# The Software Measurement Body of Knowledge

Luigi Buglione, Alain Abran

## Abstract

*The Software Engineering Body of Knowledge (SWEBOK) project of the IEEE Computer Society has developed an international consensus on a Guide to the key knowledge in the Software Engineering domain This SWEBOK Guide is being adopted by the international standardization community as ISO 19759.*

*The SWEBOK Guide includes 10 distinct Knowledge Areas (KAs) and three common themes: Quality, Tools and Measurement. As Measurement is present in all the KAs, some reviewers have suggested representing Measurement as a distinct KA.*

*A recent analysis of software measurement topics comparing SWEBOK to the ISO standard on Metrology and the Abran/Jacquet measurement process model has highlighted a lack of "generally accepted" sources, as well as some missing knowledge types, even in the area of exploitation of measurement results in quality and prediction models. According to the "generally accepted" criteria of the Project Management Institute in the PMBOK, software engineering measurement, as of 2003, would still be considered rather immature in terms of knowledge maturity.*

*At the same time, the speed of research on software measurement has recently been on the increase, and several international standards on software measurement are coming out, both for software processes (CMMI, ISO 15504, 15939) and for software products (ISO 14143, 19761, 9126, etc.). Such results are strengthening the knowledge developed over the last 30 years in terms of measurement processes and methods. We are therefore of the opinion that such recent work is rapidly closing important gaps in software-related measurement knowledge, which could move relatively quickly towards the "generally accepted" threshold for establishing a new KA in the SWEBOK.*

*This paper therefore proposes, on the basis of the Trial version of the SWEBOK Guide, of recent work and of the SWEBOK editorial criteria, an initial taxonomy for a Software Measurement body of knowledge.*

## 1. Introduction

A project with major impact on the recognition of software engineering as a *bona fide* engineering discipline is without doubt the Software Engineering Body of Knowledge (SWEBOK), a research project of the IEEE Computer Society [5], *supported by a consortium of industrial sponsors*. The main goal of the project was to develop a consensus on a guide to the "key knowledge" in the Software Engineering domain. The relevance of the SWEBOK is also enhanced by its upcoming publication as an ISO technical report (ISO/IEC 19759:2004).

The SWEBOK Guide is composed of 10 Knowledge Areas (KAs) and a series of common themes: *Quality*, *Tools* and *Measurement*. In particular, the theme of Measurement is present in almost all the KAs, and some reviewers and researchers have even suggested increasing the visibility of the knowledge on measurement [1,7,11,14] through recognizing it as a distinct, eleventh KA specific to Software Measurement.

The editorial criteria for the inclusion of a KA in the structure of the SWEBOK was documented in the initial phases of the SWEBOK project: a specific type of knowledge had to:

- be recognized *as is* within the ISO structuring standard of ISO/IEC 12207,
- be common in most university software engineering curricula at that time, and
- be common to three of the most widely distributed general books on software engineering.

In addition, it was specified that the content of each KA would have to meet the criteria of generally accepted knowledge, as defined in [6]. It was also, of course, recognized right from the start of the SWEBOK project that the knowledge within this discipline was evolving rapidly and would require subsequent updating, to keep pace with the growing maturity of the discipline.

While the Measurement theme did not meet these criteria at the inception of the SWEBOK project, there has since been both an evolution of the knowledge on software measurement and an increase in awareness of this theme; however, is it mature enough yet for inclusion as an additional KA, as suggested by some reviewers? This paper investigates this issue and presents an evaluation of the state of the art on this topic. It also provides a documented basis for discussion and proposes an initial draft for the structure of a Software Measurement body of knowledge.

## 2. SWEBOK: the Software Engineering Body of Knowledge

### 2.1. Objectives of the SWEBOK Guide

The Guide to the Software Engineering Body of Knowledge (SWEBOK) was initiated by the IEEE Computer Society to address the following five objectives:

1. To promote a consistent view of software engineering worldwide;
2. To clarify the place and set the boundary of software engineering with respect to other disciplines, such as computer science, project management, computer engineering and mathematics;
3. To characterize the content of the software engineering discipline;
4. To provide a topical access to the Software Engineering Body of Knowledge;
5. To provide a foundation for curriculum development and individual certification and licensing material.

The first of these objectives, a consistent worldwide view of software engineering, was supported by a development process which has engaged approximately 500 reviewers from 41 countries[1].

At the same time, Frailey [2] has observed that, while the well-known software process improvement (SPI) models, such as SW-CMM, help organizations in their aim to consistently improve the delivery of software solutions, several SPI initiatives have failed. It has also been observed that the cause of such implementation failure is due to inadequate training, and that, even though people, through these documented SPI initiatives, knew *what* to do, they did not understand

[1] More information regarding the development process can be found in the Preface to the Guide to SWEBOK and on www.swebok.org

*why* they should do it, and could not, therefore, develop the tailoring required for any unusual context.

Frailey, in his use of the SWEBOK Guide, has indicated that it provides relevant support for a common ground on software engineering in both industry and university teaching environments:

- At the industrial level: the SWEBOK Guide meets the need for a reference model for better understanding training needs and for designing career paths [2];
- At the academic level: the SWEBOK Guide meets the need for a reference model for filling gaps in the building of course programs and for ensuring their structural integrity [2][3][4].

### 2.2. SWEBOK phases and versions

The SWEBOK project has been articulated in three phases: Straw Man, Stone Man and Iron Man (Figure 1), and is scheduled for completion by early 2004. The Iron Man review in particular is based on feedback received on the actual usage of the SWEBOK Guide in a variety of organizational contexts, as was hoped in the set-up of the expected audiences for such a Guide.

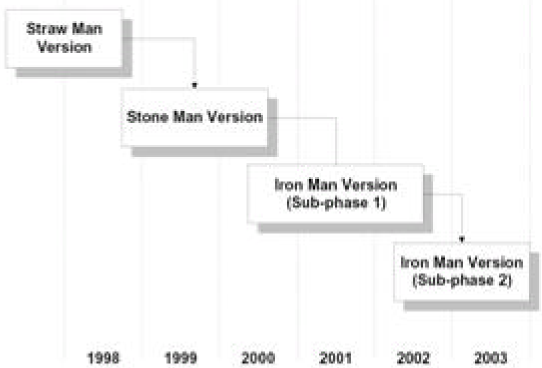


***Figure 1: Phases and timing for the SWEBOK project***

As of December 2003, the final round of comments on Trial version 1.0 of the SWEBOK Guide was being analyzed for inclusion in the next version of the Guide, that is, the Iron Man version.

### 2.3. SWEBOK architecture

The SWEBOK Guide [5] is subdivided into 10 KAs, as listed in Table 1. The descriptions of the KAs are designed to discriminate among the various important concepts, allowing readers to quickly find the information in which they are interested.

***Table 1: SWEBOK Knowledge Areas***

| | |
|---|---|
| KA.01 Software requirements | KA.06 Software configuration management |
| KA.02 Software design | KA.07 Software engineering management |
| KA.03 Software construction | KA.08 Software engineering process |
| KA.04 Software testing | KA.09 Software engineering tools and methods |
| KA.05 Software maintenance | KA.10 Software quality |

The first five (5) KAs can be mapped easily to the primary processes for software development, as defined in the ISO/IEC 12207:1995 standard; the other five KAs can be mapped to some extent to the 12207 taxonomy of supporting processes. By design, each KA has been documented on the basis of a hierarchical organization, as illustrated in Figure 2: a breakdown of topics, a description of each topic, its position within Bloom's taxonomy, and related seminal references.

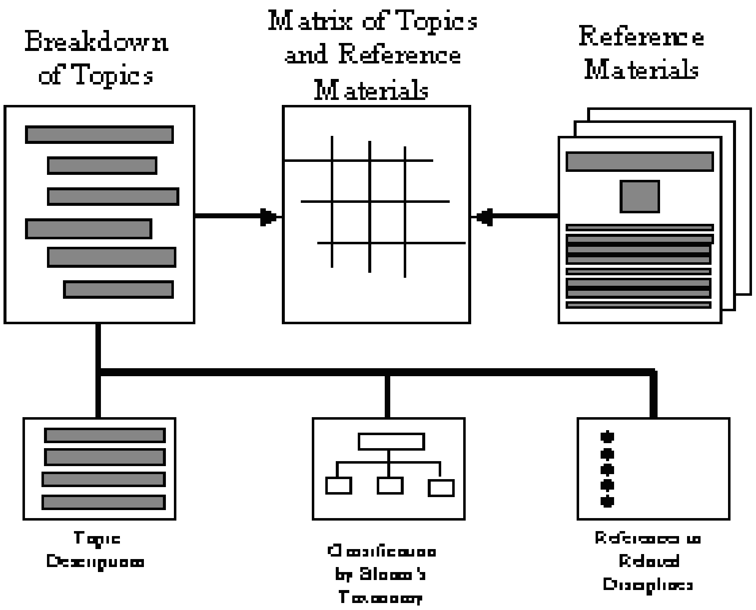


***Figure 2: Organization of a KA description***

## 2.4. Common themes

SWEBOK proposes three common themes which run through the 10 KAs: Quality, Tools and Measurement. Two of these also made it as distinct KAs, that is, Quality and Tools. Why did this not happen to the Measurement theme?

## 2.5. The Generally Accepted principle

For acceptance of knowledge within its Guide, the SWEBOK project has adopted the PMI criteria of what is "generally accepted". The PMBOK explicitly states the following:

"***Generally accepted*** *means that the knowledge and practices described are applicable to most project most of the time, and that there is widespread consensus about their value and usefulness.* ***Generally accepted*** *does not mean that the knowledge and practices described are or should be applied uniformly on all projects; the project management team is always responsible for determining what is appropriate for any given project.*" [6, p.3].

What this means precisely is that the specific concepts of a body of knowledge (BOK), and sources of information about such knowledge, must be sufficiently widespread and broadly applied within the target community for that BOK.

Of these three common themes identified in the 1997 SWEBOK specifications criteria (Quality, Tools and Measurement), only two (Quality and Tools) met the criteria for acceptance and explicit inclusion in the structure of the SWEBOK Guide for both the 1998 Straw Man and 2001 Trial versions.

Of course, since then, the BOK on Software Measurement has kept evolving, but did it do so enough to merit inclusion in the SWEBOK Guide, as suggested by a few reviewers in the 2003 review cycle for the Iron Man version?

In the next section, we investigate how much it has evolved and how much acceptance it is gaining in the general software engineering community.

## 3. Software Measurement in Metrology, in SWEBOK, in SPI models and Standards

The following four areas are analyzed to position the evolution at the end of 2003 on the BOK on Software Measurement with respect to the generally accepted criteria: metrology, the SWEBOK itself, SPI models and international standards.

### 3.1. Metrology

Abran, Sellami and Suryn [7] have recently surveyed and analyzed the SWEBOK KA topics using the measurement concepts presented in the ISO Vocabulary on Metrology [10], the Abran/Jacquet Measurement Process model [8] and the Guide to the Verification of Functional Size Measurement Methods (ISO/IEC 14143-3) [9]. These authors have summarized their findings in the following way: "*Even though measurement-related statements appear throughout the SWEBOK document, they overwhelmingly concern the use of measurement results in assessment and predictive models. By contrast, there is […] very little widely recognized validated knowledge from an engineering perspective, little on the quality of the quantitative inputs to these models, and almost nothing on supporting measuring instruments necessary to obtain these inputs*", that "*the field of software measurement has not yet been fully addressed by current research,*" and that "*much work remains to be done to support software engineering as an engineering discipline based on quantitative data and adequate measurement methods meeting the classic set of criteria for measuring instruments, as described by the metrology body of knowledge in large [sic] use in the engineering disciplines.*"

Buglione [11] has also analyzed how pervasive measurement is in the Trial version of the SWEBOK Guide, through a survey of the measurement-related items in the SWEBOK chapters (see Table 2).

*Table 2: Measurement-related items in the SWEBOK Trial version 1.0*

| | KA | 1st-level topic | 2nd-level topic | 3rd-level topic | What |
|---|---|---|---|---|---|
| 01 | **Software Requirements** | 01. Requirements Engineering process | 04. Process quality & improvement | --- | - Req.Eng. coverage by SPI standards & models<br>- Req.Eng. measures & benchmarking<br>- improvement planning & implementation |
| 02 | **Software design** | 04. SW Design Analysis & Evaluation | 03. Measures | 01. Function-oriented design measures | Quantitative metrics |
| | | | | 02. OO design measures | Quantitative metrics |
| 03 | **Software construction** | --- | --- | --- | **None** |
| 04 | **Software testing** | 04. Test-related Measures | 01. Evaluation of the Program Under Test | --- | Quantitative metrics |
| | | | 02. Evaluation of the Tests performed | --- | Quantitative metrics |
| 05 | **Software maintenance** | 03. Key issues in Software Maintenance | 04. SW Maintenance Measurement | --- | Quantitative metrics |
| 06 | **Software configuration management** | 01. Management of the SCM process | 05. Surveillance of SCM | 01. SCM Measures and Measurement | Quantitative metrics (*no references*) |
| 07 | **Software engineering management** | 03. Software Engineering Measurement | 01. Goals | --- | ISO/IEC 15939; BSC concept |
| | | | 02. Measurement Selection | --- | GQM approach; measurement validity |
| | | | 03. Measuring Software and its development | --- | Size - Structure-Resource - Quality Measurement |
| | | | 04. Collection of data | --- | Survey techniques and form design; automated and manual data collection |
| | | | 05. Software Measurement models | --- | Model building, calibration and evaluation; implementation, interpretation and refinement of models |
| 08 | **Software engineering process** | 03. Process Measurement | 01. Methodology in process measurement | --- | ISO/IEC 15939; reliability & validation (IEEE 1061) |
| | | | 02. Process Measurement paradigms | --- | Analytic (QIP, Process simulation, Orthogonal Defect classification, SPC, PSP) & Benchmarking (SW-CMM, SPICE, ISO 9001, Trillium, Bootstrap, …) paradigms |
| 09 | **Software engineering tools and methods** | 01. Software tools | 09. Software Engineering Management tools | --- | Measurement tools are just mentioned as one of the three categories of tools |
| 10 | **Software quality** | 04. Measurement applied to SQA and V&V | 01. Fundamentals of Measurement | --- | Measurement scales; ISO/IEC 15939 process; PDCA cycle; Measurement plans; Experimental Software Engineering |
| | | | 02. Measures | --- | List of categories for Quality measures |
| | | | 03. Measurement Analysis Technique | --- | List of four categories of techniques |

| | | | | | |
|---|---|---|---|---|---|
| | | | 04. Defect Characterization | --- | IEEE 1044; list of four references |
| | | | 05. Additional uses of SQA and V&V data | --- | IEEE 1012; just two more references ([12] [13]) |

An analysis of Table 2 provides documentary evidence supporting the following observations:

1. "*Measurement is recognized as a key element of engineering and, because of design criteria in the Guide to SWEBOK, it is pervasive in the Guide*" [14]. Nine (9) out of the ten (10) KAs contain references to Software Measurement process and/or measures, with the exception of the Software Construction KA.
2. The instances of measurement-related statements are spread across all KAs, but are not necessarily synchronized, either in terms of the entities to be measured (resources, processes, products) or in terms of the granularity of the measurement concepts discussed within the Trial version of the Guide.
3. Similarly, coverage of the knowledge about Software Measurement is still limited within the range of the full set of typical concepts from the metrology domain; that is, the full spectrum of metrology concepts has not yet been addressed within the software engineering community, either by industry or by researchers.

## 3.2. Measurement within the SPI models and standards communities

At the same time, however, awareness of the Measurement topic has significantly increased over the past five (5) years in both the Software Process Improvements (SPI) and in the international standards communities. This section documents cases in which measurement-related topics have gained significant recognition in software engineering communities over the past five years.

### *3.2.1. Software Process Assessment (SPI)*

When the measurement theme is compared in the two successive versions of the SEI assessment models, it becomes obvious that Measurement, as an engineering-relevant knowledge type, has gained considerable recognition among software engineering professionals:

- **SW-CMM** v1.1 [15] – One of the five Common Features (CF) of all KPAs is the "Measurement and Analysis" feature; it "*describes the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed*" [PAUL93, p.0-28]. For each KPA, a list of process-related measures is provided;
- **CMMI** [16] [17] – The integration of several CMM models has resulted in four Common Features, and Measurement Analysis has become a Process Area at Level 2, referred to as MEA, while the Direct Implementing (DI) common feature suggests process measures under the "*DI3 – Maturity and Control the Process*" label;
- **PSM** [18] – In the United States Department of Defense, the Practical Software Measurement (PSM) Guide provides the foundation for objective project measurement, and is based on ISO 15939.

### *3.2.2. Process-related standards*

*Over the past five years, the international standards community on software engineering has developed, and gained ISO international official recognition for, the following ISO/IEC standards (and technical reports, where relevant)*

- **ISO/IEC 15504:1998 (SPICE)** [19]. Within this ISO document, there is an organizational process (ORG.5) dedicated to Software Measurement, empowering and refining the process model originally proposed in ISO 12207 [20] for the *process* dimension. Similarly, in the *capability* dimension of this ISO document, two process attributes (PAs) deal with measurement (PA4.1 - *measurement* attribute and PA4.2 - *process control* attribute) to demonstrate conformity to Capability Level 4 (CL4 – Predictable process). SPICE does not provide a list of specific measures, but rather process verification criteria.
- **ISO/IEC 15939:2002** [21]. This standard focuses on the Software Measurement process, and represents the extension of the ORG.5 process in SPICE. Further implementation guidance is provided in [18].
- **ISO/IEC 15288:1997** [22]. This is the ISO standard on Systems Life Cycle processes, mapping to 12207, but from the System-wide viewpoint.
- **ISO/IEC 14598 series** [33]. This set of ISO documents provides guidance for successful implementation of a measurement program for software product evaluation.
- **IEEE 1045-1992** [23]. The IEEE standard on Software Productivity Metrics
- **IEEE 1074-1997** [24]. The IEEE standard on the Software Life Cycle processes, such as ISO/IEC 12207.
- **IEEE 1220-1998** [25]. The IEEE standard on the Application and Management of Systems Engineering Process.

*3.2.3. Software product-related standards*

- **ISO/IEC 9126** [26]. This four-part set of ISO documents defines the ISO software quality model, and related candidate measures, for each of the viewpoints selected.
- **ISO/IEC 9241 series**. This is the ISO series of standards on software usability. Part 11 [27] refers to the criteria for rating usability using the SUMI questionnaire.
- **ISO/IEC 14143 series** [28]. This is a meta-standard which documents the requirements for ISO measurement-related standards.

  While the ISO set of standards on product quality measurements [26] has left the software engineering standardization community with some large sets of candidate measures for quality characteristics, the functional size measurement community has defined a much more structured set of specifications for functional size standards. This has led the international community to formally recognize only four specific software functional size measurement methods:
- **ISO/IEC 19761 – COSMIC FFP** [29] – The 2nd generation of Functional Size Measurement (FSM) methods, designed to measure both real-time embedded software size and management information system (MIS) software size, for either single or multi-layered software applications.
- **ISO 20926 – Function Point Analysis (FPA)** [31] – based on Albrecht's 1979 methodology for measuring the size of a software project in a Management Information Systems (MIS) environment;
- **ISO/IEC 20968 – Mark II FPA** [30] – based on the 1988 improvements by C. Symons to the initial FPA method;

- **ISO 24570 – NESMA FPA** [32] – the Dutch version of FPA, with a focus on Software Maintenance and enhancements;

To summarize, the software engineering community has reached an international consensus over the past five years on an impressive number of new measurement-related standards. Such official international recognition of measurement-related documents provides strong evidence of increased "generally accepted" recognition for a number of software measurement topics.

Such recent standardization work on measurement has certainly filled a number of gaps in software-related measurement knowledge. This knowledge, of course, comprises, candidate knowledge with respect to the "generally accepted" criteria for establishing a new KA in the SWEBOK.

## 4. A proposal for a Software Measurement Body of Knowledge

In this section, an initial version of a taxonomy for a BOK on measurement is presented, on the basis of information which already exists in the Trial version of the SWEBOK Guide (see Figure 3).

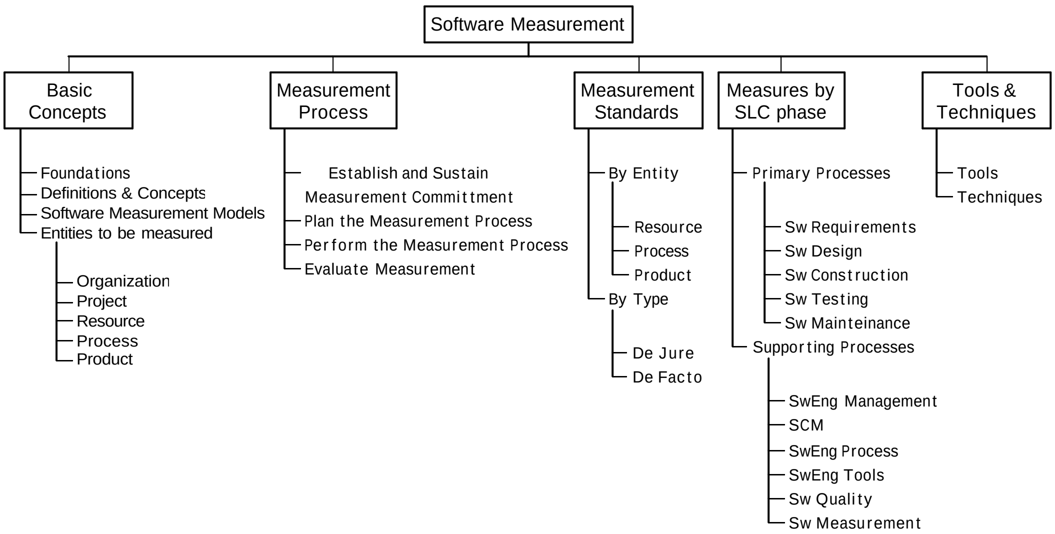


***Figure 3: Proposed taxonomy for a Software Measurement KA***

Of course, further reviews of the taxonomy proposed in Figure 3 will be required to improve it and to progressively develop an international consensus on a measurement BOK.

## 5. Conclusions & Prospects

The SWEBOK Guide has quickly gained widespread international recognition for its contribution to the development of an international consensus on the content of software engineering, thereby providing strong support for its recognition as a *bona fide* engineering domain, albeit not a fully mature one.

While Measurement was not recognized as a distinct topic in the first two official versions of the SWEBOK Guide (Straw Man and Trial versions), our analysis of both its content and its current parallel evolution in the international software engineering communities tends to indicate that the knowledge on software measurement is progressing towards a consensus, to the point that reviewers have recommended that measurement become a distinct KA within the SWEBOK Guide.

An initial candidate taxonomy for a measurement BOK has been proposed here. Of course, such a candidate taxonomy needs to go through further review cycles to ensure its completeness and full internal consistency. While it is already too late to include it in the upcoming Iron Man version of the SWEBOK Guide, it could be tabled for inclusion in the program of work for the next iteration of the evolution of this Guide in the next three-to-five-year plan.

## 6. References


[1] Schneidewind, N.F., "Body of Knowledge for Software Quality Measurement," IEEE Computer, Vol. 35, No. 2, February 2002, pp.77-83

[2] Frailey, D.J. and Mason, J., "Using SWEBOK for Education Programs in Industry and Academia," CSEET'02 Conference, Feb. 25-27, 2002, Covington (USA), URL: http://www.site.uottawa.ca/cseet2002/slides/cseet2002FraileyMason.ppt

[3] Ludi, S. and Collofello, J., "An Analysis of the Gap between the Knowledge and Skills Learned in Academic Software Engineering Course Projects and those required in Real Projects," 31st ASEE/IEEE Frontiers in Education Conference, Oct 10-13 2001, Reno (NV, USA), Session T2D, URL: http://fie.engrng.pitt.edu/fie2001/papers/1187.pdf

[4] Matsumoto, Y., "A Web-based Usage of Software Engineering Body of Knowledge", URL: http://www.sft.cs.musashi-tech.ac.jp/mst/mst.pdf

[5] Abran, A., Moore, J.W., Bourque, P., Dupuis, R. and Tripp, L.T., "Guide to the Software Engineering Body of Knowledge (SWEBOK), Trial Version 1.0," May 2001, IEEE/ACM, URL: http://www.swebok.org

[6] Project Management Institute, "A Guide to the Project Management Body of Knowledge," 1996, URL: www.pmi.org

[7] Abran, A., Sellami, A. and Suryn, W., "Metrology, Measurement and Metrics in Software Engineering," Proceedings of the Metrics 2003 Conference, September, 3-5 2003, Sydney (Australia)

[8] Jacquet J.P. and Abran A., "From Software Metrics to Software Measurement: A Process Model," The 3th IEEE International Software Engineering Standard Symposium, ISESS'97, Walnut Creek, CA, June 2-6, 1997, IEEE Computer Society Press

[9] ISO ISO/IEC 14143-3: 2003, "Software Engineering-Functional Size Measurement- Part 3: Verification of Functional Size Measurement Methods," International Organization for Standardization - ISO, Geneva, 2003

[10] ISO, "International Vocabulary of Basic and General Terms in Metrology," International Organization for Standardization - ISO, Geneva, 1993

[11] Buglione, L., "SWEBOK e Software Measurement: alcune riflessioni," presentazione, GUFPI-ISMA meeting, Rome, 17/07/2003

[12] Rakitin, S.R., "Software Verification and Validation, A Practitioner's Guide," Artech House, Inc., 1997

[13] McConnell, S.C., "Code Complete: a practical handbook of software construction", Microsoft Press, 1993

[14] Abran, A., Bourque, P. and Dupuis, R., "The SWEBOK Initiative and Software Measurement Intentions," Proceedings of the 12th International Workshop on Software Measurement (IWSM2002), October 7-9, 2002, Magdeburg (Germany), Shaker Verlag, ISBN 3-8322-0765-1, pp. 168-178, URL: http://www.lrgl.uqam.ca/publications/pdf/779.pdf

[15] Paulk, M.C., Weber, C.V., Garcia, S.M., Chrissis, M.B. and Bush, M., "Key Practices of the Capability Maturity Model Version 1.1," Software Engineering Institute/Carnegie Mellon

University, CMU/SEI-93-TR-25, February 1993, URL: http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf

[16] CMMI PRODUCT DEVELOPMENT TEAM, "CMMI for Systems Engineering / Software Engineering / Integrated Product and Process Development," Version 1.1, CMMI-SE/SW/IPPD v1.1, Staged Representation, CMU/SEI-2002-TR-004, Technical Report, Software Engineering Institute, December 2001

[17] CMMI PRODUCT DEVELOPMENT TEAM, "CMMI for Systems Engineering / Software Engineering / Integrated Product and Process Development," Version 1.1, CMMI-SE/SW/IPPD v1.1, Continuous Representation, CMU/SEI-2002-TR-003, Technical Report, Software Engineering Institute, December 2001

[18] Dept. of Defense and US Army, "PSM: Practical Software and Systems Measurement – A Foundation for Objective Project Management," Version 4.0c, March 2003

[19] ISO, ISO/IEC JTC1/SC7/WG10, TR 15504-5, "Software Process Assessment - Part 5: An Assessment Model and indicator guidance," v.3.03, 1998

[20] ISO, ISO/IEC JTC1/SC7/WG7, "International Standard 12207 - Information Technology: Software Life Cycle Processes," February 22, 1995

[21] ISO, ISO/IEC, "IS 15939: Software Engineering – Software Measurement Process Framework," International Organization for Standardization, Geneva, 2003

[22] ISO, ISO/IEC, "IS 15288:2002: Systems Engineering – System Life Cycle Processes," International Organization for Standardization, 2002, Geneva

[23] IEEE, "Std 1045-1992: Standard for Software Productivity Metrics," 1992

[24] IEEE, "Std 1074-1997: Standard for Developing Software Life Cycle Processes," 1997

[25] IEEE, "Std. 1220-1998: IEEE Standard for Application and Management of the Systems Engineering Process," 1998

[26] ISO, ISO/IEC, "9126: Software Engineering - Product Quality Parts 1-4," International Organization for Standardization, 2001-2003

[27] ISO, ISO/IEC, "9241-11: 1988 Guidance of Usability," International Organization for Standardization, 1998

[28] ISO, ISO/IEC, "14143: Software Engineering – Functional Size Measurement Parts 1-6," International Organization for Standardization, 1998-2003

[29] ISO ISO/IEC, "IS 19761:2003: Software Engineering – COSMIC FFP – A functional size measurement method," International Organization for Standardization, 2003, Geneva

[30] ISO, ISO/IEC, "IS 20968:2002: Software Engineering – MkII Function Point Analysis – Counting Practice Manual," International Organization for Standardization, 2002, Geneva

[31] ISO, ISO/IEC, "IS 20926:2003: Software Engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practice Manual," International Organization for Standardization, 2003, Geneva

[32] ISO, ISO/IEC, "DIS 24570 – Software Engineering – Definitions and counting guidelines for the application of Function Point Analysis," N2865, 02/06/2003

[33] ISO, ISO/IEC, "14598 - Information technology -- Software product evaluation, Parts 1-6" International Organization for Standardization, 1999-2001

**Deleted:** TR15288: …………………………………………