Bloom's Taxonomy Levels for Three Software Engineer Profiles

P. Bourque
École de technologie supérieure
Montreal, Canada
pbourque@ele.etsmtl.ca

L. Buglione
École de technologie supérieure
Montreal, Canada
luigi.buglione@computer.org

A. Abran
École de technologie supérieure
Montreal, Canada
aabran@ele.etsmtl.ca

A. April
École de technologie supérieure
Montreal, Canada
aapril@ele.etsmtl.ca

Abstract

This paper is the product of a workshop held in Amsterdam during the Software Technology and Practice Conference (STEP 2003). The purpose of the paper is to propose Bloom's taxonomy levels for the Guide to the Software Engineering Body of Knowledge (SWEBOK) topics for three software engineer profiles: a new graduate, a graduate with four years of experience, and an experienced member of a software engineering process group. Bloom's taxonomy levels are proposed for topics of four Knowledge Areas of the SWEBOK Guide: software maintenance, software engineering management, software engineering process, and software quality. By proposing Bloom's taxonomy in this way, the paper aims to illustrate how such profiles could be used as a tool in defining job descriptions, software engineering role descriptions within a software engineering process definition, professional development paths, and training programs.

Index Terms—*Guide to the Software Engineering Body of Knowledge, SWEBOK, Bloom's Taxonomy.*

1. Introduction.

This paper is the product of a workshop entitled "Expected Levels of Understanding of SWEBOK Topics," which took place during the Software Technology and Practice (STEP 2003) conference held in Amsterdam. The Guide to the Software Engineering Body of Knowledge (SWEBOK) [1] includes an evaluation of SWEBOK topics according to Bloom's taxonomy [2] to help audiences wishing to use the SWEBOK Guide as a tool in designing course material, university programs or accreditation criteria, job descriptions, role descriptions within a software engineering process definition, professional development paths, and professional training programs, however it does so only for the profile of a graduate with four years of experience. In order to illustrate how Bloom's taxonomy levels of SWEBOK Guide topics can be adapted for, and useful in, varying contexts, this paper proposes Bloom's taxonomy levels for two additional software engineer profiles: a new graduate and an experienced member of a software engineering process group.

The use of Bloom's taxonomy levels in conjunction with the SWEBOK Guide breakdown of topics has been shown to be useful in other studies. They were instrumental, for example, when Surendran et al. defined a framework for software engineering apprenticeships [3],, and Ramakrishnan and Cambrell combined them to develop a Web-based tool permitting students to interact with a content map of the undergraduate software engineering curriculum at Monash University [4]. Based on their progress in the curriculum, students are capable of viewing what they have learnt to date in terms of Bloom's taxonomy levels of SWEBOK Guide topics. Benediktsson [5] analyzed the coverage of undergraduate and graduate curricula in software engineering at the University of Iceland using the SWEBOK Guide breakdown and Bloom's taxonomy levels as proposed in the Trial Version of the SWEBOK Guide [6]. Ludi and Collofello used Bloom's taxonomy levels of SWEBOK Guide topics to identify improvements in an undergraduate software engineering course [7]. Bloom's taxonomy levels are also proposed for topics of the Software Engineering Education body of Knowledge (SEEK) included in the Software Engineering volume of the Computing Curriculum [8], the purpose of which is to provide guidance on the contents of an undergraduate software engineering curriculum. Though distinct, notably in terms of scope, the two bodies of knowledge are closely related [9]. Construx Software proposes an elaborate professional development ladder for software engineers in their organization based on SWEBOK Guide Knowledge Areas and a four-level "capability" schema:

introductory, competence, leadership, and mastery [10].

This paper presents an overview of the SWEBOK Guide in section 2. Section 3 briefly presents Bloom's taxonomy levels. Section 4 proposes taxonomy levels for three software engineer profiles and discusses some of the difficulties encountered. A summary concludes the paper.

2. The Guide to the Software Engineering Body of Knowledge (SWEBOK).

The objectives of the SWEBOK Guide are to characterize the content of the software engineering discipline, to promote a consistent view of software engineering worldwide, to clarify the place, and set the boundary, of software engineering with respect to other disciplines, and to provide a foundation for curriculum development and individual licensing material. All deliverables are available at no charge at www.swebok.org.

The SWEBOK Guide seeks to identify and describe the subset of software engineering knowledge that is generally accepted. Generally accepted knowledge applies to most projects most of the time, and widespread consensus validates its value and effectiveness [11]. A complementary definition states that generally accepted knowledge should be included in the study material for a software engineering licensing examination that graduates would take after gaining four years of work experience. Although this criterion is specific to the U.S. style of education and does not necessarily apply to other countries, it was deemed useful. Research topics and specialized topics, meaning topics that apply only to certain kinds of software, are therefore outside its scope.

However, the term "generally accepted" should not be taken to mean that this knowledge is uniformly applicable to all software engineering endeavors—each project's needs determine that—but it does imply that competent, capable software engineers should be equipped with this knowledge for potential application.

The SWEBOK Guide is oriented toward a variety of audiences, all over the world. It is aimed at serving public and private organizations in need of a consistent view of software engineering for defining education and training requirements [12], classifying jobs [13], and developing performance evaluation policies and career paths [10]. It also addresses the needs of practicing software engineers and software engineering managers, as well as the officials responsible for making public policy [14], in addition to addressing the definition of licensing and professional guidelines [15]. Moreover, professional societies defining their certification rules[1] and educators drawing up accreditation policies for university curricula[2] will benefit from consulting the SWEBOK Guide, as will students of software engineering and educators and trainers engaged in defining curricula [3], [4], [5], [8], [12] and course content [7].

The SWEBOK Guide is subdivided into ten Knowledge Areas, the descriptions of which are designed to discriminate among the various important concepts, permitting readers to find their way quickly to subjects of interest. Upon finding such a subject, readers are referred to key papers or book chapters selected because they present the knowledge succinctly. The ten Knowledge Areas are listed in Table 1. Each of them is treated as a chapter in the SWEBOK Guide.

Table 1. SWEBOK Guide Knowledge Areas.

Software requirements
Software design
Software construction
Software testing
Software maintenance
Software configuration management
Software engineering management
Software engineering process
Software engineering tools and methods
Software quality

The SWEBOK Guide uses a hierarchical organization to decompose each Knowledge Area into a set of topics with recognizable labels. A two- or three-level breakdown provides a reasonable way to find topics of interest.

In establishing a boundary, it is also important to identify what disciplines share such a boundary, and often a common intersection, with software engineering. To this end, the SWEBOK Guide also recognizes eight related disciplines, as listed in Table 2. Software engineers should, of course, know material from these fields. It is not, however, an objective of the SWEBOK Guide to characterize the knowledge of the related disciplines.

[1] See http://www.computer.org/certification/ .
[2] See http://www.ipsj.or.jp (in Japanese)

Table 2. SWEBOK Guide Related Disciplines.

Computer engineering
Computer science
Management
Mathematics
Project management
Quality management
Software ergonomics
Systems engineering

The SWEBOK Guide is a three-phase project begun in 1998. A first prototype version, known as the Straw Man version, was published in 1998. A second complete edition, known as the Trial Version, was published in 2001 [6]. The Trial Version was developed through a managed consensus process involving 8,000 comments collected through three review cycles involving, in total, close to 500 reviewers from over 40 countries. Based notably on feedback received from users of the Trial Version and on an additional review cycle, the 2004 Version of the SWEBOK Guide is now available [1].

The SWEBOK Guide is a project of the IEEE Computer Society, with support from the following organizations: Boeing, the Canadian Council of Professional Engineers, Construx Software, the MITRE Corporation, the National Institute of Standards & Technology, the National Research Council of Canada, Rational Software, Raytheon, and SAP Labs Canada.

The 2004 Version of the SWEBOK Guide was approved by the Board of Governors of the IEEE Computer Society. It will also be published by the International Organization for Standardization (ISO) as Technical Report 19759.

3. Bloom's taxonomy.

Bloom's Taxonomy of the Cognitive Domain proposed in 1956 contains six levels. Table 3[3] presents these levels and keywords often associated with each level.

Table 3. Bloom's Taxonomy.

Bloom's Taxonomy Level	Associated Keywords
Knowledge: Recall of data.	Defines, describes, identifies, knows, labels, lists, matches, names, outlines, recalls, recognizes, reproduces, selects, states.
Comprehension: Understand the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.	Comprehends, converts, defends, distinguishes, estimates, explains, extends, generalizes, gives examples, infers, interprets, paraphrases, predicts, rewrites, summarizes, translates.
Application: Use a concept in a new situation, Applies what was learned in the classroom to novel situations in the workplace.	Applies, changes, computes, constructs, demonstrates, discovers, manipulates, modifies, operates, predicts, prepares, produces, relates, shows, solves, uses.
Analysis: Separate material or concepts into component parts so that its organizational structure can be understood. Distinguishes between facts and inferences.	Analyzes, breaks down, compares, contrasts, deconstructs, differentiates, discriminates, distinguishes, identifies, illustrates, infers, outlines, relates, selects, separates.
Synthesis: Build a structure or pattern from diverse elements. Put parts together to form a whole, with the emphasis on creating a new meaning or structure.	Categorizes, combines, compiles, composes, creates, devises, designs, explains, generates, modifies, organizes, plans, rearranges, reconstructs, relates, reorganizes, revises, rewrites, summarizes, tells, writes.
Evaluation: Make judgments about the value of ideas or materials.	Appraises compares, concludes, contrasts, criticizes, critiques, defends, describes, discriminates, evaluates, explains, interprets, justifies, relates, summarizes, supports.

4. Bloom's taxonomy ratings for three software engineer profiles.

The 2004 Version of the SWEBOK Guide proposes, in an appendix, Bloom's taxonomy levels for all Knowledge Areas for one software engineer profile: a graduate with four years of experience. This is the "target" of the SWEBOK Guide, as defined by what is meant by generally accepted knowledge.

[3] Table taken from http://www.nwlink.com/~donclark/hrd/bloom.html

To illustrate how Bloom's taxonomy levels could be used as a tool in defining job descriptions, software engineering role descriptions within a software engineering process definition, professional development paths, and professional training requirements, Tables 4 to 7 propose Bloom's taxonomy levels for two additional software engineer profiles: a new graduate and an experienced software engineer working in a software engineering process group. Evaluations are proposed for four Knowledge Areas: Software Maintenance, Software Engineering Management, Software Engineering Process, and Software Quality.

The motivation for the selection of these Knowledge Areas comprised the following:

- Software Engineering Management, Software Engineering Process, and Software Quality are strongly related through measurement issues.
- Software Engineering Management, Software Engineering Process, and Software Quality are all "secondary" processes (meaning, roughly, not primary processes) in the software life cycle, as described in the ISO/IEC 12207 [16] classification.
- Since software maintenance is so widely practiced in industry and often not treated explicitly in the literature, the Software Maintenance Knowledge Area was included in the selection.

The following guidelines were followed when proposing Bloom's taxonomy levels for these four Knowledge Areas:

- Very few topics were assigned a rating higher than Application for the new graduate profile. This is coherent with the approach taken in [8], where no topic of the Software Engineering Education Knowledge, a body of knowledge developed for university software engineering curriculum design purposes, is assigned a rating higher than Application.
- The synthesis rating was interpreted as being at the level of an experienced software engineer, and the evaluation level was interpreted as being relevant to an expert on a given topic. This is why these two ratings were only assigned to the profile of an experienced software engineer working in a software engineering process group. For this profile, topics that were strongly related to the duties of a software engineering process group, but not directly focused on the definition, management, and improvement of the software engineering processes themselves, were assigned the Synthesis rating (e.g. the majority of maintenance costs in the Software Maintenance Knowledge Area). Topics directly related to the duties of a senior software engineer employed in a software engineering process group were assigned the Evaluation rating (e.g. process planning in the Software Engineering Process Knowledge Area).

Some of the difficulties encountered while assigning these ratings were the following:

- Some levels are difficult to interpret for certain types of knowledge. For example, it is difficult to interpret the Application level for topics that are definitional in nature;
- Assigning ratings for only four Knowledge Areas out of ten is problematic, since a balance must be achieved across all Knowledge Areas;
- Assigning ratings for the profile of the graduate with four years of experience is difficult, since their practical experience may vary considerably. For example, in some organizations, a recent graduate may be given management duties for small projects, while in others this would rarely be the case. The assigned ratings for this profile should therefore be seen as "minimum requirements". Actual ratings for each individual will be higher in the Knowledge Areas and for topics more closely related to their practical experience.

Table 4. Bloom's Taxonomy Levels for Software Maintenance.

	NG	G+4	ESWE
I. Fundamentals			
Definitions and terminology	C	C	AN
Nature of maintenance	C	C	S
Need for maintenance	C	C	S
Majority of maintenance costs	C	C	S
Evolution of software	C	C	S
Categories of maintenance	AP	AP	S
II. Key issues in software maintenance			
Technical			
Limited Understanding	C	C	AN
Testing	AP	AP	AN
Impact Analysis	AP	AN	AN
Maintainability	C	AN	AN
Management issues			
Alignment with organizational objectives	C	C	S
Staffing	C	C	AN
Process	C	C	E
Organizational aspects of maintenance	C	C	AN
Outsourcing	C	C	AN
Maintenance cost estimation			
Cost estimation	C	AP	AN
Parametric models	C	C	AN
Experience	C	AP	AN
Software maintenance measurement	C	AP	AN
III. Maintenance process			
Maintenance processes	AP	C	E
Maintenance activities			
Unique Activities	C	AP	E
Supporting Activities	C	AP	E
IV. Techniques for maintenance			
Program comprehension	AP	AN	AN
Re-engineering	C	C	AN
Reverse engineering	C	C	AN

Legend: NG: New Graduate, G+4: Graduate with four years of experience, EWSE: Experienced software engineer working in a software engineering process group

K: Knowledge, C: Comprehension, AP: Application, AN : Analysis, S: Synthesis, E: Evaluation

Table 5. Bloom's Taxonomy Levels for Software Engineering Management.

	NG	G+4	ESWE
I. Initiation and scope definition			
Determination and negotiation of requirements	C	AP	AN
Feasibility analysis	AP	AP	AN
Process for the review and revision of requirements	C	C	E
II. Software project planning			
Process planning	C	C	E
Determine deliverables	AP	AP	S
Effort, schedule, and cost estimation	AP	AP	AN
Resource allocation	C	AP	AN
Risk management	C	AP	S
Quality management	C	AP	S
Plan management	C	C	S
III. Software project enactment			
Implementation of plans	AP	AP	S
Supplier contract management	C	C	AP
Implementation of measurement process	AP	AP	E
Monitor process	AP	AN	E
Control process	AP	AP	E
Reporting	AP	AP	E
IV. Review and evaluation			
Determining satisfaction of requirements	C	AP	AN
Reviewing and evaluating performance	AP	AP	S
V. Closure			
Determining closure	AP	AP	S
Closure activities	C	AP	S
VI. Software Engineering Measurement			
Establish and sustain measurement commitment	C	C	E
Plan the measurement process	C	C	E
Perform the measurement process	C	C	E
Evaluate measurement	C	C	E

Table 6. Bloom's Taxonomy Levels for Software Engineering Process.

	NG	G+4	ESWE
I. PROCESS IMPLEMENTATION AND CHANGE			
Process infrastructure			
Software engineering process group	C	C	E
Experience factory	C	C	E
Software process management cycle	AP	AP	E
Models for process implementation and change	C	K	E
Practical considerations	C	C	E
II. PROCESS DEFINITION			
Software life cycle models	C	AP	E
Software life cycle processes	C	C	E
Notations for process definitions	C	C	AP
Process adaptation	C	C	E
Automation	C	C	AP
III. PROCESS ASSESSMENT			
Process assessment models	C	C	AN
Process assessment methods	C	C	AN
IV. PRODUCT AND PROCESS MEASUREMENT			
Process measurement	AP	AP	AN
Software product measurement	AP	AP	AN
Size measurement	AP	AP	AN
Structure measurement	AP	AP	AN
Quality measurement	AP	AP	AN
Quality of measurement results	C	AN	AN
Software information models			
Model building	C	AP	AN
Model implementation	C	AP	AN
Process measurement techniques			
Analytical techniques	C	AP	AN
Benchmarking techniques	C	C	AN

Table 7. Bloom's Taxonomy Levels for Software Quality.

	NG	G+4	ESWE
I. SOFTWARE QUALITY FUNDAMENTALS			
Software engineering culture and ethics	AN	AN	AN
Value and costs of quality	AP	AN	S
Models and quality characteristics			
Software engineering process quality	AP	AN	E
Software product quality	AP	AN	S
Quality improvement	C	AP	S
II. SOFTWARE QUALITY MANAGEMENT PROCESSES			
Software quality assurance	C	AP	AN
Verification and validation	AP	AP	AN
Reviews and audits			
Management reviews	C	C	AN
Technical reviews	C	AP	AN
Inspection	AP	AP	AN
Walkthrough	AP	AP	AN
Audits	C	C	AP
III. PRACTICAL CONSIDERATIONS			
Application quality requirements			
Influence factors	C	C	AN
Dependability	C	C	AN
Integrity levels of software	C	C	AN
Defect characterization	C	AP	E
Software quality management techniques			
Static techniques	AP	AP	S
People-intensive techniques	AP	AP	S
Analytical techniques	AP	AP	S
Dynamic techniques	AP	AP	S
Testing	AP	AP	S
Software quality measurement	AP	AP	S

5. Summary.

This paper is the product of a workshop held in Amsterdam during the Software Technology and Practice Conference (STEP 2003). The paper presented, and provided examples of the usage of, the SWEBOK Guide, which seeks to identify and describe the subset of software engineering knowledge that is generally accepted. The six levels of Bloom's taxonomy of cognitive goals were then presented. Bloom's taxonomy levels are proposed for a subset of four Knowledge Areas for three software engineer profiles.

A second paper [17], resulting from the same STEP 2003 workshop that produced this paper, further discusses some of the difficulties presented in this paper which are encountered when applying Bloom's taxonomy to software engineering, and presents some alternative approaches and solutions to these challenges.

The aim of this paper has been to illustrate how software engineer profiles developed with Bloom's taxonomy could be used as a tool in defining job descriptions, software engineering role descriptions within a software engineering process definition, professional development paths, and training programs. Further work now needs to be accomplished in terms of conducting trials of the proposed profiles and refining them based on actual usage.

6. Acknowledgment

In addition to the authors, the contributors to this workshop were: Motei Azuma, François Coallier, Juan Garbajosa, Patricia Lago, and Hans Van Vliet.

7. References

[1] A. Abran, J. W. Moore, P. Bourque, and R. Dupuis, "Guide to the Software Engineering Body of Knowledge," 2004 Version, ed: IEEE Computer Society Press, 2004.

[2] B. Bloom, "Taxonomy of Educational Objectives: The Classification of Educational Goals," Mackay, 1956.

[3] K. Surendran, H. Hays, and A. Macfarlane, "Simulating a Software Engineering Apprenticeship," *IEEE Software*, vol. 19, pp. 49-56, 2002.

[4] S. Ramakrishnan and A. Cambrell, "An In-forming Web-based Environment for a Bachelor of Software Engineering Degree - DoIT," presented at IS2002 Informing Science + IT Education Conference, 2002.

[5] O. Benediktsson, "Software Engineering Body of Knowledge and Curriculum Development.," presented at Views on Software Development in the New Millennium, 2000.

[6] A. Abran, J. W. Moore, P. Bourque, and R. Dupuis, "Guide to the Software Engineering Body of Knowledge," Trial Version, ed: IEEE Computer Society Press, 2001.

[7] S. Ludi and J. Collofello, "An Analysis of the Gap between the Knowledge and Skills Learned in Academic Software Engineering Course Projects and those Required in Real Projects," presented at 31st ASEE/IEEE Frontiers in Education Conference, 2001. http://fie.engrng.pitt.edu/fie2001/papers/1187.pdf, 2001.

[8] "Computing Curricula - Software Engineering Volume - Public Draft 3.1," Joint Task Force on Computing Curricula - IEEE Computer Society Association for Computing Machinery 2004.

[9] P. Bourque, F. Robert, J.-M. J.-M. Lavoie, A. Lee, S. Trudel, and T. Lethbridge, "Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Body of Knowledge (SEEK) - A Preliminary Mapping," presented at Tenth International Software Technology and Engineering Practice Conference (STEP 2002),, Montreal, 2002.

[10] S. McConnell, *Professional Software Development*: Addison-Wesley, 2004.

[11] "A Guide to the Project Management Body of Knowledge," Project Management Institute 2000.

[12] D. Frailey and J. Mason, "Using SWEBOK for Education Programs in Industry and Academia," presented at 15th Conf. Software Engineering Education and Training Conference (CSEET 2002), 2002.

[13] Y. Belkebir, "Analyse et amélioration des définitions de rôles du processus d'ingénierie logicielle du centre de compétence en génie logicielle de Bombardier Transport," in *Dept. of Electrical Engineering*. Montreal: École de technologie supérieure, 2003.

[14] T. Aytaç, S. Ikiz, and M. Aykol, " A SPICE-Oriented, SWEBOK-Based, Software Process-Based Assessment on a National Scale: Turkish Sector Software Survey - 2001," presented at 3rd International SPICE Conference, 2003.

[15] "Software Engineering Definitive Report," Ordre des ingénieurs du Québec, Montréal, Québec 2000.

[16] "IEEE/EIA 12207.0-1996 IEEE/EIA Standard Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207) Standard for Information Technology Software Life Cycle Processes," Institute of Electrical and Electronics Engineers 1998.

[17] M. Azuma, F. Coallier, and J. Garbajosa, "How to Apply Bloom's Taxonomy to Software Engineering," presented at STEP 2003 - Software Technology and Practice, Amsterdam, 2004.