
Approved Baseline for a List of Knowledge Areas
for the Stone Man Version of the Guide to the
Software Engineering Body of Knowledge

Approved by the Industrial Advisory Board
Guide to the Software Engineering Body of Knowledge Project

Prepared by

Pierre Bourque
Robert Dupuis
Alain Abran
James W. Moore
Leonard Tripp
Dennis Frailey

January 19, 1999

Goal of Document

At the kick-off meeting of the Industrial Advisory Board held in Mont-Tremblant, it was agreed upon:

- That the list of Knowledge Areas proposed in the Straw Man report are within the scope of the Guide to the SWEBOK, with the following modifications:
 - Remove *detailed* and keep *design* as a whole
 - Add *management process*
 - Replace *reliability* with *dependability*
- That the following topics be covered in some fashion in the Guide to the SWEBOK:
 - Performance and conformance testing
 - Standard designs
 - Integration
- The number of Knowledge Areas should be reduced from the 17 suggested in the Straw Man report.

An action item was therefore set for the Editorial Team to suggest a shorter list of KAS based notably on the discussions held in Mont-Tremblant. This document presents such a proposal.

Related Disciplines

It was agreed by the Industrial Advisory Board that the Stone Man version of the Guide to the SWEBOK would present the core Body of Knowledge, i.e. the generally accepted knowledge in the field expected from a graduate with four years of experience. It was also agreed in Mont-Tremblant that expected knowledge of other Related Disciplines would only be referenced in the Guide to the SWEBOK. Additionally, these references will be more oriented towards definitions and basic concepts rather than content material *per se*.

For the reader to better evaluate the proposed baseline for a list of Knowledge Areas within the complete picture, these Related Disciplines are listed below. As presented and discussed in Mont-Tremblant, they are:

- Computer Science
 - For the moment, the reference for this Related Discipline is Computing Curricula 1991 - Report of the ACM/IEEE-CS Joint Curriculum Task

Force¹. These curricula are currently being updated through an initiative called the IEEE Computer Society and ACM Joint Task Force on "Year 2001 Model Curricula for Computing: CC-2001". To ensure proper coordination with this initiative, Carl Chang, Joint Task Force Co-Chair is a member of the Industrial Advisory Board and was present in Mont-Tremblant.

- Project Management
 - The reference for this Related Discipline will be "A Guide to the Project Management Body of Knowledge » published by the Project Management Institute². This document is currently being adopted as an IEEE software engineering standard.
- Computer Engineering
- Mathematics
 - It was agreed in Mont-Tremblant that the Computing Curricula 2001 initiative would be the "conduit" to mathematics.
- Telecommunications/Networks
- Management
- Science
- Other Engineering Disciplines
- Cognitive Sciences
- Management Sciences
- Systems Engineering (to be determined)

Criteria and Requirements for the proposed baseline for a list of Knowledge Areas

The following requirements and criteria were used in proposing this baseline for a list of Knowledge Areas:

- The proposed baseline for a list of Knowledge Areas must decompose the subset of the Software Engineering Body of Knowledge that is "generally accepted."³

¹ See <http://computer.org/educate/cc1991/>

² See www.pmi.org to download this report

³ The Project Management Institute in its Guide to the Project Management Body of Knowledge defines "generally accepted" knowledge for project management in the following manner:

- The proposed baseline for a list of Knowledge Areas must not presume specific application domains, business needs, sizes of organizations, organizational structures, management philosophies, software life cycle models, software technologies or software development methods.
- The proposed baseline for a list of Knowledge Areas must cover every candidate Knowledge Area presented in the Straw Man version of the Guide to the Software Engineering Body of Knowledge. The mapping between the list of candidate Knowledge Areas in the Straw Man version and the proposed baseline for a list for the Stone Man version must be documented.
- The proposed baseline for a list of Knowledge Areas must cover the additional «expectations» put forward by the Industrial Advisory Board in Mont-Tremblant and not covered in the list of candidate Knowledge Areas presented in the Straw Man version of the Guide to Software Engineering Body of Knowledge. The mapping of these additional «expectations» with the proposed baseline for a list of Knowledge Areas must be documented.
- The proposed baseline for a list of Knowledge Areas must contain a smaller number of Knowledge Areas than 17.
- The proposed baseline for a list of Knowledge Areas must be “compatible” with the breakdown of software engineering generally found in industry and in the software engineering literature and standards.
- The proposed baseline for a list of Knowledge Areas must be checked for major omissions against other Body of Knowledge proposals.
- Proposed Knowledge Area names should be general enough to allow references to knowledge, skills, tools, methods and techniques.
- Proposed Knowledge Area names must be significant enough to be meaningful even when cited outside the Guide to the Software Engineering Body of Knowledge.

“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 Guide to the Project Management Body of Knowledge is now an IEEE Standard.

In Mont-Tremblant, the Industrial Advisory Board better defined “generally accepted” as 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.

Proposed Baseline of a List of Knowledge Areas

The proposed list of Knowledge Areas is intended as a baseline, i.e. an agreed upon starting point, for the Stone Man phase. Based on the work and the Knowledge Area descriptions that will be completed during the Stone Man phase, adjustments can be made to this list. Based on the criteria and requirements stated above, the proposed baseline for a list of Knowledge Areas is:

- Software Requirements Analysis
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Quality Analysis
- Software Engineering Management⁴
- Software Engineering Infrastructure
- Software Engineering Process

Please take note that we still have not figured how we should handle “orthogonal” topics that “transcend” or “run across” the entire proposed baseline for a list and are therefore difficult to place within a given Knowledge Area. Examples of such topics could be “software measurement for technical rather than managerial purposes”, “technical risk identification” or reuse.

Some of these topics are probably more suitably presented as a whole and should be discussed within another Knowledge Area. Other “orthogonal “ topics are probably more appropriately discussed within each Knowledge Area. Which criteria should be used to make such decisions remains an outstanding issue and we are looking for options to resolve it judiciously. This issue was discussed in Mont-Tremblant.

In point form, here are some ideas that may help move along this issue:

⁴ Since this refers to the management of software engineering *per se* rather than management tasks completed by the software itself (I/O management, memory management, etc.), this proposed Knowledge Area is called “Software Engineering Management” rather than “Software Management”. The same can be said of “Software Engineering Infrastructure” and “Software Engineering Process”.

- This will always be a difficult issue and no matter how we decide to handle these themes we will be criticized.
- We may not be able to resolve this issue until we better understand the decomposition of topics within each Knowledge Area. This will happen when 1) all the jumpstart documents are finished in mid-January and more importantly when the Knowledge Area Specialists have delivered version 0.1 of their Knowledge Area description in mid-March⁵.
- We could state this as an open issue in our "Knowledge Area Description" document and ask for preferences on how to handle it from the Knowledge Area Specialists.
- For discussion purposes, here are a few ways on how these types of "orthogonal topics" could be handled:
 - Group them together into a single "infrastructure" type of Knowledge Area while stating that this Knowledge Area is different from the other ones.
 - We could name the other Knowledge Area Specialists as reviewers of this Knowledge Area.
 - We could ask each Knowledge Area Specialist to delegate a "reviewer" for this Knowledge Area.
 - Discuss these "orthogonal topics" within each Knowledge Area:
 - We could add a mandatory section in each Knowledge Area for each one of these topics
 - We could name an "associate editor" for each one of these topics who would be responsible for editing the content of these sections and this for each Knowledge Area.
 - What is applicable within these topics across all Knowledge Areas could be discussed within one of the relevant Knowledge Areas while what is specific to a given Knowledge Area could be discussed within that specific given Knowledge Area.

Appendix 1 documents the mapping between the list of candidate Knowledge Areas in the Straw Man version of the Guide to the Software Engineering Body of Knowledge and the proposed baseline for a list of Knowledge Areas. Appendix 1 also indicates where the "additional expectations" put forward by the Industrial Advisory Board in Mont-Tremblant should be discussed.

⁵ See proposed plan for the Stone Man version.

Appendix 2 documents the mapping between the proposed baseline for a list of Knowledge Areas and three other Software Engineering Body of Knowledge proposals. It also documents a mapping between the proposed baseline for a list Knowledge Areas and the processes described in the ISO/IEC 12207 standard.

Appendix 1 Mapping between proposed baseline for a list of Knowledge Areas and Straw Man Knowledge Areas and additional “expectations” put forward by the Industrial Advisory Board in Mont-Tremblant.

This table is presented to show general mapping. The Knowledge Area Specialists will not be restricted to the contents of this table and will be free to suggest any topic they consider “generally accepted” within their assigned Knowledge Area.

Proposed Baseline Stone Man Knowledge Area	Corresponding Straw Man Knowledge Areas	Notes
Software Requirements Analysis	Requirements Analysis	
Software Design	Detailed Design	The IAB decided in Mont-Tremblant to cover all “design” from the start and not to distinguish between “architectural design” and “detailed design”. This distinction is made in the 12207 standard.
Software Construction	Coding	According to 12207, coding includes unit testing. This would be true in the Guide to the SWEBOK as well.
Software Testing	Testing	Performance and conformance testing are expectations put forward by the IAB in Mont-Tremblant. They would be discussed within this Knowledge Area.
Software Maintenance	Maintenance Process	
Software Configuration Management	Configuration Management	
Software Quality Analysis	Quality Assurance Verification and Validation Software Reliability	All “quality-related” areas have been grouped in a single Knowledge Area. Dependability was also an expectation of the IAB in Mont-Tremblant. It would be discussed within this Knowledge Area.

Proposed Baseline Stone Man Knowledge Area	Corresponding Straw Man Knowledge Area	Notes
Software Engineering Management	Management Process Measurement/Metrics	The definition of this process in 12207 mentions the use of quantifiable data for decision-making. Measurement/Metrics was therefore grouped with "Management process".
Software Engineering Infrastructure	Development Methods (object-oriented, formal methods, prototyping) Software Development Environments	<p>Moore associates tools (environments) and reuse as the two main elements of "infrastructure process". Reuse here refers to 'reuse libraries', the practice of reuse should be covered in each life-cycle dependant Knowledge Area concerned. Methods would be added here since the tools are also often built to implement a particular method, and thus it would seem logical to cover both together. It seems difficult to cover tools without covering methods and vice-versa.</p> <p>Standard designs are associated to reuse. As discussed by the IAB in Mont-Tremblant, "integration" refers more to the integration of off-the-shelf components than only to the integration of modules developed within a system. For this reason, emphasis would be put on the explicit problems of this type of integration. If it is decided that emphasis be put on the "life-cycle operation of integration", then an additional Knowledge Area would be included between software construction and software testing, as in 12207.</p> <p>"Standard designs" and "integration" are expectations put forward by the IAB in Mont-Tremblant. They would be discussed within this Knowledge Area.</p>

Proposed Baseline Stone Man Knowledge Area	Corresponding Straw Man Knowledge Area	Notes
Software Engineering Process	Improvement Process	
	Software Engineering Overview & Definition	<p>This was included in the Straw Man list because it is in virtually all textbooks, and we will certainly need some introduction to the Stone Man version, but it is not included as a Knowledge Area as such. Once we have established the final format of the Stone Man version, we will decide how this introduction will be produced and who will produce it.</p>

Appendix 2 Mapping between proposed baseline for a list of Knowledge Areas and other Body of Knowledge proposals and the ISO/IEC 12207 standard

This table is presented to show general mapping. The Knowledge Area specialists will not be restricted to the contents of this table and will be free to suggest any topic they consider “generally accepted”.

Proposed Baseline Stone Man Knowledge Area	ISO/IEC 12207 Standard	D.L. PARNAS ⁶	“Reeker-Wallace Proposal” ⁷	WGSEET ⁸⁹
	PRIMARY PROCESSES			
	Development Process			
Software Requirements Analysis	Requirements Analysis	Participate in the DESIGN of the COMPUTER SYSTEM, determining which functions will be implemented in software, and selecting the basic hardware and software component Analyze the intended application to determine the REQUIREMENTS that must be satisfied and record those requirements in a precise, well-organised, and easily-used document.	Domain Analysis and Requirements Elicitation; Functional System Specification and Analysis	Software Requirements (core area component a)

⁶ [1] D. L. Parnas, “Software Engineering Programmes are not Computer Science Programmes,” McMaster University, Hamilton, Ontario CRL Report no. 361, April 1998.

⁷ Correspondance between Dolores Wallace and Larry Reeker from NIST and UQAM

⁸ The following proposed Knowledge Components are not included in this table because they are not within the scope of the Stone Man version of Guide to the Software Engineering Body of Knowledge.:

Foundations Area: Computing Fundamentals Component, Human Factors Component, Application Domain Component

Recurring Area: Ethics and Professionalism

Supporting Area: These components are by definition related to other fields of study than software engineering

⁹ This document was supplied to us by Tom Hilburn, member of the Working Group on Software Engineering Education and Training. See <http://www.sei.cmu.edu/collaborating/ed/workgroup-ed.html>

Proposed Baseline Stone Man Knowledge Area	ISO/IEC 12207 Standard	D.L. PARNAS	“Reeker-Wallace Proposal”	WGSEET
Software Design	Architectural Design Detailed Design	DESIGN the basic structure of SOFTWARE, i.e. its division into modules, the interfaces for those modules, and the structure of individual programs while precisely documenting all software design decisions	System design User Interfaces	Software Design (core area component b)
Software Construction	Coding	IMPLEMENT the software as a set of well structured and well documented programs	Implementation/Coding and Internal Documentation	Software Construction (core area component c)
	Integration	INTEGRATE new software with existing “off the shelf” software		
Software Testing	Testing	Perform systematic and statistical TESTING of the software and integrated computer system	System testing Trouble-shooting	
	Installation			
	Acceptance Support			
	OPERATION PROCESS			
	System Operation			
	User Support			

Proposed Baseline Stone Man Knowledge Areas	ISO/IEEE 12207 Standard	D.L. PARNAS	“Reeker-Wallace Proposal”	WGSEET
Software Maintenance	MAINTENANCE PROCESS	REVISE AND ENHANCE software systems, maintaining their conceptual integrity and keeping all documents complete and accurate	Improvement, Maintenance and Updating	Software Evolution (core area component e)
	SUPPORTING PROCESSES			
	Documentation		User Documentation and Aids	Documentation (recurring area component g)
Software Configuration Management	Configuration Management		Configuration Management	Project Management (core area d)
Software Quality Analysis	Quality Assurance V & V Joint Review Audits	ANALYZE the software structure for COMPLETENESS, CONSISTENCY, and SUITABILITY for the intended application ANALYZE the performance of a proposed design (either analytically or by simulation) to make sure that the proposed system can meet the application's requirements	Dependability Verification & Validation	Software Quality (recurring area component c)
	Problem Resolution Processes			

Proposed Baseline Stone Man Knowledge Areas	ISO/IEC 12207 Standard	D.L. PARNAS	“Reeker-Wallace Proposal”	WGSEET
	ORGANIZATIONAL PROCESSES			
Software Engineering Management	Management Processes		Measurement and Benchmarking	Project Management (core area component d) Software Metrics (recurring area component e)
Software Engineering Infrastructure	Infrastructure Process		Object Oriented Methodology; Formal Methods System Integration	Tools and Environments (recurring area component f) Software Modeling (recurring area component d)
Software Engineering Process	Improvement Process			Software Processes (recurring area component b)
	Training Process			