The Impact of SWEBOK Version 3 on Software Engineering Education and Training

Richard E. (Dick) Fairley
Software and Systems Engineering Associates, USA
d.fairley@computer.org

Pierre Bourque
École de technologie supérieure, Canada
pierre.bourque@etsmtl.ca

John Keppler
IEEE Computer Society, USA
jkeppler@computer.org

Abstract

The Guide to the Software Engineering Body of Knowledge (the SWEBOK Guide) provides generally accepted knowledge for the software engineering profession. The content of the SWEBOK Guide is derived from academic sources and the best practices of the profession. The SWEBOK Guide is a reference document for many academic and industrial curricula, certification programs, accreditation criteria, and professional licensure. This paper provides an overview of the recently published Version 3 of the SWEBOK Guide and examines the potential impact of the changes in Version 3 on software engineering education and training for the academic and industrial sectors.

1. Introduction

The Guide to the Software Engineering Body of Knowledge (the SWEBOK Guide) is an acknowledged foundation document for the software engineering profession [1]. Publication of the 2004 version of the Guide (SWEBOK 2004) was an important milestone in establishing software engineering as a recognized engineering discipline. Many academic curricula and industrial training programs are based on the SWEBOK Guide, as are certification programs, accreditation criteria, and licensure criteria. Version 3 of the SWEBOK Guide has recently been published. The goal in developing this update was to improve the currency, readability, consistency, and usability of the Guide.

This paper provides an overview of SWEBOK Version 3 and describes the likely impact of changes in Version 3 on software engineering education and training for the academic and industrial sectors.

The objectives for the SWEBOK Guide have been, and continue to be:

1. To promote a consistent view of software engineering worldwide.
2. To specify the scope of software engineering with respect to other related disciplines.
3. To characterize the contents of the software engineering discipline.
4. To provide topical access to the Software Engineering Body of Knowledge.

978-1-4799-4970-0/14/$31.00 © 2014 IEEE CSEE&T 2014, Klagenfurt, Austria

For SWEBOK Version 3, objective 1 – to promote a consistent view of software engineering worldwide – has been achieved by a development process that engaged approximately 150 reviewers from 33 countries, whose comments were adjudicated during preparation of Version 3. The editors who edited the knowledge areas were from eight countries.

Objective 2, to specify the scope of software engineering with respect to other related disciplines, has been achieved by defining the scope of the software engineering discipline in terms of the first 10 knowledge areas (KAs) listed below. In addition, seven related disciplines are listed below.

Objective 3, to characterize the contents of the software engineering discipline, has been achieved by using a hierarchical structure for content. The hierarchical organization partitions each KA into a set of topics with distinct and recognizable labels. Furthermore, each topic is further broken down into subtopics. The topics in each KA are included in column 2 of the Appendix to this paper.

Objective 4, to provide topical access to the software engineering body of knowledge, has been achieved by identifying and citing authoritative reference material for the topics in the KAs. A matrix that relates the cited references to the KA topics is included in each KA.

Objective 5, to provide a foundation for curriculum development, certification, and licensing has been achieved by applying the criterion that generally accepted knowledge is included and cited in the references. Advanced, research, and specialized knowledge has been excluded.

SWEBOK 2004 includes 10 knowledge areas. SWEBOK Version 3 has been expanded to include 15 knowledge areas; five new KAs have been added and the Software Engineering Methods and Tools KA in SWEBOK 2004 has been replaced with a Software Engineering Models and Methods KA in SWEBOK Version 3. A software engineering tools topic is included in each of the other nine original KAs. The original nine KAs and the six new KAs are listed as follows. Italics are used to denote the new KAs.

Software Requirements
Software Design
Software Construction
Software Testing
Software Maintenance
Software Configuration Management
Software Engineering Management
Software Engineering Process
Software Quality
Software Engineering Models and Methods
Software Engineering Professional Practice
Software Engineering Economics
Computing Foundations
Mathematical Foundations
Engineering Foundations

2. Related Disciplines

In addition to the 15 knowledge areas, SWEBOK Version 3 lists seven related disciplines that intersect with software engineering. They are:

Computer engineering
Systems engineering
Project management
Quality management
General management
Computer science
Mathematics

These seven related disciplines indicate the scope and boundaries of software engineering practice. It is not an objective of SWEBOK Version 3 to describe the knowledge content of these related disciplines other than to describe the relevant elements of computer science and mathematics in the Computing Foundations and Mathematical Foundations KAs of this Guide.

3. Six New Knowledge Areas

As indicated above, the six new knowledge areas in SWEBOK Version 3 include a Software Engineering Models and Methods KA, a Software Engineering Professional Practice KA, and four foundation knowledge areas. The models and methods KA replaces the methods and tools KA in SWEBOK 2004; the professional practices KA expands SWEBOK to address a perceived need; and the four foundation KAs provide foundations for the other KAs in Version 3. A brief overview of these new knowledge areas follows.

Software Engineering Models and Methods vary widely in scope; they range from models and methods that address individual phases of the software life cycle to those that cover the full software life cycle. Software engineering models provide approaches to problem solving, notations, and procedures for model construction and analysis. Methods provide an approach to the systematic specification, design, construction, test, and verification of the end-item software and associated work products. The emphasis in this knowledge area is on the software engineering models and methods that encompass multiple phases of the software life cycle; models and methods specific to single life cycle phases are covered in other KAs.

Software Engineering Professional Practice has been included in SWEBOK V3 because software engineers should have the knowledge, skills, and attitudes needed to practice software engineering in a professional, responsible, and ethical manner. The topics included in the Professional Practice KA are listed in the Appendix to this paper; the topics are professionalism, group dynamics and psychology, and communication skills.

Software Engineering Economics has been included as a foundation KA because software engineers make engineering decisions within a business context. The success of a software product, service, or solution depends on understanding the business context and the impact of engineering decisions on business practices; the converse impact of business practices on software requirements and design constraints must also be understood by software engineers. This knowledge area provides an overview of the software engineering economics topics listed in the Appendix.

Computing Foundations in SWEBOK Version 3 is concerned with the computer science foundations that support the design and construction of software products. It also includes

basic knowledge about transforming a software design into a software implementation; the tools used during this process; and the various software development methods.

Mathematical Foundations provides knowledge of mathematics used in software engineering and presents the unambiguous rules needed to apply logical reasoning to software products being developed and modified.

Engineering Foundations includes engineering skills and techniques that are useful for a software engineer. This knowledge area is included because at its core software engineering is an engineering discipline. The focus is on topics that support other knowledge areas while minimizing duplication of topics covered elsewhere in SWEBOK Version 3.

4. Other New and Modified Elements

Software Engineering Process in SWEBOK Version 3 has significantly different structure and content from the software process KA in SWEBOK 2004. The changes in this KA reflect the significant advances in understanding and application of software engineering processes during the intervening time period.

User Interface Design is a new subarea in the Software Design knowledge area. It includes seven topics that range from General User Interface Design Principles to Metaphors and Conceptual Models.

Software Construction Technologies is a new subarea in the Software Construction knowledge area of SWEBOK Version 3. Sixteen topics are included in this subarea; they range from API Design and Use to State-Based and Table-Driven Construction Techniques to Performance Analysis to Platform Standards.

Software Engineering Tools: A Software Engineering Methods and Tools KA was included SWEBOK 2004, with a separate subarea for tools in each of the other nine KAs. Software engineering tools are included as subareas in each of the first nine KAs of Version 3 listed above, as is apparent in the Appendix. Software Engineering Methods (formerly in the Software Engineering Methods and Tools KA) is included in the Software Engineering Models and Methods KA of SWEBOK Version 3.

Modified Topics and Subareas: Most of the breakdowns of topic names in the knowledge areas of SWEBOK Version 3 are retained from SWEBOK 2004 and the topics have been updated to reflect the evolution of software engineering during the intervening period. Some subareas in SWEBOK 2004 have been modified, while some others have been deleted, and some new topics have been added. Space limitations prevent inclusion of subtopic details in this paper; the reader is referred to SWEBOK Version 3.

Appendix A: Knowledge Area Description Specifications contains the specifications provided to the knowledge area editors who prepared the knowledge areas. This appendix enables readers, reviewers, and users to clearly understand the specifications that were used regarding the format and content of this version of the SWEBOK Guide.

Appendix B: Standards provides an introduction to the applicability of standards in software engineering and extensive. An annotated list of IEEE and ISO/IEC standards applicable to each of the KAs is included.

Appendix C: the Consolidated Reference List contains a list of approximately 40 recommended references that are cited within the topics of the fifteen knowledge areas.

Objectives for this Consolidated Reference List are:

Complete: Covering the entire scope of the SWEBOK Guide.

Sufficient: Providing enough information to describe generally accepted knowledge.

Consistent: No contradictory knowledge or conflicting practices.

Credible: Recognized as providing expert treatment.

Current: Treating the topics in a manner that is commensurate with current and generally accepted knowledge.

Succinct: As short as possible (both in number of reference items and in total page count) without failing other objectives.

The Consolidated Reference List (CLR) has replaced the one in SWEBOK 2004 to update it and organize it to be much more useful. KA Editors who prepared the knowledge areas used the references from the CLR as the primary references allocated to their KAs. In addition, many of the KA Editors included Further Readings that are helpful but not essential to gaining increased understanding of the material in the knowledge areas.

5. Impacts of SWEBOK Version 3 on Education and Training

There are many potential impacts of SWEBOK Version 3 on software engineering education and training. The degree of impact will vary depending on the level of education or training being addressed.

In the context of undergraduate and graduate academic curricula, modifications and updates to existing course materials may be required to prepare the next generation of software engineers and to upgrade the knowledge base of practicing software engineers. New graduates and practicing software engineers should also have some knowledge of related disciplines such as project management and systems engineering.

For continuing education and professional development these related disciplines offer a fertile ground for development of educational and training materials. Case studies and practical examples of relevant experiences in the field can provide a valuable education for professionals already active in the field.

Moreover, the four new foundation knowledge areas include topics that provide the foundation elements for a software engineering degree program. Existing programs and new programs of study can use these knowledge areas to determine the extent to which they include, or will include the topics in these knowledge areas in their curricula. The new Software Engineering Professional Practice and Software Engineering Models and Methods KAs add important knowledge areas to the software engineering body of knowledge that may be added to, or enhanced in academic curricula and industrial training programs. In a similar manner, the updated Software Engineering Process KA and the new subareas of User Interface Design (in the Software Design KA) and Software Construction Technologies (in the Software Construction KA) provide information and references for including those subareas in revised and new curricula and training programs. Significant material has been added on software engineering models in the Software Engineering Models and Methods KA, which provides a basis for increased emphasis in educational curricula and industrial training programs.

Software engineering tools have received increased emphasis in SWEBOK Version 3 by including them as subareas in the nine updated knowledge areas retained from SWEBOK 2004. Educators and trainers can use this knowledge to increase the emphasis on software engineering tools in their curricula and in their educational and training laboratories.

SWEBOK Version 3 is a foundation document for the Certified Software Development Associate) [2] and Certified Software Development Professional [3] certification programs sponsored by the IEEE Computer Society. Questions included in the CSDA and CSDP examinations are traced to topics in the SWEBOK Guide. In order to preserve the integrity of the examination process those who prepare training material are not allowed to see the questions in the CSDA and CSDA examination question banks. They based the training material on the SWEBOK Guide and the cited references, on which the questions are based, to maintain alignment with the examination questions. Updates to the CSDA and CSDP training materials and exams will reflect the new and modified material in the Guide. In the United States, professional engineer licensure training material and examination questions for software engineers are based in part on the SWEBOK Guide and will likely be updated to reflect the changes in SWEBOK Version 3 [4].

The Computer Science Accreditation Board of ABET (CSAB) specifies the accreditation criteria for ABET accredited programs in computer science, software engineering, information technology, and information science. The SWEBOK Guide is a foundation document for establishing and updating the software engineering accreditation criteria. The CSAB members will be invited to review SWEBOK Version 3 for potential changes in the criteria.

The Professional Activities Board of the IEEE Computer Society is currently developing a Software Engineering Competency Model (SECOM) that is based primarily on SWEBOK Version 3. SECOM describes skill areas (e.g., requirements engineering), skills within skill areas (e.g., elicitation), and activities within skills (e.g., prototyping to elicit requirements). Activities within skills are specified at five levels of increasing competency. SECOM provides a basis for competency-based education and training programs. Academics can use SECOM to develop or modify learning objectives and educational outcomes for their students and modify their curricula accordingly. Industrial trainers can use SECOM to develop competency-based training programs to achieve needed competency levels for designated activities, skills, and skill areas.

Appendix B of the SWEBOK Guide (Standards) provides annotated descriptions of standards applicable to each knowledge area that can be incorporated into education and training programs and that can provide authoritative basis material for academic curricula and industrial training programs.

The approximately 40 references cited in Appendix C (Consolidated Reference List) can be consulted for additional information beyond the descriptions in the knowledge areas. Each knowledge area provides relevant references from the consolidated list of references. Educational and industrial organizations may want to acquire some or all of the cited references to provide a library resource for software engineering. Some of the Further Readings may also be acquired for areas of specialized interest.

6. Conclusion

SWEBOK Version 3 reflects changes in software engineering that have occurred since publication of SWEBOK 2004. Major changes include addition of four new foundation knowledge areas (KAs), a new Software Engineering Professional Practices KA, and a Software Engineering Models and Methods KA has replaced the Software Engineering Methods and Tools KA; Software Engineering Tools is now a subarea in each of the nine KAs retained from SWEBOK 2004. The Software Engineering Process KA has been extensively updated and the new subareas of User Interface Design in the Software Design KA and Software Construction Technologies in the Software Construction KA have been added. Three appendices provide the specifications for the KA descriptions, an annotated set of relevant standards for each KA, and a listing of the references cited in the Guide. In addition, many smaller changes have been made to improve the currency, readability, consistency, and usability of the knowledge areas in the Guide.

This paper has also described the potential impacts of the new and modified material in SWEBOK Version 3 on software engineering education and training. The actual impacts will, of course, be determined by the initiatives of educators, trainers, certifiers, licensers, and accreditors who use SWEBOK as the foundation document for their programs.

References

[1] Guide to the Software Engineering Body of Knowledge (SWEBOK Version 3), IEEE Computer Society, 2014, www.swebok.org

[2] Certified Software Development Associate http://www.computer.org/portal/web/certification/csda

[3] Certified Software Development Professional http://www.computer.org/portal/web/certification/csdp

[4] NCEES introduces PE exam for software engineering http://ncees.org/about-ncees/news/ncees-introduces-pe-exam-for-software-engineering/

Appendix

SWEBOK Version 3 Knowledge Areas and Topics

Italics indicate new knowledge areas and topics in SWEBOK Version 3

Knowledge Areas	**Topics**
Software Requirements	Requirements fundamentals, requirements process, requirements elicitation, requirements analysis, requirements specification, requirements validation, practical considerations, and software requirements tools.
Software Design	Design fundamentals, key issues in software design, software structure and architecture, user interface design, software design quality analysis and evaluation, software design notations, software design strategies and methods, software design tools.
Software Construction	Software construction fundamentals, managing construction, practical

	considerations, construction technologies, software construction tools.
Software Testing	Software testing fundamentals, test levels, test techniques, test related measures, test process, software testing tools.
Software Maintenance	Software maintenance fundamentals, key issues in software maintenance, maintenance process, techniques for maintenance, software maintenance tools.
Software Configuration Management	Management of the SCM process, software configuration identification, software configuration control, software configuration status accounting, software configuration auditing, software release management and delivery, software configuration management tools.
Software Engineering Management	Initiation and scope definition, software project planning, software project enactment, review and evaluation, closure, software engineering measurement, software engineering management tools.
Software Engineering Process	Software process definition, software life cycles, software process assessment and improvement, software measurement, software engineering process tools.
Software Quality	Software quality fundamentals, software quality management processes, practical considerations, software quality tools.
Software Engineering Models and Methods	Modeling, types of models, analysis of models, software engineering methods.
Software Engineering Professional Practice	Professionalism, group dynamics and psychology, communication skills.
Software Engineering Economics	Software engineering economics fundamentals, life cycle economics, risk and uncertainty, economic analysis methods, practical considerations
Computing Foundations	Problem solving techniques, abstraction, programming fundamentals, programming language basics, debugging tools and techniques, data structure and representation, algorithms and complexity, basic concept of a system, computer organization, operating systems basics, compiler basics, data base

	basics and data management, network communication basics, parallel and distributed computing, basic user human factors, basic developer human factors, secure software development and maintenance.
Mathematical Foundations	Sets, relations, functions; basic logic; proof techniques; basics of counting; graphs and trees; discrete probability; finite state machines; grammars; numerical precision, accuracy, and errors; number theory; algebraic structures.
Engineering Foundations	Empirical methods and experimental techniques; statistical analysis; measurement; engineering design; modeling, simulation, and prototyping; standards; root cause analysis.