SWEBOK: Software engineering management knowledge area description version 0.5

Stephen G. MacDonell and Andrew R. Gray
Software Metrics Research Laboratory
University of Otago, Dunedin, New Zealand
+64 3 479 8135 (phone) +64 3 479 8311 (fax)
{stevemac}{agray}@infoscience.otago.ac.nz

1 Introduction

This is the current draft (version 0.5) of the knowledge area description for software engineering management. The relevant jump-start document [Gra99] was quite detailed in certain respects and some of its material has been incorporated here (with suitable modifications and extensions). Refinements from version 0.1 of this document reflect the more detailed presentation required and some much appreciated feedback from reviewers. It is apparent that there are several different viewpoints as to the breakdown of this topic and the reviewers' alternative perspectives have been very helpful in improving this document – both in terms of clarity and content.

The primary goal of this draft is to present the breakdown of the knowledge area into topics, list reference material, and to provide the topic-reference matrix. Additionally, an initial sketch of Bloom's taxonomy for certification courses is included. The tables showing Vincenti's classifications for the topics are still being revised from those shown in the jump-start document and version 0.1 of the draft, as is the table of standards and references.

Much of this material is therefore still incomplete, as is to be expected with a draft document at this stage of its life. A considerable amount of additional material is included to assist the authors and reviewers to work towards the final version. The eventually completed document is expected to be much closer to the ten page suggested guideline from [BDA+99] once this material is removed.

The software engineering management knowledge area consists of both the measurement/metrics and management process sub-areas (as defined below based on the jump-start document). Whilst these two topics are often regarded (and generally taught) as being separate, and indeed they do possess many mutually unique aspects, their close relationship necessitates the combined treatment taken here as part of the SWEBOK. In essence, management without measurement – qualitative or quantitative – suggests a lack of rigor and measurement without management suggests a lack of purpose or context

The following working definitions are based on those from the software engineering management jump-start document [Gra99].

> Measurement/metrics refers to the assignment of values and labels to aspects of software development (products, processes, and resources) and the models that may be derived therefrom.
>
> Management process refers to the activities that are undertaken in order to ensure that the software development process is performed in a manner consistent with the organization's policies, goals, and requirements.
>
> The management process area makes (in theory at least) extensive use of the measurement/metrics area – ideally this exchange between the two processes occurs continuously throughout the development process.

As is apparent from these definitions, the software engineering management knowledge area overlaps with several other knowledge areas, most notably, software construction, software testing, configuration management, and software quality analysis. This overlap is explored a little more fully in Appendix E.

2 References used

All of the references from the jump-start document are used here, namely [DT97, FP97, Moo98, Pfl98, Pre97b, Som96][1]. These form a core of texts that provides a solid understanding of most of the area and are all expected to be updated with sufficient regularity to ensure timely and complete coverage. The latter point is of greatest importance for the management process area which encompasses many disparate topics.

[1] With respect to the [DT97] text, only a subset of the papers were used in the jump-start document. They are [Boe97a, Boe97b, Bro97a, Bro97b, BB97, Car97, Com97, DeM97, Fai97, Gib97, Hee97, Hur97, NB97, Pau97, Pre97a, Tha97a, Tha97c].

An additional reference added is a basic statistics text to complement and extend the material covered in [FP97] – which is by necessity limited in treatment. The text we have chosen is [MM98] although any general introductory statistics book would suffice equally well here. However this text is very highly regarded for introductory statistics courses and should be fairly accessible to software engineers in general. As well as having excellent presentation it contains a plentiful source of sample applications that should be useful in software engineering management practice.

Other references added include [Rei97], [Tha97b], [Kar96], and [Zus97]. [Rei97] contains extensive supplementary material on the management process topics presented in the breakdown, whilst [Zus97] provides more comprehensive theoretical coverage of the measurement/metrics area. [Tha97b] complements the collections of papers found in [Rei97] and [DT97], although there is some overlap. In order to provide a more general and comprehensive text on risk management we have also included [Kar96]. As a group these supplementary references provide broader coverage of the issues associated with the *management* of software engineering as compared to the set of core texts that are more closely focused on the *process* of software engineering.

3 Outline of knowledge area

The outline of the knowledge area topics is shown firstly as the draft from the jump start document, and then in a revised form (to illustrate its evolution) . We are still preparing another breakdown (as referred to in version 0.1 of the draft) based on topics rather than the life-cycle approach taken here. We envisage that both breakdowns will be useful (and complementary) in education and practice.

Original outline

The jump start document for this knowledge area identified the following topics before classifying them (using slightly different terminology) using Vincenti's [Vin90] classification scheme [Gra99]. See Appendix B or the original jump-start document for this classification.

1. Measurement
   (a) Selection of measurements
   (b) Collection of data
2. Software metric models
   (a) Model building and calibration
   (b) Model evaluation
   (c) Implementation of models
   (d) Refinement of models
3. Existing metric models
   (a) Function Point Analysis
   (b) COCOMO
4. Initiation and scope definition
   (a) Collection of requirements
   (b) Feasibility analysis
   (c) Process for the revision of requirements
5. Planning
   (a) Schedule and cost estimation
   (b) Risk assessment
   (c) Resource allocation
   (d) Task and responsibility allocation
   (e) Quality control
6. Execution

   (a) Implementation of plan
   (b) Monitor process (including reporting)
   (c) Control process
   (d) Feedback
7. Review and evaluation
   (a) Determining satisfaction of requirements
   (b) Reviewing and evaluating performance
8. Closure
   (a) Determining closure
   (b) Archival activities

As can be seen in this breakdown the topics divide into two sections. Measurement/metrics covers the first three headings, whilst management process concerns the following five.

Revised outline number one
The changes that have been made through versions 0.1 to 0.5 of the draft are as follows:

- The three major topics relating to measurement have now been "collapsed" into one so that its relative importance in the breakdown is reflected more realistically.
- We have augmented the topic area of coordination, to now consist of portfolio management, policy management, personnel management, and general management activities. Communication is a vital part of all management endeavors – irrespective of the domain of application. These are all higher-level issues that were omitted in the jump-start which focused more on individual projects (although it did make mention of portfolio management). As such we assume that this set of tasks is ongoing throughout all software processes. However, the focus is (necessarily) on the management of software development projects and related activities rather than general management.
- Feasibility analysis is now considered to be more of an on-going activity and has been added to the planning topic as well as still being part of initiation and scope definition. This differs from risk assessment and management in that feasibility analysis deals with the question "should we continue with the project?" Furthermore feasibility can be broken down into two types – technical and economic, the former dealing with "can it be done?" and the latter with "can we afford to do it?". Risk assessment and management on the other hand deal with the uncertainties that occur as part of the project with the intention that the project will continue. This can include allowances for delays, alternative paths to a goal, contingency plan development, and such like.
- Change control and configuration management are now mentioned as part of controlling the development process under the execution topic. This further stresses the interrelationship between the two knowledge areas.
- In some cases, third-level subtopics have been added to clarify which aspects of second-level subtopics are considered important here. These are not intended to be comprehensive, but rather are indicative of the important subtopics that we consider should be covered.

The current outline is very much a "life-cycle" based breakdown. Topics tend to appear in each of the two threads in the same order as their associated activities are enacted in a software development project – with the obvious exception of the "coordination" topic.

In several places specific techniques are listed. This generally indicates that the technique is suggested as being a good tutorial/case-study example of the overall concept. Other specifics could be used to replace these if desired.

The current outline is proposed as follows.

1. Measurement
   (a) The goal of a measurement program

i. Organizational objectives
ii. Software process improvement goals
iii. Determining measurement goals

(b) Selection of measurements
i. The Goal/Question/Metric approach (as an example)
ii. Other metric frameworks (such as Practical Software Measurement (PSM) and GQM++)
iii. Measurement validity

(c) Collection of data
i. Survey techniques and questionnaire design
ii. Automated and manual data extraction
iii. Costing data collection

(d) Software metric models
i. Model building, calibration and evaluation
ii. Implementation and refinement of models
iii. Existing models (examples as case-studies)

2. Coordination

(a) Portfolio management
i. Project selection
ii. Portfolio construction (risk minimization)

(b) Policy management
i. Means of policy development
ii. Policy dissemination and enforcement

(c) Personnel management
i. Hiring and staffing
ii. Directing personnel
iii. Team structures

(d) Communication

(e) General management issues

3. Initiation and scope definition

(a) Collection of requirements

(b) Feasibility analysis
i. Technical feasibility
ii. Investment analysis

(c) Process for the revision of requirements

4. Planning

(a) Schedule and cost estimation
i. Effort estimation
ii. Task dependencies
iii. Duration estimation

(b) Risk assessment
i. Critical risk analysis
ii. Techniques for modeling risk
iii. Contingency planning

(c) Resource allocation
(d) Task and responsibility allocation
(e) Quality control
(f) Feasibility analysis
    i. Investment analysis
    ii. Project abandonment policies

5. Execution
   (a) Implementation of plan
   (b) Monitor process
       i. Reporting
       ii. Variance analysis
   (c) Control process
       i. Change control
       ii. Configuration management
       iii. Scenario analysis
   (d) Feedback

6. Review and evaluation
   (a) Determining satisfaction of requirements
   (b) Reviewing and evaluating performance
       i. Personnel performance
       ii. Tool and technique evaluation
       iii. Process assessment

7. Closure
   (a) Determining closure
   (b) Archival activities
       i. Measurement database

These topics are not listed in temporal order since there are in fact two distinct processes being performed here as was mentioned with respect to the previous iteration, namely, measurement/metrics and management process. Figure 1 shows this more clearly. We have decided to treat the former as the actual activity of developing and releasing of models, and the latter as the usage of those pre-existing models. This is discussed in more detail later in the document.

4 Descriptions of topics

The overall topic is, in this breakdown, divided into the measurement/metrics and management process sub-areas.

Within the measurement/metrics topic area four main subtopics are addressed: measurement program goals, measurement selection, data collection and model development. The first three subtopics are primarily concerned with the actual theory and purpose behind measurement and address issues such as measurement scales and measure selection (such as by GQM). The collection of measures is included as an issue to be addressed here. This involves both technical issues (automated extraction) and human issues (questionnaire design, responses to measurements being taken). The fourth subtopic (model development) is concerned with the task of building models using both data and knowledge. Such models need to be evaluated (for example, by testing their performance on hold-out samples) to ensure that their levels of accuracy are both sufficient and that their limitations are known. The refinement of models, which could take place during or after projects are completed is another activity here. The implementation of metric models is more management-oriented since the use of such models has an influential effect on the *subject's* (for want of a better word) behavior.

(Note: We have continued to use the common terminology (in software engineering circles) of *software metrics* here, rather than limiting ourselves to measurement. We recognize that this could lead to some confusion with engineers familiar with

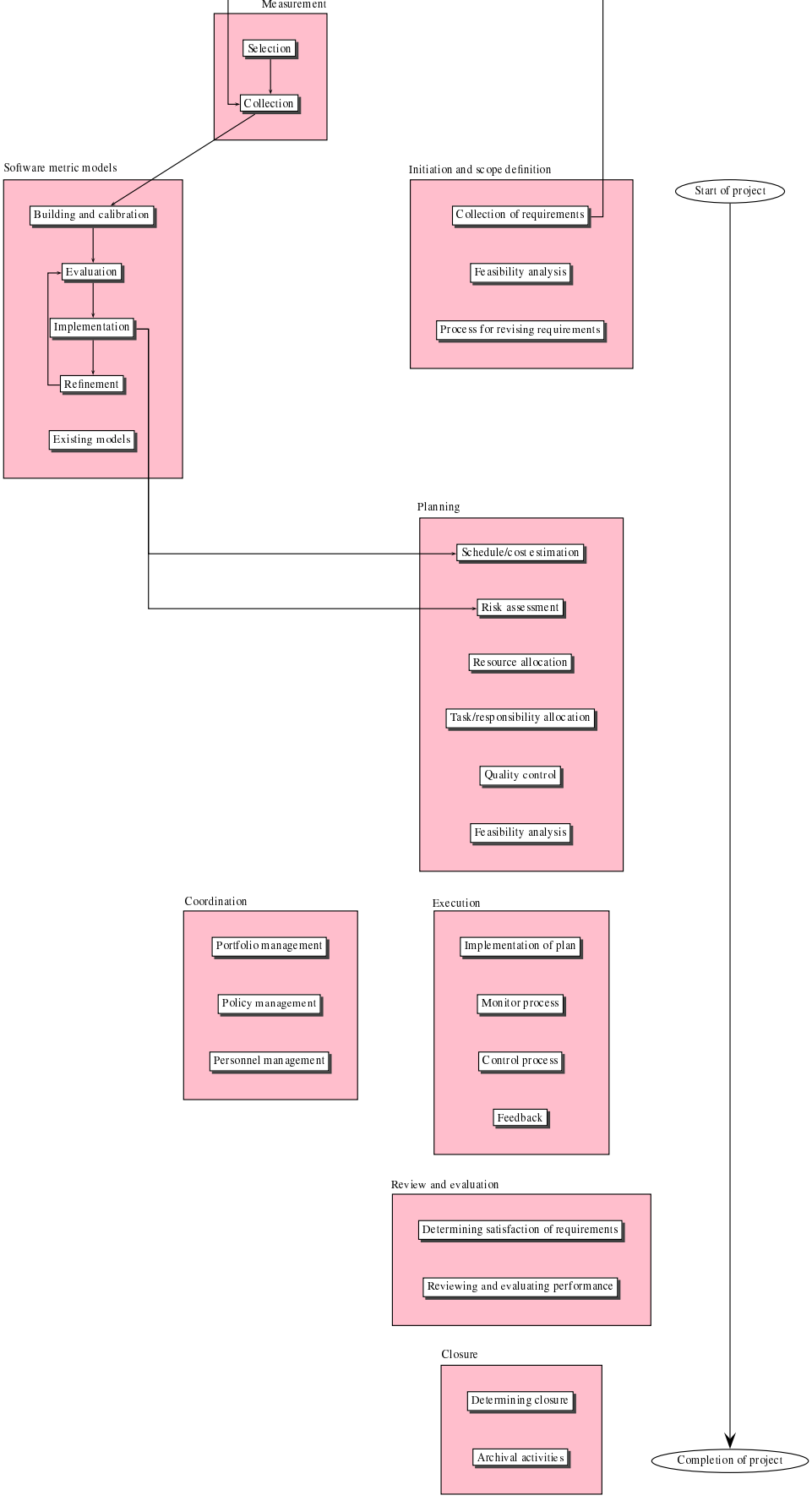

Figure 1: Measurement/metrics and management process flowchart (draft)

the empirical model-building process from another discipline, necessitating careful wording. The alternative of using more standard terminology however, whilst well intentioned, would make less obvious the connection between this work and many excellent papers and books (including Fenton and Pfleeger's seminal work). On the other hand Zuse's excellent book does include "measurement" in the title rather than "metrics". Here it seems that the best solution is to use both sets of expressions in a somewhat interchangeable manner so that practitioners are familiar with both.)

In the management process sub-area the notion of management "in the large" is considered in the coordination topic, addressing issues including project selection, the development and implementation of standards, project staffing, and team development. The remaining topics then correspond to stages in the project development life-cycle. First is the initiation and scope-definition topic which covers the management of the requirements gathering process and the specification of procedures for their revision. Feasibility analysis is included as part of this topic even though this is an ongoing activity. Here the focus is on high-level feasibility, as in "is it possible". It is during this sub-task that the communication between the two sub-areas begins. Feasibility may well be determined by reference to some formal model.

Planning is the next set of activities for a software engineering manager. This continues the interaction between the two processes. Schedule and cost estimation are perhaps the most vital tasks, although without some form of risk analysis estimates can be quite worthless. Again, feasibility analysis needs to be performed, this time from the perspective of the project. Given schedule estimates it is possible to perform task allocation. Responsibilities need to be allocated and quality control procedures implemented. The outcome of this stage would be a series of plans.

These plans are then put into action in the execution topic. The project must then be monitored for deviations and corrective actions may be taken. Change control and configuration management are important activities at this stage in the development process. The timeliness and format of reports is also important if feedback is to be successful.

The review topic involves determining that the requirements have indeed been satisfied by the system. Performance assessment, of individuals, tools, techniques and processes is necessary for performance improvement and as part of the organization's learning process.

Finally, the project needs to be closed and all useful information securely recorded. These archival activities are often neglected in both practice and education so we would like to emphasize their necessity for supporting a measurement program.

5 Justification of revised outline

The above breakdown of topics is based on a division into measurement/metrics and management process. The former refers to the actual creation of models, which can then be used as part of the latter. These activities may be performed by the same person, but they could then be seen to be "wearing different hats."

The division of topics within measurement/metrics consists of the preliminary activities (measure selection, preferably via a controlled process such as GQM, and then data collection for calibration). This then leads to model development. Here the actual models are created, implemented and refined.

Apart from the "meta-project management" issues considered in the coordination topic, the management process section follows very much a life-cycle approach, with topics covering each stage in a project from its initiation though to archival activities. Since management involves all other activities, directly or indirectly, there is considerable overlap with other knowledge areas.

6 Matrix of material versus topics

In Table 1 a '■' indicates that the topic is covered to a reasonable degree. The threshold of *reasonableness* is subjective.

Topic	[DT97][2]	[FP97]	[Kar96]	[MM98]	[Pfl98]	[Pre97b]	[Rei97]	[Som96]	[Tha97b]	[Zus97]
Archival activities						■		■		
Collection of data		■		■						■
Collection of requirements	■				■	■		■	■	
Communication	■	■	■		■	■	■	■	■	
Control process	■				■	■	■	■	■	
Determining closure						■		■		
Determining satisfaction of requirements	■				■	■		■	■	
Feasibility analysis						■		■		
Feedback						■	■	■	■	
General management issues	■	■	■		■	■	■	■	■	
Goal of measurement	■	■	■		■	■		■	■	■
Implementation of plan	■				■	■	■	■	■	
Monitor process	■				■	■	■	■	■	
Personnel management			■		■	■	■	■	■	
Policy management					■	■	■	■	■	
Portfolio management										
Process for the revision of requirements	■				■	■		■	■	
Quality control	■					■	■	■	■	
Resource allocation					■	■		■	■	
Reviewing and evaluating performance		■			■	■	■	■	■	■
Risk assessment	■		■		■	■	■	■	■	■
Schedule and cost estimation	■	■	■		■	■	■	■	■	■
Selection of measurements		■				■		■	■	■
Software metric models	■	■	■	■	■	■	■	■	■	■
Task and responsibility allocation					■	■		■		

Table 1: Topics and their references

A Assessing the breakdown using the criteria

The following subsections each discuss how the proposed draft of the KA meets the criteria given in the guide to producing this document [BDA+99]. In cases where the relevant material has yet to be written this is noted as "TO DO".

One or two breakdowns with identical topics

A single breakdown of topics is shown at present along with the draft version from the jump start document.

Soundness and reasonableness

The primary references and secondary sources were examined quite thoroughly in order to list all main topics. The division of the management process into life-cycle based topics seems both plausible and useful in terms of educational presentation.

Generally acceptable

In our view the material in this knowledge area description meets the criterion of being generally acceptable in terms of being "applicable to most projects, most of the time" and having "widespread consensus about their value and usefulness" [Dun96]. These topics are those that receive the greatest coverage in both the original texts and additional materials suggested herein.

Similarly, the Industrial Advisory Board definition of "study material of a software engineering licensing exam that a graduate would pass after completing four years of work experience" appears to be met. However, in this case the specific responsibilities of the graduate will obvious influence what areas they have the opportunity to gain experience in. Project management is often a more senior position and as such graduates with four years of experience may not have had significant experience in managing, at least with large-scale, projects.

The importance of measurement and better management practices is widely acknowledged and so its importance can only increase in the coming three to five years. Measurement has become one of the cornerstones of organizational maturity.

The only material that we feel *could* be seen to too specialized is that of software metric modeling using statistics. However, the level required here is roughly equivalent to a solid introductory course and should not be overly demanding. The understanding is not merely required for the development of models, but also for understanding existing measures and models and their limitations.

Avoid presuming specific features of applicability

The measurement/metric activities of model development, implementation and revision are seen as essential to any managed software development process. Similarly, the management activities are mostly generic to management in general.

Compatible with various schools of thought within software engineering

Excluding debate on measurement theoretic issues there is little intense debate in the measurement/metrics field.

There is nothing that appears to be controversial in the management process sub-area.

Compatible with breakdown in industry, literatures, and standards

TO DO

Inclusive as opposed to limited coverage

We have included some areas, such as statistical model building and assessment, that may be considered too specialized by some.

Inclusion of themes (quality, measurement, tools, and standards)

Quality is an important aspect of management and this is noted in Appendix E.

The second of these, measurement, is obviously included as one of the sub-areas.

Tools TO DO

Standards are listed in Appendix C.

Depth and node density

The suggested guidelines have been met here.

Meaningful topic names

Apart from the issue relating to the use of the term "metrics" (as discussed above) we feel that the names are indeed meaningful.

Breakdown based on Vincenti's classification scheme

See Appendix B for a draft breakdown from the original jump-start document. The new version, using the new references and new topic areas, is currently being prepared for the next version of this document.

Brevity of topic descriptions
The descriptions seem adequately brief and to the point.

Rationale of breakdown
We believe that the breakdowns are both rational and suitable for education. The section above on justification for the breakdown provides more information relating to this criterion.

Bloom's taxonomy
A draft version is shown in Appendix D. This will be revised and justification added in the next version of this document.

Specific reference material
Additional reference material for more specialized topics not covered adequately in the primary reference material has been added.

Proposed reference material (publicly available)
All material is publicly available.

Language of material must be English
Yes.

Maximum number of reference materials is 15
We are aiming to adhere to this limit.

Preference to IEEE or ACM copyrighted material
This is evident in the selection of reference material, especially the collections of papers.

Matrix of reference material versus topics
A draft of this is presented in 6.

Licensing exam material
TO DO

B Vincenti's categories

This is the version of the classification found in the jump-start documentation [Gra99]. A newer version with the additional references and the new topics added here is being prepared for the next version of this document.

The following Tables 2, 3, 4, 5, 6, and 7 list the main topics in the software engineering management Knowledge Area under each of Vincenti's categories [Vin90]. The relevant literature for each topic is also indicated. These categories, as noted by Vincenti, are neither mutually exclusive nor exhaustive. With the former point in mind, many topics appear in more than one table. The area of software engineering management does not, in the authors' opinion, fit easily into Vincenti's categories.

In these tables a ■ indicates that the topic is covered to a reasonable degree. The threshold of *reasonableness* is of course subjective.

Fundamental Design Concepts	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
Project planning	■		■	■	■
Project management	■		■	■	■
Quality assurance	■			■	■
Characteristics of projects	■	■	■	■	■
Risk assessment and management	■		■	■	■
Software metric models	■	■	■	■	■

Table 2: Vincenti's categories – fundamental design concepts

C Standards

Table 8 is a possibly incomplete list of standards that are listed and/or discussed within the reference publications. This is the same as was provided in the jump-start document [Gra99]. The next step will be to reference the newer texts and map topics to standards.

D Bloom's taxonomy

Criteria and Specifications	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
Project proposal				■	■
System requirements	■		■	■	■
System characteristics		■	■	■	■
User characteristics			■	■	

Table 3: Vincenti's categories – criteria and specifications

Theoretical Tools	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
COCOMO	■	■	■	■	■
Function Point Analysis	■	■		■	■
Goal/Question/Metric Framework	■	■			■
Project schedule	■	■	■	■	■
Software metric models	■	■	■	■	■
Statistical techniques for model building		■			
Statistical techniques for analyzing experimental data		■			

Table 4: Vincenti's categories – theoretical tools

Quantitative Data	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
Complexity measurements	■	■		■	■
Cost estimates	■	■	■	■	■
Developer productivity	■	■	■	■	■
Effort estimates	■	■	■	■	■
Metric model assessment		■	■		
Quality measurements	■	■		■	■
Reliability measurements	■	■	■	■	■
Size measurements	■	■	■	■	■
Software standards	■		■	■	■
System performance requirements	■		■	■	■

Table 5: Vincenti's categories – quantitative data

Practical Considerations	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
Expert opinion for metrics estimation	■	■	■		■
Process maturity	■	■	■	■	■

Table 6: Vincenti's categories – practical considerations

Design Instrumentalities	[DT97]	[FP97]	[Pfl98]	[Pre97b]	[Som96]
Project management	■	■	■	■	■
Software metric model building	■	■	■	■	■

Table 7: Vincenti's categories – design instrumentalities

Table 9 shows the level of mastery that we feel a "graduate plus four years experience" should possess for each topic. This is still a very rough draft, but indicates our approach to this task.

E Related knowledge areas

Some parts of these topic areas overlap with other Knowledge Areas, most notably software construction (which must be compatible vis-á-vis software engineering management), software testing (which should involve software metrics in some capacity), and software quality analysis (which should also make use of software metrics and management-initiated reviews).

F Related disciplines

In terms of related disciplines (which are defined in terms of the authors' own experience) the most important are as follows. Again, this is an updated version of material from the jump-start document [Gra99].

Computer science
Without some understanding of the fundamental concepts underlying the activity of software development some areas of software engineering management would be inaccessible or prohibitively buried in computer science terminology and concepts. For example, many software metric models require some understanding of data structures as independent variables. Similarly, feasibility analysis requires a basic understanding of computer science in terms of hardware performance, storage devices, etcetera. The history of computing is also useful when projects require some judgements on likely changes to technologies in the industry.

Project management
This is obviously a major component of the software engineering management Knowledge Area.

Management
Many principles in managing software engineering projects are common to generic management processes. These include personnel issues, planning techniques and tools, budgeting, and project selection methods (both under limited resources and between mutually exclusive alternatives).

Cognitive science
This discipline is mainly invoked here as part of quality assurance in terms of usability. Other aspects making use of this discipline include motivation, task allocation, and training issues.

Management sciences
Many management science topics can be readily used without adaptation in a software engineering management context. This especially includes the use of modeling techniques.

REFERENCES

[BB97] David Budgen and Pearl Brereton. *A software engineering bibliography*, pages 503–508. In Dorfman and Thayer [DT97], 1997.

[BDA+99] Pierre Bourque, Robert Dupuis, Alain Abran, James W. Moore, and Leonard Tripp. Knowledge area description specifications for the stone man version of the guide to the software engineering body of knowledge version 0.2. Submitted to the Industrial Advisory Board for SWEBOK, March 1999.

[Boe97a] Barry W. Boehm. *Industrial software metrics top 10 list*, page 494. In Dorfman and Thayer [DT97], 1997.

[Boe97b] Barry W. Boehm. *A spiral model of software development and enhancement*, pages 415–426. In Dorfman and Thayer [DT97], 1997.

[Bro97a] Frederick P. Brooks. *The mythical man-month*, pages 350–357. In Dorfman and Thayer [DT97], 1997.

[Bro97b] Frederick P. Brooks. *No silver bullet: essence and accidents of software engineering*, pages 13–22. In Dorfman and Thayer [DT97], 1997.

[Car97] J. M. Carey. *Prototyping: alternative systems development methodology*, pages 461–468. In Dorfman and Thayer [DT97], 1997.

[Com97] Edward R. Comer. *Alternative software life cycle models*, pages 404–414. In Dorfman and Thayer [DT97], 1997.

[DeM97] Tom DeMarco. *Why does software cost so much?*, pages 372–373. In Dorfman and Thayer [DT97], 1997.

[DT97] Merlin Dorfman and Richard H. Thayer, editors. *Software engineering*. IEEE Computer Society, 1997.

[Dun96] W.R. Duncan. A guide to the project management body of knowledge. Project Management Institute, 1996.

[Fai97] Richard Fairly. *Risk management for software development*, pages 387–400. In Dorfman and Thayer [DT97], 1997.

[FP97] Norman E. Fenton and Shari Lawrence Pfleeger. *Software metrics: a rigorous & practical approach.* PWS Publishing Company, second edition, 1997.

[Gib97] Wayt Gibbs. *Software's chronic crisis*, pages 4–12. In Dorfman and Thayer [DT97], 1997.

[Gra99] Andrew R. Gray. Swebok knowledge area jump-start document for software engineering management. Unpublished report for SWEBOK, February 1999.

[Hee97] F. J. Heemstra. *Software cost estimation*, pages 374–386. In Dorfman and Thayer [DT97], 1997.

[Hur97] Patricia W. Hurst. *Software quality assurance: a survey of an emerging view*, pages 308–319. In Dorfman and Thayer [DT97], 1997.

[Kar96] Dale Walter Karolak. *Software engineering risk management.* IEEE Computer Society, 1996.

[MM98] David S. Moore and George P. McCabe. *Introduction to the practice of statistics.* W H Freeman & Co., third edition, 1998.

[Moo98] James W. Moore. *Software engineering standards: a user's road map.* IEEE Computer Society, 1998.

[NB97] Ronald E. Nusenoff and Dennis C. Bunde. *A guidebook and a spreadsheet tool for a corporate metrics program*, pages 483–493. In Dorfman and Thayer [DT97], 1997.

[Pau97] Mark C. Paulk. *The Capability Maturity Model for software*, pages 427–438. In Dorfman and Thayer [DT97], 1997.

[Pfl98] Shari Lawrence Pfleeger. *Software engineering: theory and practice.* Prentice Hall, 1998.

[Pre97a] Roger S. Pressman. *Software engineering*, pages 57–74. In Dorfman and Thayer [DT97], 1997.

[Pre97b] Roger S. Pressman. *Software engineering: a practitioner's approach.* McGraw-Hill, fourth edition, 1997.

[Rei97] Donald J. Reifer, editor. *Software management.* IEEE Computer Society, fifth edition, 1997.

[Som96] Ian Sommerville. *Software engineering.* Addison-Wesley, 1996.

[Tha97a] Richard H. Thayer. *Software engineering project management*, pages 358–371. In Dorfman and Thayer [DT97], 1997.

[Tha97b] Richard H. Thayer, editor. *Software engineering project management.* IEEE Computer Society, 1997.

[Tha97c] Richard H. Thayer. *Software engineering standards*, pages 509–523. In Dorfman and Thayer [DT97], 1997.

[Vin90] W.G. Vincenti. *What engineers know and how they know it - analytical studies from aeronautical history.* John Hopkins, 1990.

[Zus97] Horst Zuse. *A framework of software measurement.* Walter de Gruyter, 1997.

Standard	[DT97][3]	[FP97]	[Moo98]	[Pfl98]	[Pre97b]	[Som96]
AFNOR						
Z67-101-1FD	■					
ASTM						
E 622-94	■					
E 792-87	■					
E E1113-86	■					
DOD						
AFSCP 800-43	■					
AFSCP 800-45	■					
MIL STD 1521B	■					
EIA						
CRB 1-89	■					
DMG-1-86	■					
DMG-2-89	■					
ESA						
PSS-05-08	■					
GER MOD						
PROSIS	■					
IEEE Std.						
982.1	■		■			
982.2	■		■			
1044			■			
1044.1			■			
1045	■		■			
1058.1	■		■			
1061	■		■			
1209	■					
1220	■					
ISO						
9000 series		■		■	■	■
9000-3		■		■	■	■
9001		■		■	■	■
9126		■		■		
ISO/IEC DIS						
14143.1			■			
14756			■			
JPL						
D-4011	■					
NATO						
NAT-PRC-1	■					
NAT-STAN-7	■					
PMI						
1996			■			

One of the papers in [DT97], namely [Tha97c] is in fact a comprehensive list of software engineering standards. All standards indicated as being mentioned in this volume came from this one paper in §4.14 (Management Standards).

Table 8: Standards relevant to software engineering management

Topic	Level
Archival activities	Application
Collection of data	Analysis
Collection of requirements	Analysis
Communication	Synthesis
Control process	Evaluation
Determining closure	Application
Determining satisfaction of requirements	Analysis
Feasibility analysis	Synthesis
Feedback	Synthesis
General management issues	Synthesis
Goal of measurement	Analysis
Implementation of plan	Synthesis
Monitor process	Analysis
Personnel management	Synthesis
Policy management	Synthesis
Portfolio management	Analysis
Process for the revision of requirements	Analysis
Quality control	Evaluation
Resource allocation	Application
Reviewing and evaluating performance	Synthesis
Risk assessment	Synthesis
Schedule and cost estimation	Evaluation
Selection of measurements	Analysis
Software metric models	Analysis
Task and responsibility allocation	Analysis

Table 9: Bloom's taxonomy and mastery levels for the software engineering management topics