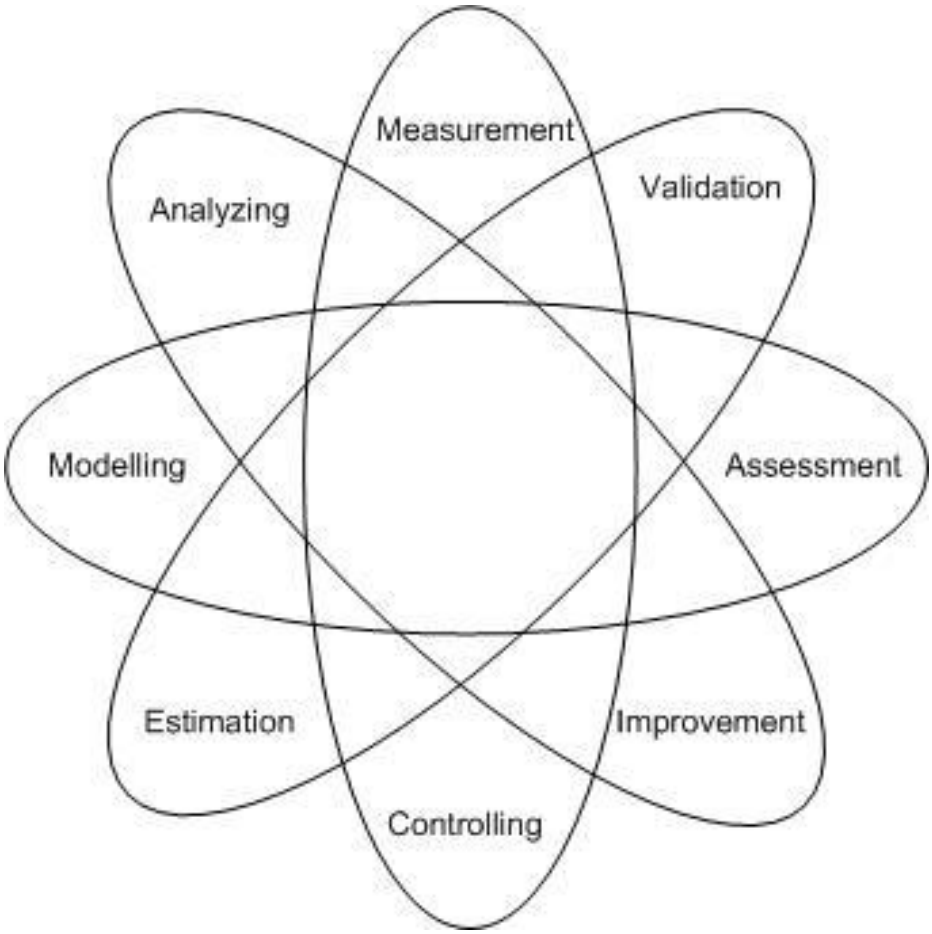
DASMA

Volume 22, Number 2, August 2017

Software Measurement News

Journal of the Software Metrics Community



Editors:

Alain Abran, Reiner Dumke, Christof Ebert, Manfred Seufert, Cornelius Wille

Université du Québec
École de technologie supérieure

COSMIC

SM L@b

CONTENTS

Announcements **3**
IWSM/Mensura 2017 Program, Tagung Software-Messung, ESAPI Workshop, ECC 2018

Community Reports **11**

Historical Views **15**
Citations from Basili et al.: Lessson Learned from 25 years of process improvement 15
Capers Jones: The Origins of Function Point Metrics 17

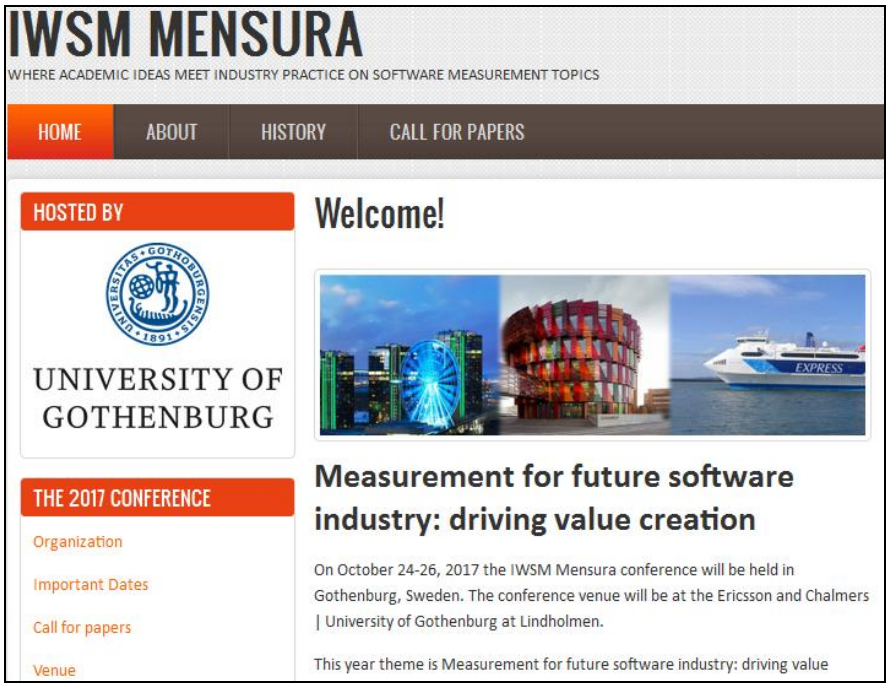
Position Papers..... **21**

Andreas Schmietendorf: An empirical evaluation of Open API approaches 21

New Books on Software Measurement **31**

Conferences Addressing Measurement Issues **36**

Software Measurement in the World-Wide Web **39**



Conference Program (October 25-26, 2017)

25th

	Room 1	Room 2
08:30	Registration + coffee	
09:00	Keynote: Jan Bosch, Software Center	
09:30		
09:45	Coffee	
	Agile/distributed software development I	Software quality
10:00	Ilaria Lunesu, Juergen Muench, Michele Marchesi and Marco Kuhrmann Using Measurement and Simulation for Understanding Distributed Development Processes in the Cloud	Thomas Fehlmann and Eberhard Kranich: Autonomous Real-Time Software & Systems Testing – A Six Sigma Approach
10:30	Ali Idri and Saad Yasser Chadli Identifying and mitigating risks of software project management in global software development	Erdir Ungan, Sylvie Trudel and Luc Poulin: Using FSM Patterns to Size Security Non Functional Requirements with COSMIC
11:00	Wilhelm Meding Effective monitoring of progress of agile software development teams in modern software companies – an industrial case study	Frank Vogelezang, Eric van der Vliet, René Nijland, Jelle de Vries, Eltjo Poort and Harry Mols: A Shortcut to Estimating Non-Functional Requirements?

11:30	Keynote: Karl-Johan Kilius, Ericsson	
12:00	Lunch	
12:30		
13:00	Keynote: Ödgärd Andersson, Volvo Car Group	
13:30		
	Measurement Process Planning	Automation (Gerts)
14:00	Wilhelm Meding: Sustainable Measurement Programs for Software Development Companies - What to measure	Justus Bogner, Stefan Wagner and Alfred Zimmermann: Automatically Measuring the Maintainability of Service- and Microservice-based Systems - a Literature Review
14:30	Regina Hebig and Haoyu Wang: Improving the Real-Time Experience for Software-Measurement System End-Users	Josh Mengerink, Alexander Serebrenik, Ramon Schiffelers and Mark Van Den Brand: Automated Analyses of Model-Driven Artifacts: Obtaining Insights Into Real-Life Application of MDE
15:00	Coffee	
	Functional Size Measurement	Project Management I
15:15	Hassan Soubra and Alain Abran: Functional Size Measurement for the Internet of Things (IoT): An example using COSMIC and the Arduino open-source platform	Francisco Valdés-Souto Earned Scope Management: A Case of Study of Scope Performance using Use Cases as Scope in a Real Project
15:45	Mariem Haoues, Asma Sellami and Hanene Ben-Abdallah: A Rapid Measurement Procedure for Sizing Software Applications based on COSMIC FSM Method	Thomas Fehlmann and Eberhard Kranich: A New Approach for Continuously Monitoring Project Deadlines in Software Development
16:15	Christian Quesada-López, Marcelo Jenkins, Luis Carlos Salas and Juan Carlos Gómez: Towards an automated functional size measurement procedure: An industrial case study	Niklas Mellegård: Using weekly open defect reports as an indicator for software process efficiency -- Theoretical framework and a longitudinal automotive industrial case study
17:00		
18:00	Ship museum	
19:00	Dinner	

	26th	
	Sailor	Cruiser
08:30	Registration + coffee	
08:15	Keynote: José Díaz López, Qlik	
08:45	Gert Frost, Grundfos - Metrics and machine learning	
09:30		
09:45	Coffee	

	Agile/distributed software development II	Software design and coding
10:00	Krzysztof Wnuk and Kalyan Chakravarthy Maddila: Agile and Lean Metrics Associated with Requirements Engineering	Igor Ilin, Anastasia Levina, O. Iliashenko and Alain Abran: Measurement of Enterprise Architecture (EA): Related Work and a Research Agenda for Improved EA Measurement
10:30	Murat Salmanoglu, Tuna Hacaloglu and Onur Demirörs: Effort Estimation for Agile Software Development: Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points	Johannes Bräuer, Reinhold Plösch, Matthias Saft and Christian Körner: Improving Object-Oriented Design Quality: A Portfolio– and Measurement-Based Approach
11:00	Alex Estevam, Denis Dennehy and Kieran Conboy: Adopting Flow Analytics in Software Development Projects	Md. Abdullah Al Mamun, Christian Berger and Jorgen Hansson: Correlations of Software Code Metrics: An Empirical Study
12:00	Lunch	
12:30		
13:00	Keynote: Tom Gilb	

14:00	Industry Talk: Michael Harris, PREMIOS - Visualizing the value of software	
	Software Quality II	Effort Estimation I
14:30	Szymon Kupiński, Bartosz Walter, Marcin Wolski and Jakub Chojnacki: Filling the gaps: imputation of missing metrics' values in quality models	Neslihan Küçükateş Ömüral and Onur Demirörs Effort estimation methods for ERP projects based on function points: a case study
15:00	Arthur Molnar and Simona Motogna: Discovering Maintainability Changes in Large Software Systems	Ali Idri, Mohamed Hosni and Alain Abran Investigating Heterogeneous Ensembles with Filter Feature Selection for Software Effort Estimation
15:30	Coffee	
	Project Management II	Effort Estimation II
15:45	Jukka Ruohonen, Sampsa Rauti, Sami Hyrynsalmi and Ville Leppänen Mining Social Networks of Open Source CVE Coordination	Przemyslaw Pospieszny Software Estimation – towards prescriptive analytics
16:15	Marcus Ciolkowski, Sebastian von Mammen and Simon Faber 3-D Visualization of Dynamic Runtime Structures in Applications	Jean-Marc Desharnais and Alain Abran COSMIC (ISO 19761) Certification: An Analysis of Examinee Results
16:45	Conference closing	

Tagung Software Messung

7. Dezember 2017 in Kaiserslautern

(Kostenlose Registrierung)

Tagungsschwerpunkte:

Welchen Wert kann ein Unternehmen aus Software Messung ziehen? Ist meine Produktivität und Qualität wettbewerbsfähig, und wie messe ich das? Wohin entwickeln sich Metriken mit Industrie 4.0 und Digitalisierung? Die **Software Messung 2017** zu Messung, Schätzung und Bewertung von Software und IT gibt die Antworten.

Termin: **Donnerstag, 7. Dezember, Fraunhofer IESE, Kaiserslautern**.

Ihr Vorteil: Aktuelle internationale Industrie-Benchmarks sowie Impulse für Ihre Arbeit aus der Forschung. Fraunhofer IESE und Vector laden Sie zur Veranstaltung in 2017 ein – **ganz ohne Tagungsgebühren**!

Agenda:

10:00-10:30 Kaffee und Begrüßung

10:30-11:30 K**eynote** – ***Prof. Dieter Rombach***, Fraunhofer IESE:
„Measurement Trends – Die Notwendigkeit guter Kennzahlen“

11:30-13:00 Workshops zu aktuellen Schwerpunkten und Ausrichtung von Software-Messung

13:00-14:00 Mittagessen auf Einladung des Fraunhofer IESE

14:00-16:00 Fachvorträge

- Eberhard Kranich, DASMA: Continuously Monitoring Project Deadlines
- Christof Ebert, Vector: Static Code Analysis – Tools, Evaluation, Practical Usage
- Jens Heidrich, IESE: Q-Rapids – Quality-aware Rapid Software Development
- Andreas Schmietendorf, HWR Berlin: Open APIs – Messung und Empirische Bewertung

16:00-17:00 Workshop-Ergebnisse - Abschlusspräsentation

18-20 Uhr Gemütlicher Ausklang im Brauhaus unter dem Motto „Bier, Benchmarks und Prozente“

Details und kostenlose Registrierung: www.software-measurement.de
Achtung: Ihre Anmeldung ist nötig trotz der kostenlosen Teilnahme.
Lesen Sie auch die zugehörige aktuelle Ausgabe der ***Measurement News:***
https://fg-metriken.gi.de/fileadmin/user_upload/news/news2017-1.pdf

Herzliche Grüße und viel Erfolg in Ihren Projekten,

-Christof Ebert, Vector, Sprecher der GI Fachgruppe 2.1.10, christof.ebert@vector.com
-Reiner Dumke, Universität Magdeburg, stellv. Sprecher
-Jens Heidrich, Fraunhofer IESE, stellv. Sprecher
-Cornelius Wille, Fachhochschule Bingen, stellv. Sprecher

Ankündigung des ESAPI-Workshops

„Evaluation of Service-APIs“

Themenschwerpunkt: *Service APIs als Enabler einer erfolgreichen Digitalisierung*
Ort: 02. November 2017 in Berlin - ***Gastgeber:*** *Deutsche Bahn AG*

Motivation

Im Internet als Service zur Verfügung gestellte Informationen, Funktionen und Algorithmen bestimmen in zunehmendem Maße die Art und Weise wie neue Softwarelösungen implementiert werden. Im technologischen Sinne handelt es sich dabei um webbasiert zur Verfügung gestellte Daten und Service-APIs, die eine „ad hoc“ Integration in neue Lösungskontexte unterstützen. Entsprechende Lösungen finden sich z.B. mit mobilen Applikationen (Apps), Anwendungen der künstlichen Intelligenz (Data Science), im Telematikbereich (Smart City) oder auch im Internet der Dinge (IoT). Wollen Unternehmen von den Möglichkeiten der Digitalisierung profitieren, ist es erforderlich, die eigenen Bedürfnisse (Nutzersicht auf APIs) und Möglichkeiten (Entwicklersicht auf APIs) realistisch bewerten zu können. Darüber hinaus gilt es den gesamten Lebenszyklus der angebotenen Daten und APIs einem Management zu unterziehen. Ein solches API-Management muss sowohl den Bedürfnissen der Serviceentwicklung, einer heute zumeist agil durchgeführten Serviceintegration als auch denen des betrieblichen Einsatzes der dann zusammengesetzten Services entsprechen. Typische Managementfunktionen beziehen sich auf den Lebenszyklus, die Zugriffskontrolle, das Monitoring, die Verrechnung oder auch auf die eingesetzten Laufzeitumgebungen der Service-APIs. Ein besonderes Spannungsfeld ergibt sich aus offen zur Verfügung gestellten Schnittstellen (Open Data/Open API) und ausschließlich kommerziell angebotenen Schnittstellen (API economy). Im Zusammenhang mit der API economy finden sich schnell wachsende Unternehmen (z.B. Fahrdienstvermittler Uber, Unterkunftsvermittler Airbnb, Onlineversand Zalando), die zwar einen breiten Marktzugang besitzen, aber nur eine geringe Ressourcebindung aufweisen.

Alleinstellungsmerkmale dieser die Digitalisierung treibenden Unternehmen beziehen sich auf die optimale Erfüllung die Kundenbedürfnisse. Entsprechend dafür benötigte Leistungen werden von einem globalen Markt unter Verwendung der hier im Mittelpunkt stehenden Service-APIs im Sinne einer agil gebildeten Lieferantenkette bezogen. Dieser Sachverhalt unterstreicht die strategische Bedeutung derart zur Verfügung gestellter Informationen, Funktionen und Algorithmen. Die aus dem Diskurs der Softwaremessung hervorgegangene Interessengemeinschaft widmet sich daraus resultierenden Bewertungsaspekten beim anstehenden Workshop in Berlin, wofür entsprechende Beiträge gesucht werden. Die Initiative ist aus der BSOA/BCloud-Interessensgemeinschaft hervorgegangen. Mit der inhaltlichen Fokussierung wird den Themenschwerpunkten der vergangenen Jahre Rechnung getragen.

Ausgewählte Themenbereiche:

- Bewertung von Service APIs als Wettbewerbsfaktor, im Sinne der Möglichkeiten für innovativer Produkte und Dienstleistungen.
- Kriterienbasierte Erfassung der Auswirkungen von Service APIs auf die Industrialisierung und Agilität unternehmerische Prozessabläufe.

- Bewertungsansätze im Zusammenhang mit der Identifikation, Spezifikation, Bewertung und Qualitätssicherung von Serviceangeboten.
- Gestaltung von Architekturen zur serviceorientierten Verzahnung von unternehmensinternen Lösungen mit Service APIs.
- Herausforderungen der diversifiziert eingesetzter Service APIs im Kontext eines kollaborativen IT-Service-Managements.
- Bewertung von Sicherheits- und Compliance-Anforderungen im Diskurs vielfältig akquirierter und betriebener Service APIs.

Die dargestellten Themen reflektieren nur ausgewählte Aspekte der vielfältigen Herausforderungen im Diskurs der API Economy. 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, Betrieb und Management API-basierten Integrationsarchitekturen tätig sind, werden gebeten, Beiträge im doc- oder pdf-Format über die unten angegebene Webseite einzureichen.
Der Umfang der Beiträge sollte 3000 Wörter nicht übersteigen. Die Formatierungsrichtlinien werden ebenfalls auf der genannten Webseite veröffentlicht. Angenommene Beiträge werden innerhalb eines 30-minütigen Vortrags präsentiert bzw. in Form eines Posters (innerhalb der Workshoppausen) vorgestellt. Alle angenommenen Beiträge des Workshops erscheinen in einem Tagungsband.

Termine

19.09.2017 Einreichung von Beiträgen
28.09.2017 Annahme/Ablehnung
02.10.2017 finales Workshop-Programm
08.10.2017 Abgabe der druckreifen Beiträge
02.11.2017 Workshop in Berlin

Webseite zum Workshop

Weitere Informationen: www.cecmg.de
Paper Submission: https://easychair.org/conferences/?conf=esapi2017evaluationo

Programmkomitee

S. Aier, Universität St. Gallen
F. Balzer, CA Deutschland
M. Binzen, DB Systel GmbH
E. Dimitrov, T-Systems
R. Dumke, Uni Magdeburg
J. M. Gomez, Uni Oldenburg
W. Greis, TPS Data & CECMG
J. Heidrich, Fraunhofer IESE
S. Kosterski, Toll Collect
M. Lother, Robert Bosch GmbH
M. Mevius, HTWG Konstanz
H. Neumann, Deutsche Bahn AG
R. Neumann, Ultra Tendency UG
K. Patzer, T-Systems Dresden
M. Rothaut, T-Systems Bonn
A. Schmietendorf, HWR Berlin
F. Simon, Zurich Insurance Group
F. Victor, FH Köln
C. Wille, FH Bingen
M. Weiß, HUK Coburg
R. Zarnekow, TU Berlin

E-Mail: Andreas.Schmietendorf@hwr-berlin.de

Kontakt zur Initiative

Prof. Dr.-Ing. habil. Andreas Schmietendorf
HWR Berlin, Berlin School of Economics and Law, Fachbereich II
Alt-Friedrichsfelde 60, 10315 Berlin

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

Enterprise Computing Conference - ECC 2018

(save the date)

April 17 - 19 2018, Hamburg

Lindner Park Hotel Hagenbeck
Hegenbeckstraße 150, 22527 Hamburg

Die umfassende Algorithmisierung von geschäftlich, öffentlich und privat genutzten Daten, Funktionen und Prozessen stellt Fach- und Führungskräfte im IT-Management vor enorme Herausforderungen. Im Mittelpinkt des damit einhergehenden Veränderungsprozesses stehen Kreativität und Interaktionsfähigkeit der involvierten Leistungsträger. Unserer nächste ECC-Konferenz bietet mit einer Podiumsdiskussion, Impulsvorträgen, World-Cafè, Seminaren und einem fachfremdem Vortrag dafür einen entsprechenden Arbeitsraum.

Aus inhaltlicher Sicht werden folgende Themen behandelt:

- Agilitätsverständnis von der App bis zum Mainframe,
- API economy und API-Management,
- IT-Security und die neue EU-Datenschutz-Grundverordnung,
- Anwendungen im Data Science.

Darüber hinaus erwarten wir renomierte Keynote-Speaker.

Angefragte Medienpartner:

SIGS DATACOM GmbH, http://www.sigs-datacom.de

dpunkt.verlag GmbH, http://www.dpunkt.de

Handelsblatt-Fachmedien GmbH, http://www.datenschutz-berater.de

Sekretariat: ceCMG Johann-Desch-Str. 20, D-63864 Glattbach,
Tel. +49(0)6021 447623, Fax: +49(0)6021 424961
Email: sekretariat@cecmg.de,
http://www.cecmg.de

Letter from Frank Vogelezang from July 2017

Dear all,

I am very pleased to send you the Annual Report 2016. It is very good to see what we have achieved with the COSMIC method in the past year.

Some of the highlights that I want to share with you:

- International cooperation, next to metrics associations like IFPUG and Nesma, also with:
  - ICEAA, *the International Cost Estimation & Analysis Association,* that is setting up a Software Cost Estimation Body of Knowledge with a certification program
  - NIST, *the US National Institute of Standards and Technology,* that is looking for generally applicable IT Standards
  - COCOMO, *the Constructive Cost Model of the University of Southern California,* that is thinking about including COSMIC in their CoCoMo III model
  - China SPI, *the Chinese organization for Software Process Improvement*, that is adopting the COSMIC standard as a national Chinese standard

- Strong interest from the real-time communities
- World-wide 108 new Entry-level certificate holders
- A guideline has been published on conversion of First Generation methods to COSMIC
- Two new case studies have been published and the Rice Cooker case study has been revised It also contains country reports from 18 countries with some interesting initiatives to promote the COSMIC method:
- Develop a COSMIC Case Study (Tunisia)
- Get COSMIC accepted as a national standard (Mexico) or as part of other accepted methods (USA)
- Host the IWSM conference (Germany) or the next (Sweden)
- Organize a webinar on COSMIC in your own language (Brazil)
- Set up a Special Interest Group (UK and Mexico)
- Small meetings with power users (Brazil)
- Translate COSMIC documents to your language (Italy)
- Write articles on the possibilities of the COSMIC method (Mexico)
- Write COSMIC pages on Wikipedia (Germany)

Please enjoy reading this report. Get inspired by the activities of other members and share this with your networks. Of course this Annual Report is available from the COSMIC website as well.

Best regards,

Frank Vogelezang
President

ANNUAL REPORT 2016

(selected parts)

Executive Summary

- As the COSMIC method is stable and 'future-proof', further work on the method aims to make it as simple and easy to use as possible, and to support new users and uses of the method.
- Whilst the method's use for measuring business software continues to expand, COSMIC is emerging as the favoured method to support real-time software development, notably in the automotive, aerospace and telecom industries.
- To spread the knowledge and the use of the method we started or intensified collaborations with other organizations in the field of software cost estimation, including ICEAA, NIST and COCOMO (USA) and China SPI.
- COSMIC's priorities for the coming years aim for a better integration with agile software development, further development of patterns for early size estimating, and functional size measurement automation."

Method Developments

A new, comprehensive 'Guideline on how to convert 'First Generation' Function Point sizes to COSMIC sizes', v1.0, was published in November

- Two Method Update Bulletins were published to improve the definitions and rules for identifying objects of interest, and to remove a minor ambiguity from the definition of a triggering entry.
- These two 'MUBs' resulted in work by the MPC to update the following Guidelines to bring them into line: On sizing real-time software (v1.1.1, published November)
- On sizing business application software (v1.3, to be published 1Q17)
- Other Guidelines to be updated in 2017 include those on sizing Data Warehouses with an addition for 'Big Data', and a small update will be needed to the Measurement Manual.
- Changes were proposed to ISO to bring the ISO/IEC 19761 standard for COSMIC FSM in line with the current definitions and rules, as part of the five-yearly review process.

Case Studies

An entirely revised and extended v2 of the 'Rice Cooker Case Study' was published in March

- A new case study on 'sizing natural language/UML Use Cases for web and mobile applications' was published courtesy of the University of Sfax, Tunisia
- A new 'ACME Car Hire System' case study illustrating how to measure existing software was supplied courtesy of Capgemini UK, to be published 1Q17.

Cooperation

- COSMIC initiated discussions with IFPUG and Nesma on how the three organizations could collaborate more to promote functional size measurement generally. Fruits of this collaboration are expected in 2017.
- COSMIC gave significant input to the US National Institute of Standards and Technology for its work that has started to standardise certain software measures
- COSMIC joined the initiative of ICEAA and Nesma to set up a curriculum and certification program for Software Cost Estimation.
- In parallel the ICEAA is setting up the SCEBOK: Software Cost Estimation Body of Knowledge to which COSMIC is an active contributor.
- COSMIC, China SPI and Measures LLC have agreed to set up a COSMIC China Chapter to spread the use of the COSMIC method in China.

Furthermore, see the ***country reports*** in the Annual Report 2016 from Australia, Brazil, Canada, China, Ecuador, Germany, India, Italy, Mexico, Netherlands, Poland, Sweden, Switzerland, Tunisia, Turkey, United Kingdom, United States and Vietnam.

Lessons learned from 25 years of process improvement

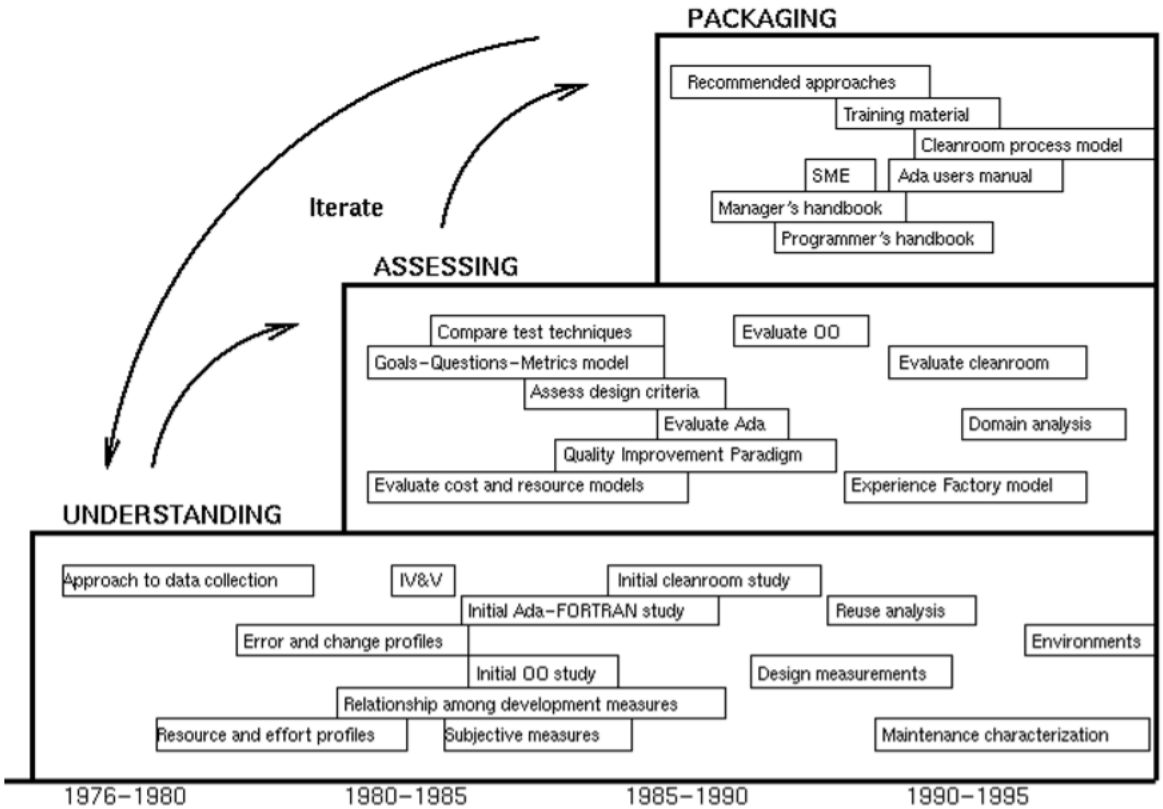
The following paper summarizes essential part of the paper “Lessons learned from 25 years of rpoecess improvement: *The Rise and Fall of the NASA Software Engineering Laboratory”* from Victor R. Basili, Frank E. McGarry, Rose Pajerski and Marvin V. Zelkowitz.
(Proceedings of the Twenty-Fourth International Conference on Software Engineering (ICSE), Orlando, FL, May 2002)

Abstract

*“For 25 years the NASA/GSFC **Software Engineering Laboratory** (**SEL**) has been a major resource in software process improvement activities. But due to a changing climate at NASA, agency reorganization, and budget cuts, the SEL has lost much of its impact. In this paper we describe the history of the SEL and give some lessons learned on what we did right, what we did wrong, and what others can learn from our experiences. We briefly describe the research that was conducted by the SEL, describe how we evolved our understanding of software process improvement, and provide a set of lessons learned and hypotheses that should enable future groups to learn from and improve on our quarter century of experiences.”*

Some SEL Studies

The following figure sows some SEL studies including their motivations, intentions and contents.



Lesson Learned

"We believe the empirical model, based upon the frameworks of QIP, GQM and EF to build an experimental science of software engineering, is the right approach. However, we listed 13 lessons whose impact we could not fully address at various times in the life of the SEL. Solving these should greatly aid future process improvement activities."

Lesson 1: *Data collection requires a rigorous process and professional staff.*

Lesson 2: *You must compromise in asking for only as much information as is feasible to obtain.*

Lesson 3: *Staff training in data collection is a never-ending vigil.*

Lesson 4: *As important as data collection is, it still takes second place to deadlines.*

Lesson 5: *Establishing a baseline of an organization's products, processes, and goals is critical to any improvement program.*

Lesson 6: *The accuracy of the measurement data will always be suspect, but you have to learn to live with it and understand its limitations.*

Lesson 7: *There will always be tension between the need to rapidly feed back information to developers and the need to devote sufficient time to do an analysis of the collected data.*

Lesson 8*: Having a shared commitment over research and development is vital for success.*

Lesson 9: *There is a symbiotic relationship between research and practice in software engineering and both activities gain from the interaction.*

Lesson 10. *Close proximity of researcher to developer aids both.*

Lesson 11: *Having upper management support is important for continued success.*

Lesson 12: *The organization trying to improve their process has to own the improvement process.*

Lesson 13: *It is difficult to make an engineering organization aware of the importance of software engineering to their mission.*

These lessons can be grouped according to several categories:

Need for collecting project data: Lessons 1, 2, 5 and 6,

Need for management buy-in to the process: Lessons 8, 11, 12 and 13,

Need for a focused research agenda: Lessons 9 and 10,

Need for continued staff support: Lessons 3, 4 and 7.

The Origins of Function Point Metrics

Capers Jones

Version 3.0
VP and CTO, Namcook Analytics LLC
Email: Capers.Jones3@gmail.com

Introduction

The author was working at IBM in the 1960's and 1970's and was able to observe the origins of several IBM technologies such as inspections, parametric estimation tools, and function point metrics. This short paper discusses the origins and evolution of function point metrics. In the 1960's and 1970's IBM was developing new programming languages such as APL, PL/I, PL/S etc. IBM executives wanted to attract customers to these new languages by showing clients higher productivity rates.

As it happens the compilers for various languages were identical in scope and had the same features. Some older compilers were coded in assembly language while newer compilers were coded in PL/S, which was a new IBM language for systems software.

When we measured the productivity of assembly-language compilers versus PL/S compilers using "lines of code" (LOC) we found that even though PL/S took less effort, the LOC metric of LOC per month favored assembly language. This problem is easiest to see when comparing products that are almost identical but merely coded in different languages. Compilers, of course, are very similar. Other products besides compilers that are close enough in feature sets to have their productivity negatively impacted by LOC metrics are PBX switches, ATM banking controls, insurance claims handling, and sorts.

To show the value of higher-level languages the first IBM approach was to convert high-level languages into "*equivalent assembly language*." In other words we measured productivity against a synthetic size based on assembly language instead of against true LOC size in the actual higher level languages. This method was used by IBM from around 1968 through 1972.

An IBM vice president, Ted Climis, said that IBM was investing a lot of money into new and better programming languages. Neither he nor clients could understand why we had to use the old assembly language as the metric to show productivity gains for new languages. This was counter-productive to the IBM strategy of moving customers to better programming languages. He wanted a better metric that was language independent and could be used to show the value of all IBM high-level languages. This led to the IBM investment in function point metrics and to the creation of a function-point development team under Al Albrecht at IBM White Plains.

Function Point Evolution

Function Point metrics were developed by the IBM team by around 1975 and used internally and successfully. In 1978 IBM placed function point metrics in the public domain and announced them via a technical paper given by Al Albrecht at a joint IBM/SHARE/Guide conference in Monterey, California.

Table 1 shows the underlying reason for the IBM function point invention based on the early comparison of assembly language and PL/S for IBM compilers. Table 1[1] shows productivity in four separate flavors:

1. Actual lines of code in the true languages.
2. Productivity based on "equivalent assembly code."
3. Productivity based on "function points per month."
4. Productivity based on "work hours per function point."

Table 1: IBM Function Point Evolution Circa 1968-1975
(Results for two IBM compilers)

	Assembly Language	**PL/S Language**
Lines of code (LOC)	17,500.00	5,000.00
Months of effort	**30.00**	**12.50**
Hours of effort	3,960.00	1,650.00
LOC per month	**583.33**	**400.00**
Equivalent assembly	17,500.00	17,500.00
Equiv. Assembly/month	583.33	1,400.00
Function points	100.00	100.00
Function Points/month	**3.33**	**8.00**
Work hours per FP	39.60	16.50

The three rows highlighted in blue show the crux of the issue. LOC metrics tend to penalize high-level languages and make low-level languages such as assembly look better than they really are. Function points metrics, on the other hand, show tangible benefits from higher-level programming languages and this matches the actual expenditure of effort and standard economic analysis. Productivity of course is defined as "***goods or services produced per unit of labor or expense***."

[1] Note: table 1 uses simple round numbers to clarify the issues noted with LOC metrics.

The creation and evolution of function point metrics was based on a need to show IBM clients the value of IBM's emerging family of high-level programming languages such as PL/I, APL, and others. This is still a valuable use of function points since there are more than 3,000 programming languages in 2016 and new languages are being created at a rate of more than one per month.

Another advantage of function point metrics vis a vis LOC metrics is that function points can measure the productivity of non-coding tasks such as creation of requirements and design documents. In fact function points can measure all software activities, while LOC can only measure coding.

Up until the explosion of higher-level programming languages occurred, assembly language was the only language used for systems software (the author programmed in assembly for several years when starting out as a young programmer). With only one programming language LOC metrics worked reasonably well. It was only when higher-level programming languages appeared that the LOC problems became apparent. It was soon realized that the essential problem with the LOC metric is really nothing more than a basic issue of manufacturing economics that had been understood by other industries for over 200 years.

This is a fundamental law of manufacturing economics: "***When a manufacturing process has a high percentage of fixed costs and there is a decline in the number of units produced, the cost per unit will go up.***"

The software non-coding work of requirements, design, and documentation act like fixed costs. When there is a move from a low-level language such as assembly to a higher-level language such as PL/S, the cost per unit will go up, assuming that LOC is the "unit" selected for measuring the product. This is because of the fixed costs of the non-code work and the reduction of code "units" for higher-level programming languages.

Function point metrics are not based on code at all, but are an abstract metric that defines the essence of the features that the software provides to users. This means that applications with the same feature sets will be the same size in terms of function points no matter what languages they are coded in. Productivity and quality can go up and down, of course, but they change in response to team skills.

Current Situation of Function Points

Once function points were released by IBM in 1978 other companies began to use them, and soon the International Function Point User's Group (IFPUG) was formed in Canada. Today in 2016 there are hundreds of thousands of function point users and hundreds of thousands of benchmarks based on function points. There are also several other varieties of function points such as COSMIC, FISMA, NESMA, etc.

Overall function points have proven to be a successful metric and are now widely used for productivity studies, quality studies, and economic analysis of software trends. Function point metrics are supported by parametric estimation tools and also by benchmark studies. There are also several flavors of automatic function point tools. There are also function point associations in most industrialized countries. There are also ISO standards for functional size measurement.

(There was never an ISO standard for code counting and counting methods vary widely from company to company and project to project. In a benchmark study performed for a "LOC" shop we found four sets of counting rules for LOC that varied by over 500%.)

Table 2 shows countries with increasing function point usage circa 2016, and it also shows the countries where function point metrics are now required for government software projects.

Table 2: Countries Expanding Use of Function Points 2016

1	Argentina	
2	Australia	
3	Belgium	
4	**Brazil**	**Required for government contracts 2008**
5	Canada	
6	China	
7	Finland	
8	France	
9	Germany	
10	India	
11	**Italy**	**Required for government contracts 2012**
12	**Japan**	**Required for government contracts 2014**
13	**Malaysia**	**Required for government contracts 2015**
14	Mexico	
15	Norway	
16	Peru	
17	Poland	
18	Singapore	
19	**South Korea**	**Required for government contracts 2014**
20	Spain	
21	Switzerland	
22	Taiwan	
23	The Netherlands	
24	United Kingdom	
25	United States	

Several other countries will probably also mandate function points for government software contracts by 2017. Poland may be next since their government is discussing function points for contracts. Eventually most countries will do this. In retrospect function point metrics have proven to be a powerful tool for software economic and quality analysis.

An empirical evaluation of Open-API approaches

(Discussion paper)

Andreas Schmietendorf

Berlin School of Economics and Law
Email: andreas.schmietendorf@hwr-berlin.de

1. Motivation for Open-APIs

Modern kinds of software applications require an internet-based integration of existing software systems as well as other things of the daily life. The wide diversification of possible data sources creates the foundation for completely new application ideas. Examples can be found among software for connected cars, home automation, weather forecasts or transportation business. Successful enterprises must be able to fit themselves for new market requirements. The cooperation must be software supported and should takes place ad hoc if needed.

Open-API plays an increasing role for the development of innovative software solutions. By the use of an Open-API it is possible to provide access to data and algorithms too. The idea behind this approach can be compared with the open source idea. Under the consideration of [OAI 2017], Open-APIs can be characterized in the following way:

- Open-APIs should be freely usable for everyone.
- Open-APIs should deliver free usable data/algorithms.
- Open-APIs are based on an open standard.

A more holistic approach can be found by the Open Data Manifest (Source: [Bitkom 2017] – translation from German language):

> "Open Data are unfiltered and machine-readable electronic data, that are public to everyone. These data are offered without any kind of binding and earmarking. The access is possible anytime, without compulsory registration and without giving reasons. They are offered ad hoc and free of charge as well as without any kind of limitations for further applications."

2. Compare of Open Data and Open-APIs

Sometimes it's difficult to explain the difference between the term Open Data and Open-API. A very often used interpretation is that Open Data are readable for everyone without specific IT knowledge and Open-APIs are well described service interfaces for software developers. Mostly the use of Open Data is connected with plain old file transfer possibilities. [Richard 2015] proposes the use of Open-APIs in contrast to Open Data and mentioned the following advantages:

- Application of consistent and current data.
- Avoidance of complicated access to the file system.
- Complicate and rarely required processing of datasets in raw format.
- Possibility for using code generators for proxy generation.

From the author's point of view, there exist reasons to use Open Data in a file system oriented manner. Examples can be found in Big Data and Data Science scenario. Within this paper, we want to deal with the software engineering perspective, which means we want to contemplate Open-API and Open Data in the same way. From the technological viewpoint we want to concentrate on HTTP-based RESTful Web-APIs which are documented under the consideration of the OpenAPI (cf. chapter 3) and other kind of "non standard" specifications.

In contrast to the mentioned characteristics and requirements for Open-API offerings (cf. chapter 1), there are some difference in real world implementations. Restrictions can be found with diverse used documentation approaches, necessary registrations, statements about the utilization behaviour or also with the expected scale possibilities.

Currently, a controversial discussion respecting freely provided data interfaces takes place in well established enterprises. In contrast to commercial delivered Web-APIs, the following benefits are expected by the usage of open interfaces (under consideration of [Bitkom 2017]):

- Interest and understanding for customers and third parties.
- Transparency about the collected and used data.
- Possibilities to improve existing products.
- Possibilities for innovations beyond the borders of the enterprise.
- Possibilities for agile implemented software.
- Democratisation of the data ownership.

3. Specification of Open-APIs

A successful Open-API distinguishes itself through a frequent application. The prepared specification (documentation) has a strong influence on the successful use. The early specification approaches for service implementations were driven by the software development communities. Examples can be found with CORBA-IDL (Object Web), WSDL (XML based Web Services) or WADL (XML for REST based Web Services). The core idea behind these technology driven approaches were the requirements to bridge heterogeneous system implementations and to support development tasks (e.g. proxy generation).

Current Open-API offerings consider a resource oriented interface paradigm and use mostly the HTTP-based REST approach. For the specification of those service offerings, the Swagger documentation style was proposed by [SmartBear 2017]. Further approaches can be found with API Blueprint or RAML (see also [Tilkov 2015]). Within this work, these should not be pursued further. Swagger builds the foundation for the OpenAPI[1] approach from the Open API Initiative (short OAI) with the following vision (Source: [OAI 2017]):

> "APIs form the connecting glue between modern applications. Nearly every application uses APIs to connect with corporate data sources, third party data services or other applications. Creating an open description format for API services that is vendor neutral, portable and open is critical to accelerating the vision of a truly connected world."

The OpenAPI specification allows the formal machine-readable interface description of a RESTful service interface by the use of YAML (JSON). YAML provides a simple markup language, human readable and easy to use. With the help of this description the following tasks can be supported:

- Foundation to establish an agreement between client and service provider, perhaps on ad hoc request.

[1] Please consider the difference between OpenAPI as specification an the generic name Open-API

- Generation of mock services, as virtual service endpoint for software test proposes.
- Generation of client and server source code stubs (proxy processes) for different kind of implementation languages.
- Preparation of service descriptions (e.g. HTML- or pdf-based) under the use of corresponding generation tools.
- Support of a dynamic invocation interface to build runtime contracts on request.
- Usage of version and source control, i.e. possibilities to track the history of service changes, too.

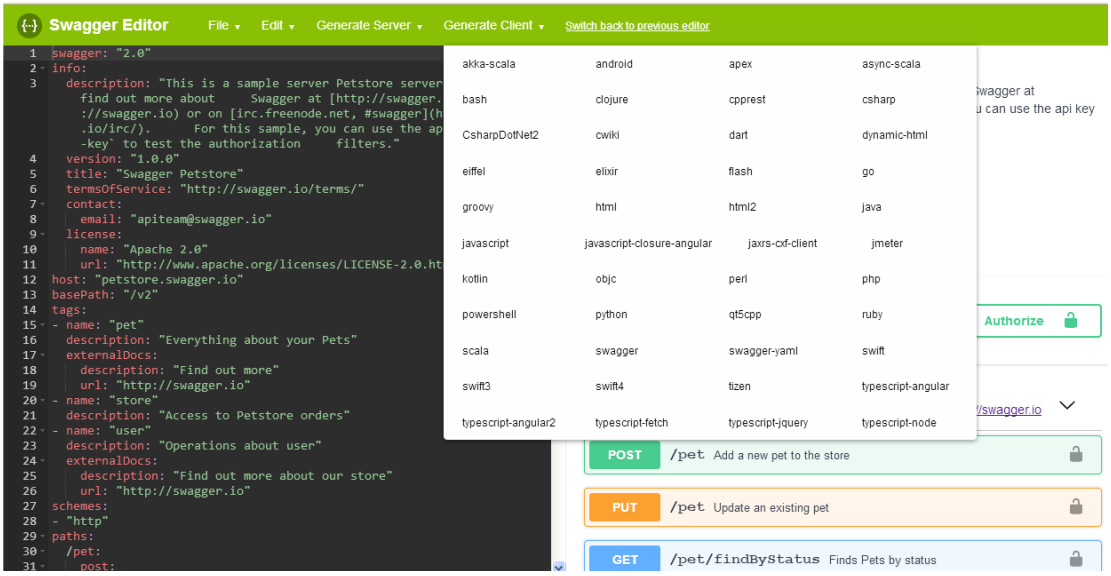


Figure 1: Swagger/OpenAPI Editor (Source: http://editor.swagger.io [Swagger 2017])

Figure 1 depicts the so called Swagger editor. This tool makes the work with OpenAPI specification easier. On the left hand side is a YAML-based service description and on the right hand side selected REST[2] conform HTTP verbs (GET, POST, PUT, DELETE). Every provided web resource consider the same CRUD (CREATE, READ, UPDATE, DELETE) oriented operations. Beside the creation of an OpenAPI specification, the user has the possibility to generate source code stubs for client and server. These Stubs can be generated for different programming languages, e.g. Eiffel, Java, JavaScript, Python or Microsoft's language C#.

4. Selected Open-API examples

This chapter provides a qualitative analysis about some real Open-API offerings. All the offerings were analysed under the use of the same template:

- *Short description.*
- *Location within the internet – URL.*
- *Number of provided APIs.*
- *Prerequisites for the application.*
- *Used technologies.*

[2] Representational State Transfer – architectural style [Fielding 2008]

- *API specification approach and kind of developer support.*
- *Measurements (access rates, availability, supported versions, licences)*

The shown examples of Open-APIs are part of an extensively executed research program. Per year, 100 APIs are analyzed approximately.

4.1 NASA Open-APIs

The American “National Aeronautics and Space Administration“ (NASA) provides data about near earth Asteroids, earth imagery, Mars rover photos or data about current natural events etc.

- URL: https://api.nasa.gov
- Number of Open-APIs: 12
- Explore and test: free
- Intensively use: a developer key is required
- Technology: RESTful web service (HTTP, JSON, XML)
- Specification: some with Swagger 2.0, extra documentation
- Provided measurements: yes – limited number of request per hour (days)
- Service availability: no statement

There are possibilities to publish own Open-API. Therefore, it is necessary to contact the NASA via email or by a GitHub request.

4.2 Island Open-APIs

The “Island Open-APIs” provides data about earthquake monitoring, international flights, television schedules or weather forecast.

- URL: https://docs.apis.is
- Number of Open-APIs: 20
- Explore and test: free
- Intensively use: free
- Technology: RESTful web service (HTTP, JSON)
- Specification: request/respond examples and some JQuery demonstrations
- Provided measurements: no
- Service availability: all time (however tests showed down times)

The information are scraped from various websites, the corresponding source code is available via GitHub. Currently, no version control possibilities exist.

4.3 World Bank Open-APIs

The World Bank provides APIs about time series data, data on the World Bank's operations and financial data.

- URL: https://datahelpdesk.worldbank.org/knowledgebase/topics/125589-developer-information

- Number of Open-APIs: 3
- Explore and test: free
- Intensively use: free (local caching is recommended)
- Technology: RESTful web service (HTTP, JSON, Atom, RDF)
- Specification: request/respond examples (hints for developers)
- Provided measurements: no
- Service availability: “no guarantee for 100%” - proposal for caching

The Open-APIs can be tested by the use of an “interactive API query builder”, results are shown as JSON-representations.

4.4 Deutsche Bahn Open-APIs

The “Deutsche Bahn Open-APIs” provides data about train delays, timetables, private transport sharing offers etc.

- URL: https://developer.deutschebahn.com
- Number of Open-APIs: 10 (27 data sets)
- Explore and test: after registration (request per minutes - selectable)
- Intensively use: after registration (request per minutes - selectable)
- Technology: RESTful web service (HTTP, JSON)
- Specification: Swagger for each API
- Provided measurements: API related the access rate
- Service availability: unknown (beta-version)

There is an authorization required for all service requests (bearer access token). The offered APIs are available in version 1 (except one Open-API).

4.5 Lufthansa Open-APIs

The “Lufthansa developer network” provides data about flight schedules, flight status, lounges, price offers etc.

- URL: https://developer.lufthansa.com
- Number of Open-APIs: 2 (public API and partner API)
- Explore and test: after registration
- Intensively use: after registration
- Technology: RESTful web service (HTTP, JSON)
- Specification: Swagger/JSON (available via github.com) – see Annex A
- Provided measurements: limited number of request per seconds (hour)
- Service availability: unknown (contract related)

Under the consideration that flight bookings were initiated through a partner, Lufthansa propose a shared revenue model. For all service requests an OAuth 2.0 authorisation is required.

4.6 European data portal

A very interesting approach is available from the European Union. There are datasets referenced for different categories, like transport, energy, finance, education or legal. Annex B shows some meta data analysis, like the most popular data formats, number of data sets in relation to countries or the mostly used license models (status: September 2017).

- URL: https://www.europeandataportal.eu
- Number of Open-APIs: 750.000 data sets (some as APIs, mostly as files)
- Explore and test: Depends on the specific supplier
- Intensively use: Depends on the specific supplier
- Technology: mostly file based (*.csv format)
- Specification: Depends on the supplier (rarely with OpenAPI/Swagger)
- Provided measurements: Depends on the supplier
- Service availability: Depends on the supplier

Under consideration of the RDF-descriptions, there is the possibility for the search of the metadata (SPARQL endpoint).

There exists a 2nd approach for access to open data produced by EU institutions and bodies. The so called “EU Open Data Portal” provides 10949 data sets and can be reached under the URL: https://data.europa.eu. (cf. Annex B – figure B6)

5. Conclusions

The introduced Open Data and Open-APIs should stimulate the discussion about required specification as well as evaluations and measurement approaches. As show in the previous section there are many data sets available but few Open-API related style. Current problems for the software developers deal with the finding problem, heterogeneous documentation styles and informal statements about the quality behaviour of provided data sets or APIs. Statements about the quality of an Open-API have to consider software development aspects and the regular operation. There are diverse approaches used for the documentation of Open-APIs. As mentioned by [Fielding 2004], a REST-based API must be hypertext driven, this is valid for the specification, too. The use of a Swagger/OpenAPI-oriented specification is currently very limited.

6. References

[Bitkom 2017] Open Data Manifest – Positionspapier, Bitkom e.V., 27.04.2017, URL: https://www.bitkom.org/Bitkom/Publikationen/Open-Data-Manifest.html (last accessed 10 September 2017)

[Fielding 2000] Fielding, R.T.: Architectural styles and the design of network-based software architectures. Dissertation, University of California, Irvine, 2000

[Fielding 2004] Fielding, R.T.: REST APIs must be hypertext-driven, 20 October 2008, http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven (last accessed 10 September 2017)

[OAI 2017] Open API Initiative, The Linux Foundation, URL: https://www.openapis.org, (last accessed 10 September 2017)

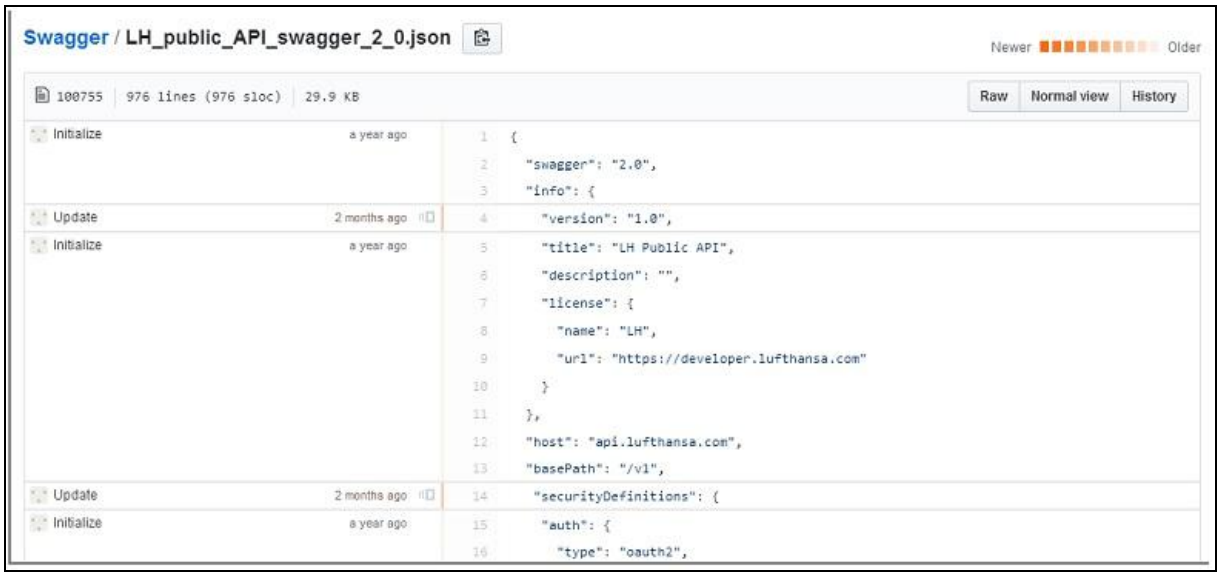
[Richard 2015] Richard, K.: Open Data gets better with Open APIs, August 2015, URL: https://wiredcraft.com/blog/open-data-open-apis, (last accessed 10 September 2017)

[Schmietendorf 2016] Schmietendorf, A.; Nadobny, K.; Hentschel, J.: Design Guidelines zur konstruktiven Qualitätssicherung von Web-APIs, SQ Magazin: Ausgabe 40, ASQF, S. 18-19, September 2016

[SmartBear 2017] SmartBear Software, Somerville, MA, URL: https://swagger.io (last accessed 10 September 2017)

[Tilkov 2015] Tilkov, S.; Eigenbrodt, M.; Schreier, S.; Wolf, O.: REST und HTTP (3. Aufl.), dpunkt.verlag, Heidelberg 2015

Annex A: Swagger file of Lufthansa Open-APIs



Source: https://github.com/LufthansaOpenAPI/Swagger/blob/master/LH_public_API_swagger_2_0.json

Annex B: Empirical Aspects of EU Open Data/Open-APIs

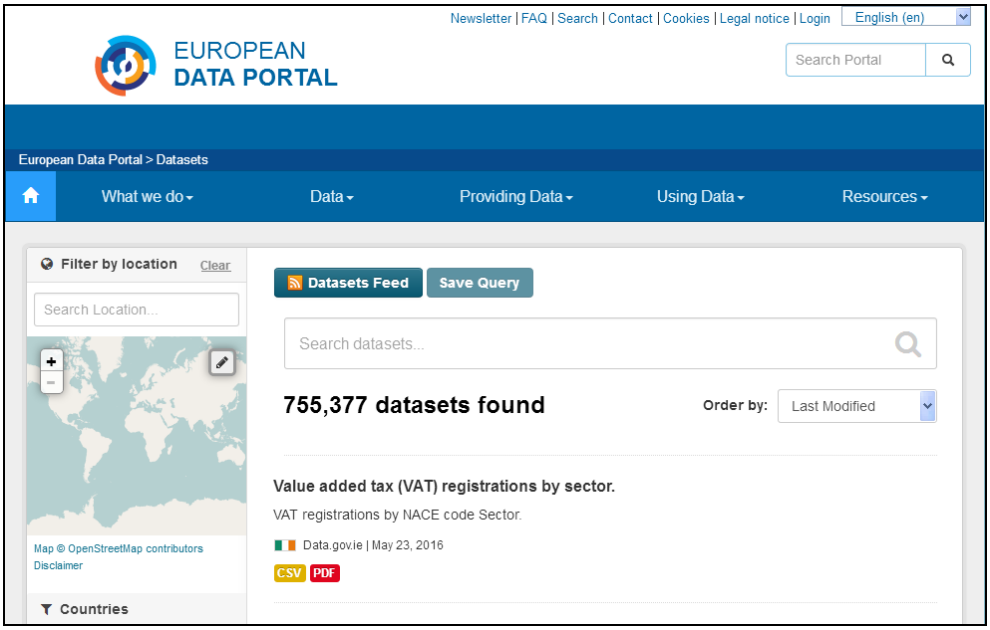


Figure B1: *Screenshot of the European data portal (last access: September 2017)*

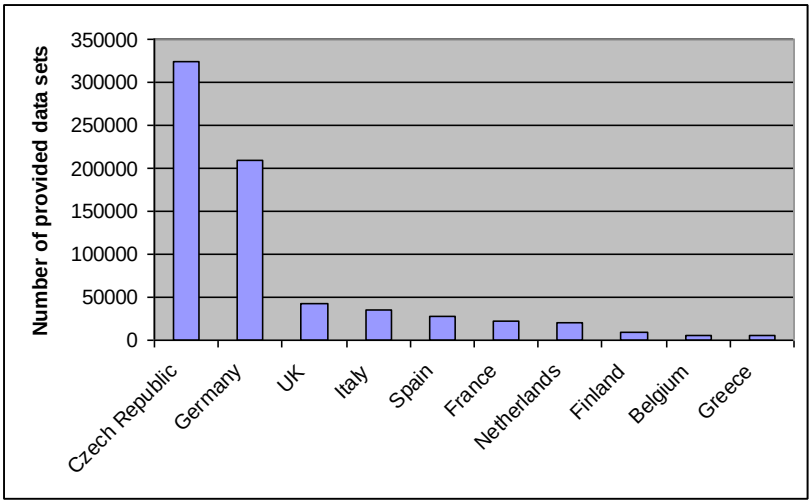


***Figure B2**: Number of provided data sets in relation to counties*

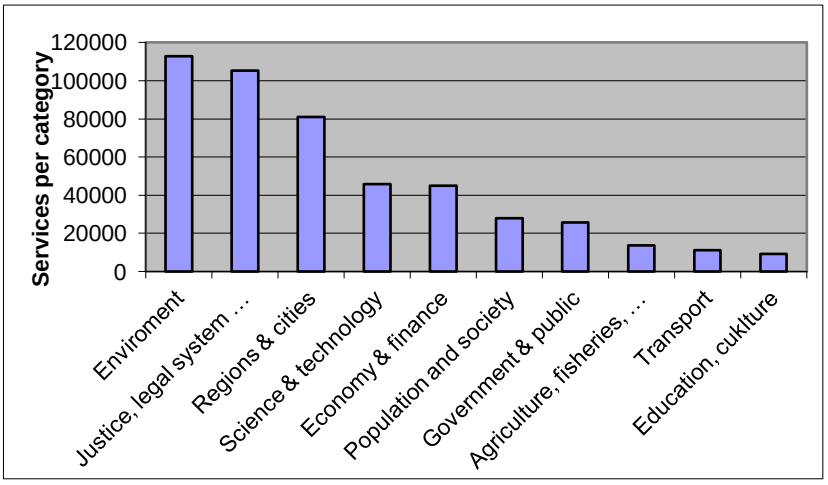


***Figure B3**: Number of provided data sets in relation to supported categories*

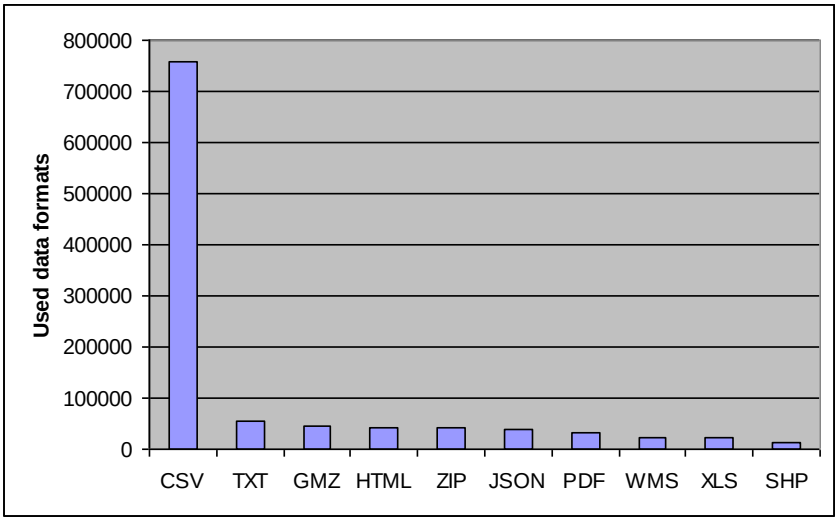


***Figure B4:** Number of provided data sets in relation to the used data format*

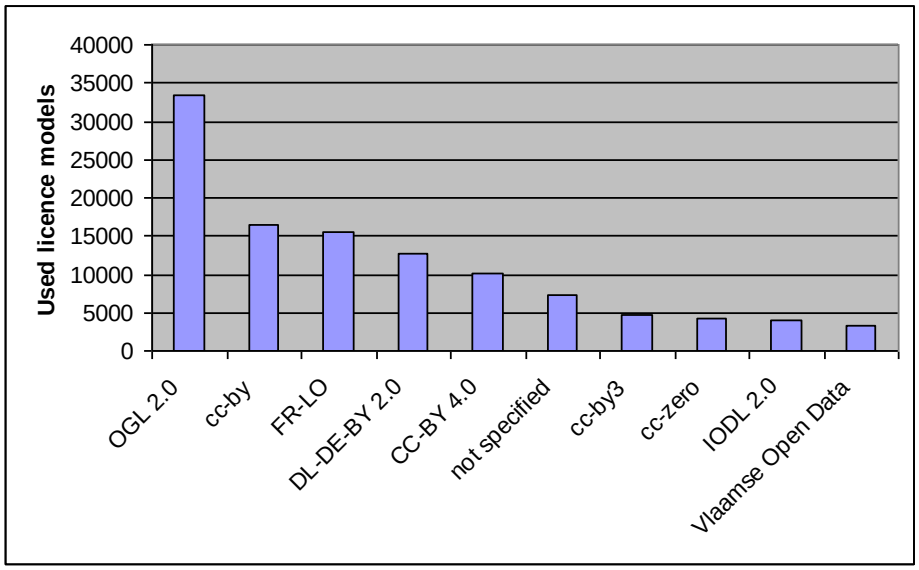


***Figure B5**: Number of provided data sets in relation to the used licence models*

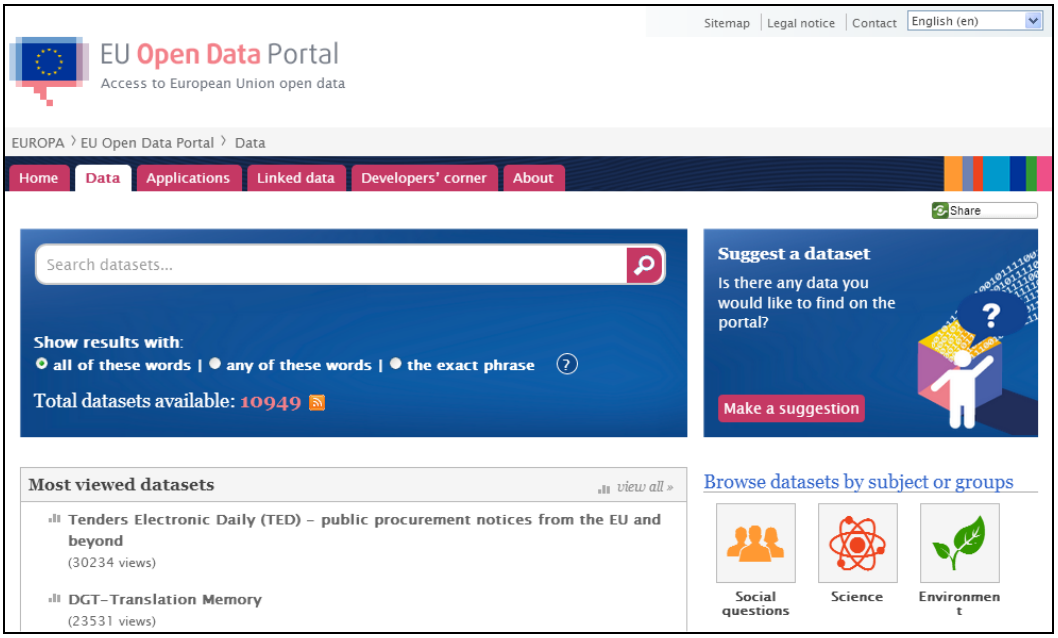


***Figure B6**: Screenshot of the EU Open Data Portal*

Source: http://data.europa.eu/euodp/en/home (last access 10 September 2017)

Capers Jones:

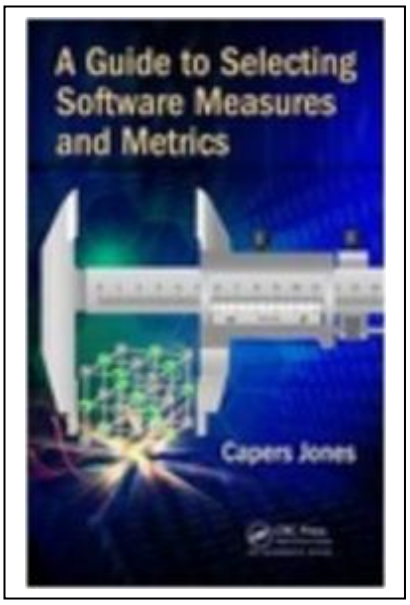
Guide to Selecting Software Measures and Metrics

Auerbach Publications, 2017, 358 p.
ISBN 978-1138-033078

The book helps software project managers and developers uncover errors in measurements so they can develop meaningful benchmarks to estimate software development efforts. It examines variations in a number of areas that include:

- Programming languages
- Development methodology
- Software reuse
- Functional and nonfunctional requirements
- Industry type
- Team size and experience

Filled with tables and charts, this book is a starting point for making measurements that reflect current software development practices and realities to arrive at meaningful benchmarks to guide successful software projects.



Heidrich, J.; Vogelezang, F.:

IWSM/Mensura 2016

Joined Conference of the 26th International Workshop on Software Measurement (IWSM) and the 11th International Conference on Software Process and Product Measurement (Mensura), IEEE Computer Society, CPS, http://www.computer.org/cps, 2016

This proceedings are available at the IEEE online publishing service.

Schmietendorf, A.; Simon, F.:

BSOA/BCloud 2016

11. Workshop Bewertungsaspekte serviceorientierter Architekturen
3. November 2016, Berlin

Shaker Verlag, Aachen, 2016 (112 Seiten), ISBN 978-3-8440-2108-0

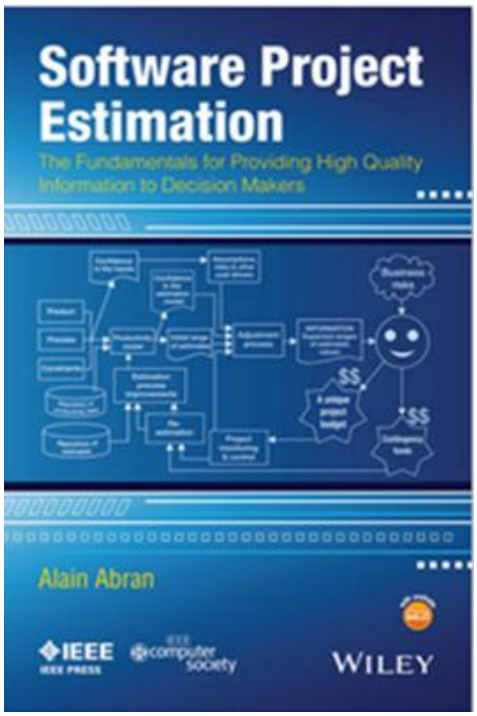
The book includes the proceedings of the BSOA/BCloud 2016 held in Berlin in November 2015, which constitute a collection of theoretical studies in the field of measurement and evaluation of service oriented and cloud architectures.

Abran, A.:

Software Project Estimation: The Fundamentals for Providing

High Quality Information to Decision Makers

Wiley IEEE Computer Society Press, 2015 (288 pages), ISBN 978-1-118-95408-9

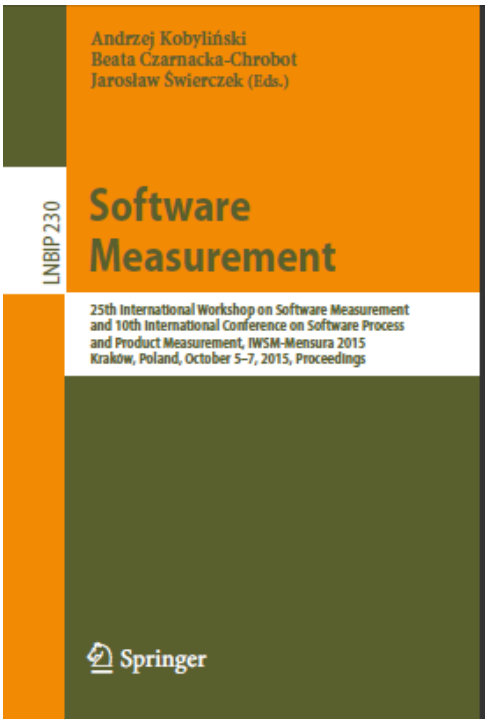


This book introduces theoretical concepts to explain the fundamentals of the design and evaluation of software estimation models. It provides software professionals with vital information on the best software management software out there.

- End-of-chapter exercises
- Over 100 figures illustrating the concepts presented throughout the book
- Examples incorporated with industry data

Andrzej Kobylinski, Beata Czarnacka-Chrobot, Jaroslaw Swierczek

IWSM/Mensura 2015



25th International Workshop on Software Measurement and 10th International Conference on Software Process and Product Measurement, Krakow, Poland, October 5-7, 2015

This book includes some chosen papers of the measurement conference in the LNBP Springer seiries.

Seufert, M.; Ebert, C, Fehlmann, T.; Pechlivanidis, S.; Dumke, R. R.:

MetriKon 2015 - Praxis der Softwaremessung

Tagungsband des DASMA Software Metrik Kongresses
5. - 6. November 2015, IBM, Köln

Shaker Verlag, Aachen, 2015 (272 Seiten)

The book includes the proceedings of the MetriKon 2015 held in Cologne in November 2015, which constitute a collection of theoretical studies in the field of software measurement and case reports on the application of software metrics in companies and universities.

Schmietendorf, A. (Hrsg.):

Eine praxisorientierte Bewertung von Architekturen und Techniken für Big Data

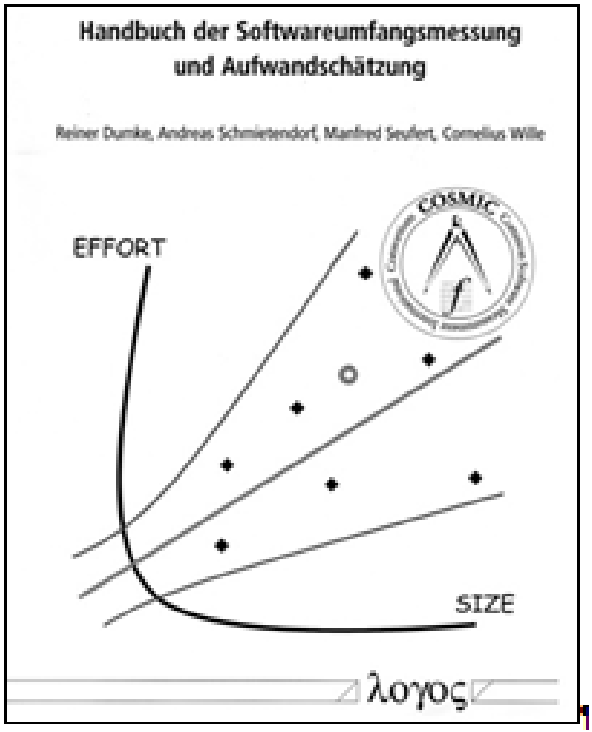
(110 Seiten) Shaker-Verlag Aachen, März 2015 ISBN 978-3-8440-2939-0

This book describes the system aspects of Big Data software infrastructures form a industrial/ practical point of view.

Dumke, R., Schmietendorf, A., Seufert, M., Wille, C.:

Handbuch der Softwareumfangsmessung und Aufwandschätzung

Logos Verlag, Berlin, 2014 (570 Seiten), ISBN 978-3-8325-3784-5

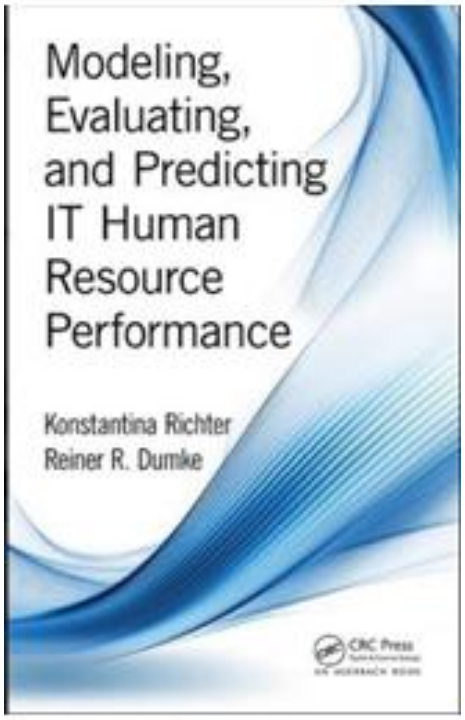


This book shows an overview about the current software size measurement and estimation approaches and methods. The essential part in this book gives a complete description of the COSMIC measurement method, their application for different systems like embedded and business software and their use for cost and effort estimation based on this modern ISO size measurement standard.

Konstantina Richter, Reiner Dumke:

Modeling, Evaluating and Predicting IT Human Resource Performance

CRC Press, Boca Raton, Florida, 2015 (275 pages)



This book explains why it is essential to account for the human factor when determining the various risks in the software engineering process. The book presents an IT human resources evaluation approach that is rooted in existing research and describes how to enhance current approaches through strict use of software measurement and statistical principles and criteria.

Software Measurement & Data Analysis Addressed Conferences

July 2017

LNCS 2017:	**Eight International Symposium on Software Quality** July 3 - 6, 2017, Trieste, Italy see: http://sq.covenantuniversity.edu.ng/?utm_source=researchbib
SAM Summit 2017:	**Conference on Software Asset Management** July, 10 - 12, 2017, Chicago, USA see: http://www.ecpmedia.com/samsummit.html
ICDM 2017:	**IEEE International Conference on Data Mining** July 12 - 16, 2017, New York, USA see: http://www.data-mining-forum.de/
SERP 2017:	**15th International Conference on Software Engineering Research and Practice** July 17 - 20, 2017, Las Vegas, Nevada, USA see: http://americancse.org/events/csce2017/conferences/serp17
DMIN'17:	**12th International Conference on Data Mining** July 17 - 20, Las Vegas, USA see: http://americancse.org/events/csce2017/conferences/dmin17
ICOMP'17:	**18th International Conference on Internet Computing and internet of Things** July 17 - 20, Las Vegas, USA See: http://americancse.org/events/csce2017/conferences/icomp17
ABDA'17:	**4th International Conference on Advances in Big Data** July 17 - 20, Las Vegas, USA see: http://americancse.org/events/csce2017/conferences/abda17
EEE'17:	**4th International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government** July 17 - 20, Las Vegas, USA see: http://americancse.org/events/csce2017/conferences/eee17
GCC'17:	**13th International Conference on Grind, Cloud, and Cluster Computing** July 17 - 20, Las Vegas, USA see: http://americancse.org/events/csce2017/conferences/gcc17
MCCSIS 2017:	**International Conference on Big Data Analytics, Data Mining and Computational Intelligence** July 21 - 23, 2017, Lisbon, Portugal see: http://bigdaci.org/

ICSOFT 2017:

12th International Conference on Software and Data Technologies
July 24 - 26, 2017, Madrid, Spain
see: http://www.icsoft.org/

August 2017

AGILE 2017:

Annual North American Agile Conference
August 7 - 11, 2017, Orlando, FL, USA
see: https://www.agilealliance.org/agile2017/

Euromicro DSD/ SEAA 2017:

Software Engineering & Advanced Application Conference
August 30 - September 1, 2017, Vienna, Austria
see: http://dsd-seaa2017.ocg.at/index.html

September 2017

ESEC/FSE 2017:

European Software Engineering Conference and **Symposium on the Foundation of Software Engineering**
September 3 - 8, 2017, Paderborn, Germany
see: http://esec-fse17.uni-paderborn.de/

QEST 2017:

14th International Conference on Quantitative Evaluation of Systems
September 5 - 7, 2017, Berlin, Germany
see: http://www.qest.org/qest2017/

RE 2017:

24th IEEE International Requirement Engineering Conference
September 4 - 8, 2017, Lisbon, Portugal
see: http://re2017.org/

EuroAsiaSPI² 2017:

24th European Systems & Software Process Improvement and Innovation Conference,
September 5 - 8, 2017, Ostrava, Czech Republic
see: http://2017.eurospi.net/

October 2017

ODSC 2017:
Open Data science Conference Europe
October 13 - 14, 2017, London, UK
see: https://www.odsc.com/london

IWSM-MENSURA 2017:
Common International Conference on Software Measurement
October 24 - 26, 2017, Gothenburg, Sweden
see: http://www.iwsm-mensura.org/

ASE 2017:
Automated Software Engineering
October 30 - November 4, 2017, Urbana-Champain, Illinois
see: http://www.ase2017.org/

November 2017

CSEE&T 2017:
30th Conference on Software Engineering Education and Training
November 7 - 9, 2017, Savannah, Georgia
see: http://www.cseet2017.com/index.html

BSOA/BCloud 2017:
12. Workshop Bewertungsaspekte Service-orientierter und Cloud-Architekturen
November, 2017, Berlin, Germany
see: http://www-ivs.cs.uni-magdeburg.de/~gi-bsoa/

ASQT 2017:
Arbeitskonferenz Softwarequalität, Test und Innovation
November 9 - 10, 2017, Graz, Austria
see: http://www.asqt.org/

ESEM 2017:
11th International Symposium on Empirical Software Engineering & Measurement
November 9 - 10, 2017, Toronto, Canada
see: https://sravyapolisetty.github.io/ESEM/cfp.html

December 2017

PROFES 2017:
International Conference on Product Focused Software Process Improvement
November 29 - December 1, 2017, Innsbruck, Austria
see http://www.profes-conferences.org/

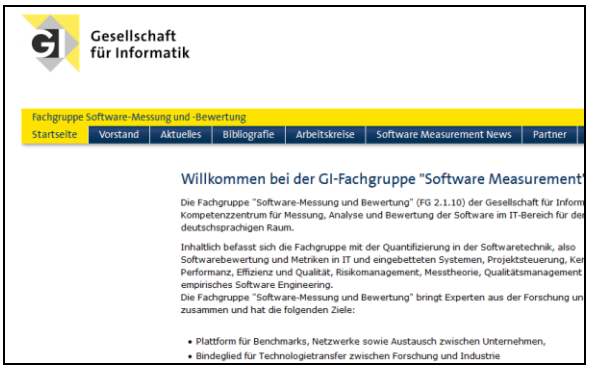
see also: Conferences Link of **Luigi Buglione (**http://www.semq.eu/leng/eveprospi.htm**)**

COMMUNITIES

GI-Fachgruppe Software-Messung und Bewertung

http://fg-metriken.gi.de/

(Measurement News Online)



Common Software Measurement International Consortium

http://cosmic-sizing.org

Deutschsprachige Anwender-gemeinschaft für Software-Metrik und Aufwandschätzung

http://www.dasma.org

International Software Bench-marking Standard Group (ISBSG)

https://www.isbsg.org

Central Europe Computer Measurement Group (ceCMG)

http://www.cecmg.de

Metrics Association's International Network (MAIN)

http://*www.mai-net.org*

Finnish Software Measurement Association (FISMA)

http://www.fisma.fi/in-english/

Netherlands Software Metrics users Association (NESMA)

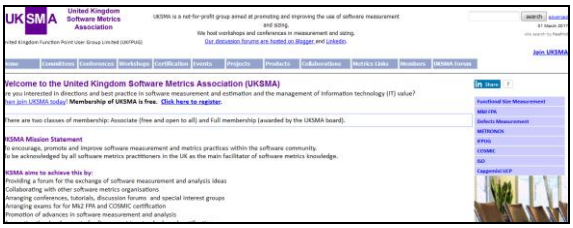
http://www.nesma.org/

Asociacion Espanola de Metricas de Software

http://www.aemes.org/

United Kongdom Software Metrics Association (UKSMA)

http://www.uksma.co.uk



Gruppo Utenti Function Point Italia - Italian Software Metrics Association (GUFPI - ISMA)

http://www.gufpi-isma.org

Anwenderkonferenz Software-qualität und Test (ASQT)

http://www.asqt.org

MEASUREMENT SERVICES

Software Measurement Laboratory (SML@b)

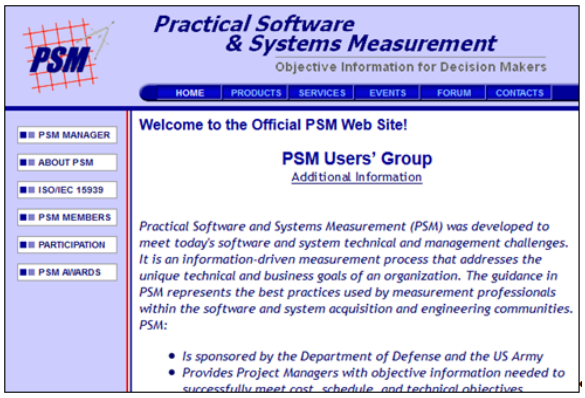
http://141.44.17.27/cms/index.php/en/home/forschung/106-smlab

International Function Point Users Group (IFPUG)

http:www.ifpug.org

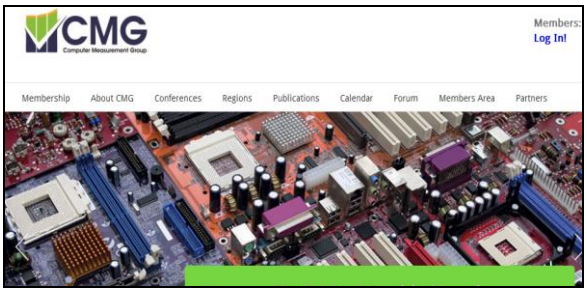
Practical Software & Systems Measurement

www.psmsc.com/:



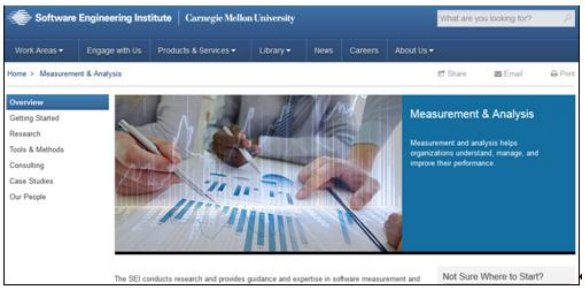
Computer Measurement Group (CMG)

http://www.cmg.org



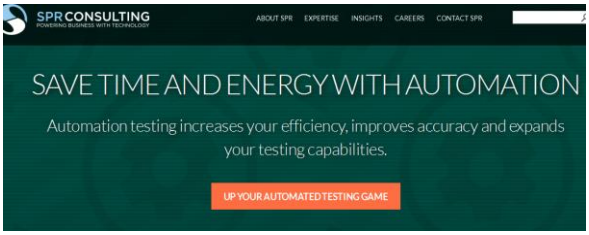
Software Engineering Institute (SEI)

www.sei.cmu.edu/measurement/



Software Productivity Research (SPR)

•*http://www.spr.com/*



McCabe & Associates

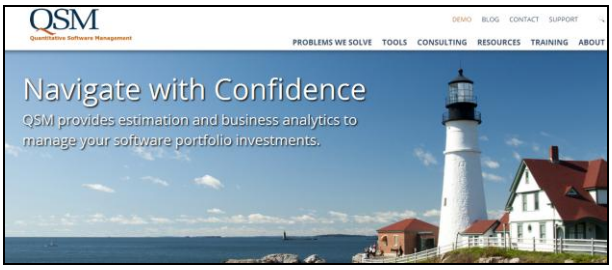
http://www.mccabe.com

SQS Gesellschaft für Software-Qualitätssicherung

http://www.sqs.de

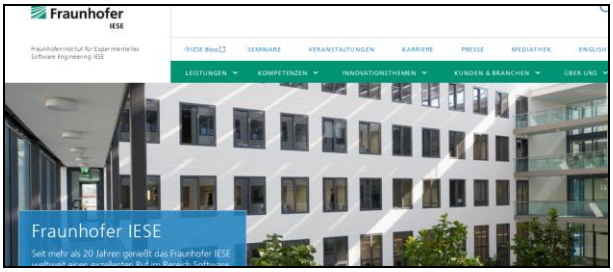
Quantitative Software Management

http://www.qsm.com/



Fraunhofer Institute for Experimental Software Engineering (IESE)

https://www.iese.fraunhofer.de/



National Institute of Standards and Technology (NIST)

https://www.nist.gov/el

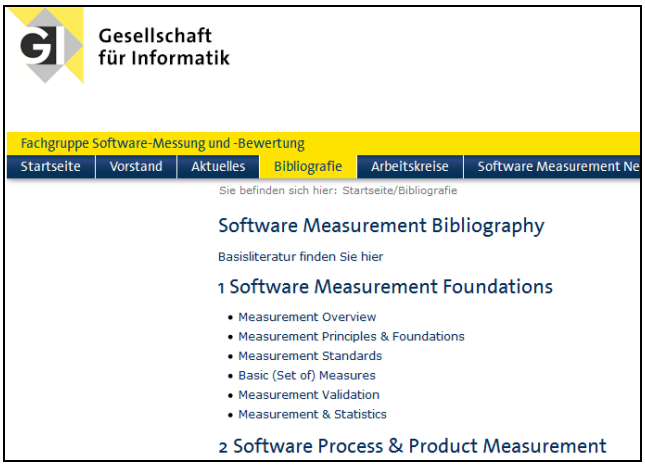
SOFTWARE MEASUREMENT INFORMATION

Software Measurement Bibliography

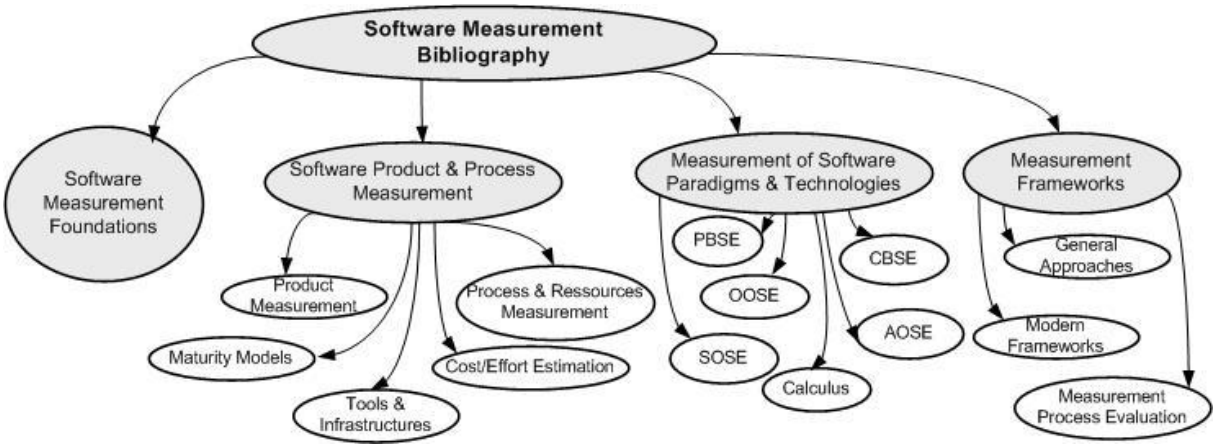
See our overview about software metrics and measurement in the Bibliography at

http://fg-metriken.gi.de/bibliografie.html

including any hundreds of books and papers

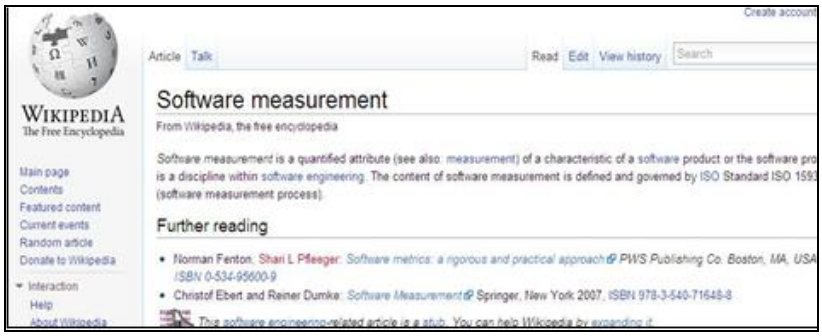


Bibliography Structure:



Software Measurement & Wikipedia

Help to qualify the software measurement knowledge and intentions in the world wide web:



Software Engineering Body of Knowledge (SWEBOK)

http://*www.swebok.orgl*

Project Management Body of Knowledge (PMBOK)

http://www.pmbook.org

SOFTWARE MEASUREMENT NEWS

VOLUME 22 2017 NUMBER 2

CONTENTS

Announcements 3

IWSM/Mensura 2017 Program, Tagung Software-Messung, ESAPI Workshop, ECC 2018

Community Reports 11

Historical Views 15

Citations from Basili et al.: Lessson Learned from 25 years of process improvement . 15
Capers Jones: The Origins of Function Point Metrics 17

Position Papers 21

Andreas Schmietendorf: An empirical evaluation of Open API approaches 21

New Books on Software Measurement 31

Conferences Addressing Measurement Issues 36

Software Measurement in the World-Wide Web 39

ISSN 1867-9196