Guide to the Software Engineering Body of Knowledge A Project Overview

Pierre Bourque[1], Robert Dupuis, Alain Abran,
James W. Moore[2], Leonard Tripp[3]

1. Introduction

Since 1993, the IEEE[4] Computer Society[5] and the ACM[6] have been actively promoting software engineering as a profession, notably through their involvement in the Joint IEEE Computer Society-ACM Software Engineering Coordinating Committee[7]. A profession has, in particular, to meet the following five criteria:

1. Creation of an organized body of knowledge;
2. Client recognition of the authority of the profession;
3. Community approval of the profession's authority;
4. A code of ethics;
5. A professional culture supported by, and advanced by, academic and professional society activities.

The purpose of this overview is to present an ongoing project which seeks to identify and describe an organized body of knowledge for software engineering. It is a three-phase effort: the Straw Man phase was completed in September of 1998, the Stone Man version is scheduled to be delivered by the end of 1999 and the Iron Man version is due for completion in 2001.

To set the overview in it's context, the paper begins by presenting some purpose and background material. The project's objectives, deliverables, intended audience and the two underlying principles of the chosen development approach are then discussed. This is followed by a summary of the Straw Man phase's results. The Stone Man development approach and a summary of the interim results of this ongoing phase are then presented followed by brief remarks on the Iron Man phase.

[1] Pierre Bourque, Robert Dupuis and Alain Abran are with the Software Engineering Management Research Laboratory of the Université du Québec à Montréal; bourque.pierre@uqam.ca, dupuis.robert@uqam.ca , abran.alain@uqam.ca. Pierre Bourque and Robert Dupuis are Co-editors of the Guide to the Software Engineering Body of Knowledge project. Alain Abran is Co-executive editor of the project.

[2] James W. Moore is with The Mitre Corporation; James.W.Moore@ieee.org. James W. Moore is Co-executive editor of the project.

[3] Leonard Tripp is 1999 President of the IEEE Computer Society and Chair of the Joint IEEE Computer Society-ACM Software Engineering Coordinating Committee. He is also known as the "project's champion". l.tripp@computer.org

[4] Institute of Electrical and Electronics Engineers. See www.ieee.org

[5] See www.computer.org

[6] Association for Computing Machinery. See www.acm.org

[7] See http://www.computer.org/tab/swecc/

2. Purpose and Background

Consensus on a Core Body Knowledge Is Crucial

In spite of the millions of software professionals worldwide and the ubiquitous presence of software in our society, software engineering has not reached the status of a legitimate engineering discipline and a recognized profession. In other engineering disciplines, the accreditation of university curricula and the licensing and certification of practicing professionals are taken very seriously. These activities are seen as critical to the constant upgrading of professionals and, hence, the improvement of the level of professional practice. Recognizing a core body of knowledge is pivotal to the development and accreditation of university curricula and the licensing and certification of professionals.

Achieving consensus by the profession on a core body of knowledge is a key milestone in all disciplines and has been identified by the Joint IEEE Computer Society-ACM Software Engineering Coordinating Committee as crucial for the evolution of software engineering toward a professional status. The Guide to the Software Engineering Body of Knowledge (SWEBOK) project[8] is an initiative completed under the auspices of this Committee to reach this consensus.

Focus on Generally Accepted Knowledge

The software engineering body of knowledge is an all-inclusive term that describes the sum of knowledge within the profession of software engineering. Since it is usually not possible to put the full body of knowledge of even an emerging discipline, such as software engineering, into a single document, there is a need for a Guide to the Software Engineering Body of Knowledge. This Guide will seek to identify and describe that subset of the body of knowledge that is generally accepted, even though software engineers must be knowledgeable not only in software engineering, but also, of course, in other related disciplines.

What do we mean by “generally accepted knowledge”?

The Guide to the Software Engineering Body of Knowledge project seeks to identify and describe that subset of the body of knowledge that is generally accepted or, in other words, the core body of knowledge. To better illustrate what “generally accepted knowledge” is relative to other types of knowledge, Figure 1 proposes a draft three-category schema for classifying knowledge.

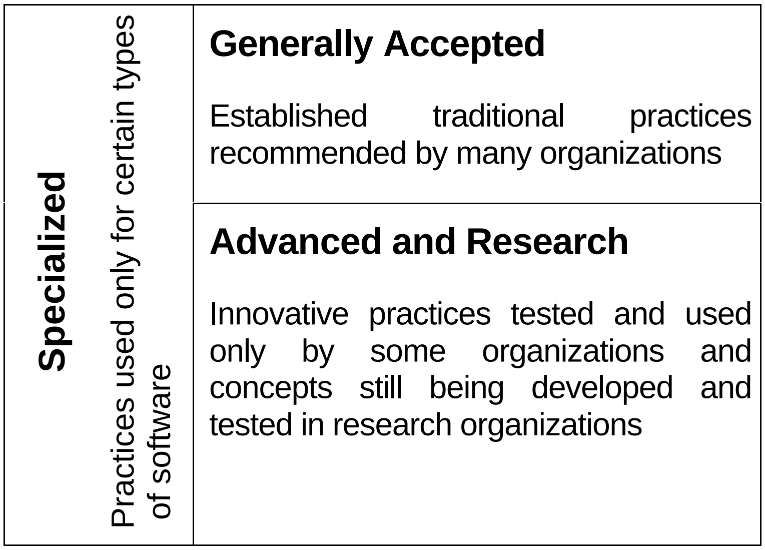
The Project Management Institute in its Guide to the Project Management Body of Knowledge [1][9] defines “generally accepted” knowledge for project management in the following manner:

[8] All final and intermediate project deliverables are available for free at www.swebok.org.

[9] The Guide to the Project Management Body of Knowledge is now also an IEEE Standard.

> *‘“Generally accepted” means that the knowledge and practices described are applicable to most projects 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.’*

The project has also defined “generally accepted” as being knowledge to be included in the study material of a software engineering licensing exam that a graduate would pass after completing four years of work experience. These two definitions should be seen as complementary.



Specialized

Practices used only for certain types of software

Generally Accepted

Established traditional practices recommended by many organizations

Advanced and Research

Innovative practices tested and used only by some organizations and concepts still being developed and tested in research organizations

• Figure 1 Categories of knowledge

Software engineering body of knowledge and curriculum are not the same

Software engineers must not only be knowledgeable in what is specific to their discipline, but they also, of course, have to know a lot more. The goal of this project is not, however, to inventory everything that software engineers should know, but to identify what forms the core of software engineering. It is the responsibility of other organizations and initiatives involved in the licensing and certification of professionals and the development of accreditation criteria and curricula to define what a software engineer must know outside software engineering. We believe that a very clear distinction must be made between the software engineering body of knowledge and the contents of software engineering curricula. Notably through the auspices of the Software Engineering Coordinating Committee, the project is collaborating and maintaining close contact with a number of these other organizations and initiatives.

3. Project Objectives

The objectives of the SWEBOK project are therefore to:

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

4. Deliverables and Intended Audiences

The deliverables of the currently ongoing Stone Man phase of this project will be:

- an approved list of Knowledge Areas of software engineering;
- an approved list of topics and relevant reference material for each Knowledge Area;
- an approved list of disciplines related to Software Engineering, and the Knowledge Areas and topics lying at the junction of Software Engineering and one or more of these Related Disciplines.

To facilitate its broad dissemination, the guide will be available at no cost on the Web.

The intended audiences of this guide are:

- public and private corporations wishing to use and promote a consistent view of software engineering within their organization, notably when defining education and training, job classification, and performance evaluation policies;
- practicing software engineers;
- makers of public policy defining software engineering licensing policies and guidelines for professionals;
- professional societies defining university software engineering program accreditation guidelines and certification policies and guidelines for professionals;
- software engineering students learning the profession;
- educators and trainers defining curricula and course content.

5. Underlying Principles

The two following principles are key to the project and underlie it's entire development approach:

- **transparency**: the development process is itself published and fully documented;
- **consensus-building**: the development process is designed to build, over time, consensus in industry, professional societies and standards-setting bodies, among practicing software developers and in academia.

6. Summary of Straw Man Phase Results

The objectives of the Straw Man phase were to define the project's strategy, to deliver what is referred to as the Straw Man version of the Guide to the Software Engineering Body of Knowledge and to gather momentum in the profession for the project. The Straw Man version was published in September of 1998 [2].

The main goal of this initial report is to propose a draft list of Knowledge Areas for the Guide to the Software Engineering Body of Knowledge. A candidate or "jumpstart" list of 17 Knowledge Areas are therefore identified in this report. This report also proposes a draft list of the ten disciplines that interact with software engineering. As its name implies, this Straw Man version is intended to be challenged and to stimulate a vigorous debate.

Knowledge Areas are the major components of a discipline, or sub-fields of study. Related Disciplines are the other disciplines with which software engineering has a non-empty intersection or shares a common boundary.

In order to propose Knowledge Areas and Related Disciplines for "generally accepted" knowledge and to do so based on recognized, public and verifiable sources of information, it was decided that the tables of contents of general software engineering textbooks, the curricula of undergraduate and graduate programs in software engineering and the admission criteria for graduate programs would constitute the input to the analysis. A total of 24 textbooks and 29 programs were examined.

For the purposes of this Straw Man version, a potential knowledge area had to be mentioned in the table of contents of at least one quarter of the textbooks sampled to qualify as a proposed Knowledge Area. The ISO/IEC 12207 standard [3] on Software Life Cycle Processes was used as the basis and vocabulary for the classification of the different topics related to the life cycle. A number of other topics not related to the lifecycle were also considered.

7. Stone Man Development Approach and Interim Results

Based on the results of the Straw Man phase, a second or Stone Man phase was instigated in the Fall of 1998 and is planned to end in late 1999. To ensure relevance of the Guide, to build consensus and momentum for the Guide and to encourage its quick uptake in the marketplace, five components are key to the proposed strategy of the current Stone Man phase: an Industrial Advisory Board, a select panel of experts, a number of *Knowledge Area specialists*, a number of electronic peer review groups, and a broad comment-gathering and consensus-building process over the Internet among the IEEE Computer and ACM memberships and the software engineering community.

The Industrial Advisory Board is composed of representatives from industry (Boeing[10], Comerica[11], National Institute of Standards and Technology[12], National Research Council of Canada[13], Raytheon Systems[14] and SAP Labs (Canada)[15]), major professional societies (IEEE Computer Society[16], ACM[17]), international standards-setting bodies (IEEE Software Engineering Standards Committee[18], ISO/IEC JTC1/SC7[19]) and the Computing Curricula 2001 initiative[20]. The panel of experts is currently composed of renowned personalities in the field: Steve McConnell[21], Roger Pressman[22] and Ian Sommerville[23].

The Board's responsibilities include providing input to ensure relevance to various audiences, reviewing and approving strategy and deliverables, overseeing the broad comment-gathering and consensus-building and electronic peer review process, assisting in promoting the Guide, and lending credibility to the initiative.

For each Knowledge Area, a Knowledge Area specialist is responsible for identifying topics and selecting key references from the existing software engineering literature based on predefined reference selection criteria (these references could be book chapters, journal articles, industrial reports, etc.). A number of electronic review groups each representing a distinct stakeholder will be asked to review the Knowledge Area descriptions. Subsequent to this, the membership of major professional societies and the software engineering community will be solicited to review and comment on the project deliverables.

[10] See www.boeing.com

[11] See www.comerica.com

[12] See www.nist.gov

[13] See www.nrc.ca

[14] See ww.raytheon.com/rsc/

[15] See www.sap.com

[16] See www.computer.org

[17] See www.acm.org

[18] See www.computer.org/standard/sesc/

[19] See www.iso.ch/meme/JTC1SC7.html

[20] See www.computer.org/educate/eab.htm

[21] See www.construx.com/stevemcc/

[22] See www.rspa.com/

[23] See www.comp.lancs.ac.uk/computing/staff/is.html

The Industrial Advisory Board is responsible for overseeing that due process has been followed.

The development of the Stone Man version of the Guide to the Software Engineering Body of Knowledge is planned to proceed through three review cycles during 1999:

- Review Cycle 1: Focus will be on the choice of topics and the definitions of Knowledge Areas by a limited set of subject area experts. Target review period: April 1999

- Review Cycle 2: Focus will be on completeness of coverage and depth of coverage across Knowledge Areas. The review will be organized by selected viewpoints: educators, trainers, regulators, small business, large business, licensing authorities, researchers, individual practitioners, etc. Target review period: May and June and July 1999.

- Review Cycle 3: Focus will be on a broad-based review by individuals and organizations representing a suitable cross-section of potential interest groups. Target review period: October 1999.

The following intermediate deliverables have been produced so far during this phase and have been approved officially by the project's Industrial Advisory Board:

- A detailed development plan for the Stone Man phase clearly stating the project's objectives, its' deliverables and intended audiences, the roles and responsibilities of the contributors and the phase's development approach and schedule [4].

- Based on the Straw Man report and on criteria and requirements developed during the Stone Man phase, a baseline list of Knowledge Areas, i.e. an agreed upon starting point, for the Stone Man phase [5]. Table 1 lists these Knowledge Areas and identifies as well their corresponding Knowledge Area Specialists.

- Based on the Straw Man report and on criteria and requirements developed during the Stone Man phase, a baseline list of Related Disciplines, i.e. an agreed upon starting point, for the Stone Man phase [6]. This document also identifies a candidate list of Knowledge Areas for each of these Related Disciplines. Table 2 contains the baseline list of Related Disciplines.

- A set of detailed requirements prepared by the Editorial Team for the Knowledge Area Specialists specifying the contents and the evaluation criteria of the Knowledge Area Description documents [7].

Knowledge Area	Knowledge Area Specialist	Affiliation
Software Requirements Analysis	Pete Sawyer and Gerald Kotonya	Lancaster University, UK
Software Design	Guy Tremblay	Université du Québec à Montréal, Canada
Software Construction	Terry Bollinger	The MITRE Corporation, USA
Software Testing	Antonia Bertolino	Istituto di Elaborazione della Informazione, CNR, Italy
Software Evolution & Maintenance	Chris Verhoef	University of Amsterdam, The Netherlands
Software Quality Analysis	Dolores Wallace and Larry Reekeer	National Institute of Standards and Technology, USA
Software Configuration Management	John Scott	The Lawrence Livermore National Laboratory, USA
Software Engineering Infrastructure	Dave Carrington	University of Queensland, Australia
Software Engineering Management	Stephen MacDonell	University of Otago, New Zealand
Software Engineering Process	Khaled El Emam	National Research Council, Canada

Table 1 Baseline List of Knowledge Areas and their Specialists

Related Discipline
Computer Science
Mathematics
Project Management
Computer Engineering
Systems Engineering
Management and Management Science
Cognitive Sciences and Human Factors

Table 1 Baseline List of Related Disciplines

The following working documents have been produced so far during this phase:

- A "jumsptart list" of topics for each Knowledge Area. These documents have been provided to each Knowledge Area Specialist as an enabler and contain a breakdown of topics based on the analysis of the contents of the four most widely sold general software engineering textbooks[24] [8] [9], [10], [11].

- Version 0.1 of the Knowledge Area Descriptions containing an initial breakdown of topics for each Knowledge Area.

[24] Based on the sales ranking available at www.amazon.com

- Review comments by a select group of three or four domain experts for each Knowledge Area. These initial breakdowns of topics were reviewed for reasonableness and major omissions.

Knowledge Area Specialists are currently working on Version 0.5 of the Knowledge Area Descriptions. The Editorial Team is currently aggressively recruiting reviewers from a wide spectrum of stakeholders from around the world for the second and third review cycles and encourage you to take an active role in this project by participating in these review cycles[25]. At least two subsequent draft versions of the Knowledge Area Descriptions will be published later this year based on the incoming comments and reviews. Detailed comment disposition reports describing how comments were handled will also be produced and made available.

8. Iron Man Phase

A subsequent Iron Man version should be completed roughly two years after the Stone Man version. The development of this version will once again probably involve an Industrial Advisory Board and various expert panels. However, an even more exhaustive review and consensus-building process to gather comments and insights from members of the profession will have to be defined for this phase of the project.

It is also expected that a much more dynamic and incremental development approach will be adopted for this phase.

9. Concluding remarks

With the ubiquitous presence of software in our society and with the increased concerns over the necessity for certification and licensing, industry-wide consensus on a Guide to Software Engineering Body of Knowledge is now a must. Leadership on this important issue must exist on a worldwide scale, otherwise future university program accreditation guidelines and certification and licensing rules for professionals will differ widely, thereby reducing the mobility of professionals. Involvement by all parties, industry, professional societies and professionals, standards-setting bodies and academia, is critical to ensure the relevancy and the credibility of results, and for a quick uptake of the results.

Acknowledgments

This project is made possible through funding provided by the IEEE Computer Society, the Association for Computer Machinery, Boeing, the National Institute of Standards and Technology, the National Research Council of Canada and SAP Labs (Canada). The project is managed by the Université du Québec à Montréal.

[25] See electronic Reviewer/Review Captain Submission Form available at www.swebok.org

References

[1] W. R. Duncan, "A Guide to the Project Management Body of Knowledge," Project Management Institute, Upper Darby, PA 1996.

[2] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, L. Tripp, K. Shyne, B. Pflug, M. Maya, and G. Tremblay, "Guide to the Software Engineering Body of Knowledge - A Straw Man Version," Université du Québec à Montréal, Montréal, Technical September 1998.

[3] IEEE/EIA, "Information Technology - Software Life Cycle Processes," International Standard, Technical ISO/IEC 12207:1995(E), 1995.

[4] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, L. L. Tripp, and D. Frailey, "Approved Plan for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge," Université du Québec à Montréal, Montréal January 19 1999.

[5] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, L. L. Tripp, and D. Frailey, "Approved Baseline for a List of Knowledge Areas for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge," Université du Québec à Montréal, Montréal January 19 1999.

[6] R. Dupuis, P. Bourque, A. Abran, J. W. Moore, and L. L. Tripp, "Approved Baseline for a List Related Disciplines for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge," Université du Québec à Montréal, Montréal March 26 1999.

[7] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, and L. L. Tripp, "Approved Knowledge Area Description Specifications for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge," Université du Québec à Montréal, Montréal Version 0.2, March 26 1999.

[8] M. Dorfman and R. H. Thayer, *Software Engineering*. Los Alamitos, California: IEEE Computer Society Press, 1997.

[9] S. L. Pfleeger, *Software Engineering - Theory and Practices*. Upper Saddler Rive, New Jersey: Prentice Hall, 1998.

[10] R. S. Pressman, *Software Engineering, A Practitionner's Approach*, Fourth ed. New York: The McGraw Hill Companies, 1997.

[11] I. Sommerville, *Software Engineering*, Fifth Edition ed. Harlow, England: Addison Wesley, 1995.