **http://www.iasted.org/conferences/cfp-598.html**

SEPG 2008:

20th Software Engineering Process Group Conference
March 17-20, 2008, Tampa, FL, USA
see: **http://gl-spin.org/SEPG08.html**

CSMR 2008:

12th European Conference on Software Maintenance and Reengineering
April 1-4, 2008, Athen, Greece
see: **http://csmr2008.uwaterloo.ca/**

SQS 2008:

Software Quality Systems Conference
April 15-18, 2008, Düsseldorf, Germany
see: **http://www.sqs-conferences.com/de/deutsch/presse/presse_d.htm**

ICSE 2008:

International Conference on Software Engineering
May 10-18, 2008, Leipzig, Germany
see: **http://icse08.upb.de/**

SMEF 2008:

Software Measurement European Forum
May 28-30, 2008, Milan, Italy
see: **http://www.iir-italy.it/smef2008eng/**

SIGMetrics 2008:

ACM SIGMetrics - Performance 2008
June 2-6, 2008, Annapolis, USA
see: **http://www1.cs.columbia.edu/~sigmet08/**

ICPC 2008:

16th International Conference on Program Comprehension
June 10-13, 2008, Amsterdam, Netherlands
see: **http://www.cs.vu.nl/icpc2008/**

PROFES 2008:

9th International Conference on Product Focused Software Process Improvement
June 23-25, 2008, Rome, Italy
see: **http://profes08.uniroma.it/**

WOSP 2008:

6th International Workshop on Software & Performance
June 23-27, 2008, Princeton, NJ, USA
see: **http://www.wosp-conference.org/**

EASE 2008:

11th International Conference on Empirical Assessment in Software Engineering
June 26-27, 2008, University of Bari
see: **http://ease.cs.keele.ac.uk/**

UKPEW 2008:

21th Annual United Kingdom Workshop on Performance Engineering
July 3-4, 2008, London, UK
see: **http://ukpew.org/**

QSIC 2008:

8th International Conference on Software Quality
August 12-13, Oxford, UK
see: **http://cms.brookes.ac.uk/staff/HongZhu/QSIC2008/**

PSQT 2008 North:

International Conference on Practical Software Quality & Testing
September 8-12, 2008, Minneapolis, USA
see: **http://www.PSQTConference.com**

ASQT 2008:

Arbeitskonferenz Softwarequalität und Test 2007
September 10-12, 2008, Klagenfurt, Austria
see: **http://conf.ifit.uni-klu.ac.at/asqt**

QEST 2008:

5rd International Conference on Quantitative Evaluation of SysTems
September 14-17, 2008, Saint Malo, France
see: **http://www.qest.org/qest2008/**

CONQUEST 2008:

11. International Conference on Software Quality
September 24-26, 2008, Potsdam, Germany
see: **http://www.conquest-conference.org/**

ESEM 2008:

International Symposium on Empirical Software Engineering & Measurement
October 9-10, 2008, Kaiserslautern, Germany
see: **http://www.esem-conferences.org/**

UKSMA 2008:

19th Annual UKSMA Conference - Managing your Software (through Measurement)
October 16 , 2008, London, UK
see: **http://www.uksma.co.uk/**

QFD 2008:

20th Symposium on Quality Function Deployment
October 20-31, 2008, Santa Fe, USA
see: **http://www.qfdi.org/**

BSOA 2008:

3. Workshop Bewertungsaspekte service-orientierte Architekturen
November 2008 ,
see: **http://ivs.cs.uni-magdeburg.de/~gi-bsoa/**

IWSM/MetriKon/Mensura 2008:

Common international Conference on Software Measurement
November 18-19, 2009, Munich, Germany
see: **http://iwsm2008.cs.uni-magdeburg.de**

see also: OOIS, ECOOP and ESEC European Conferences

Other Information Sources and Related Topics

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

News Groups

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

Software Measurement Associations

- **http://www.dasma.org**
  DASMA Deutsche Anwendergruppe für SW Metrik und 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.uksma.co.uk**
  UKSMA United Kingdom Software Metrics Association

Software Metrics Tools (Overviews and Vendors)

Tool Listings

- **http://www.cs.umd.edu/users/cml/resources/cmetrics/**
  C/C++ Metrics Tools by Christopher Lott

- **http://mdmetric.com/**
  Maryland Metrics Tools

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

- **http://user.cs.tu-berlin.de/~fetcke/measurement/products.html**
  Tool overview by Thomas Fetcke

- **http://zing.ncsl.nist.gov/WebTools/tech.html**
  An Overview about Web Metrics Tools

Tool Vendors

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

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

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

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

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

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

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

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

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

SOFTWARE MEASUREMENT NEWS

VOLUME 13 2008 NUMBER 1

CONTENTS

ISSN 1867-9196