



Software Engineering Standards and Certification

Leonard L. Tripp

1999 President IEEE Computer Society

14 September 2001



Presentation Outline

- Background
- IEEE Certification Program
- Guide to Body of Knowledge Project
- Software Engineering Standards
- Concluding remarks

Terminology

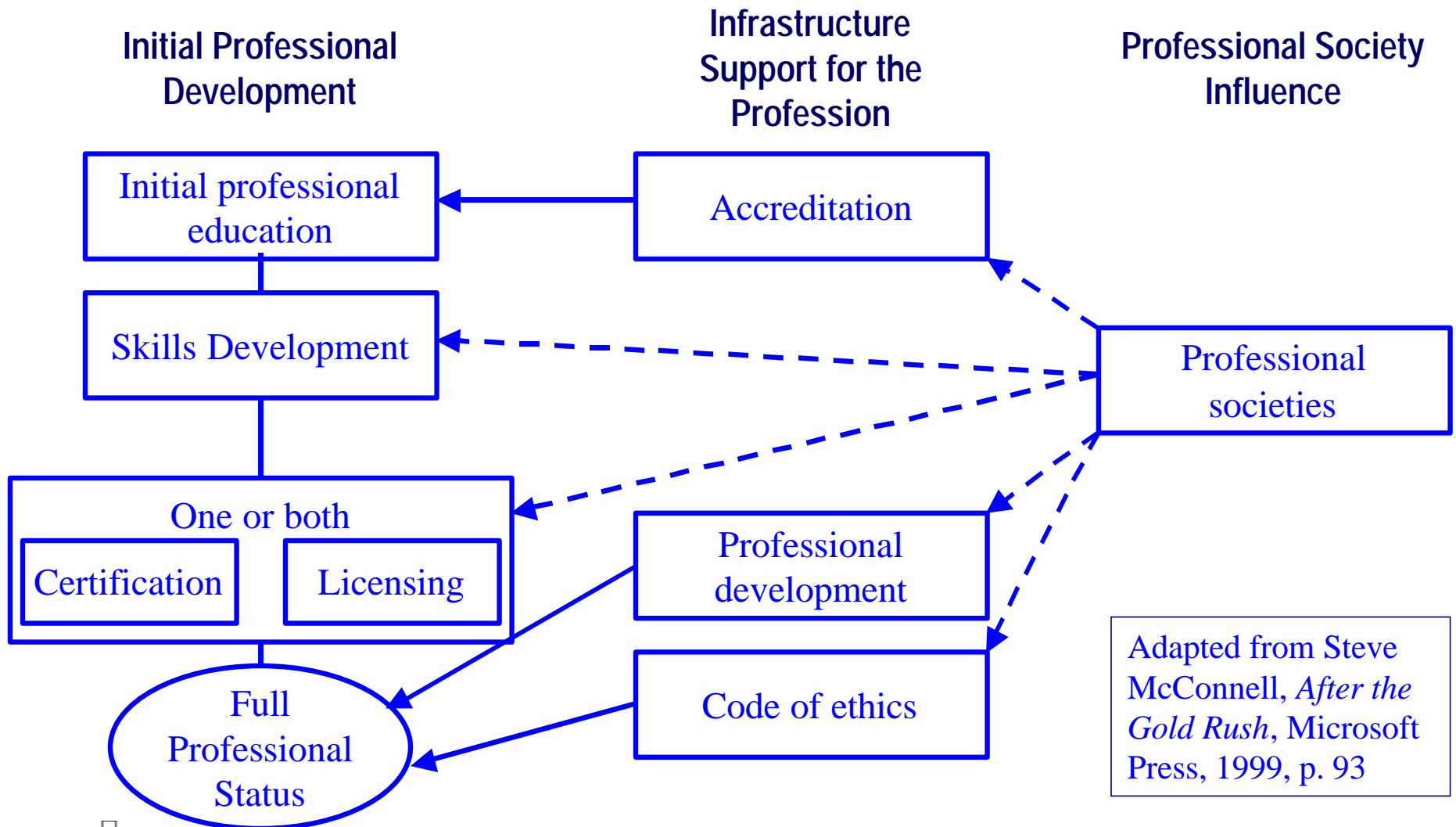
Engineering: The application of science and mathematics by which properties of matter and the sources of energy are made useful to people. (Merriam-Webster's New Collegiate Dictionary, 10th Edition)

Terminology

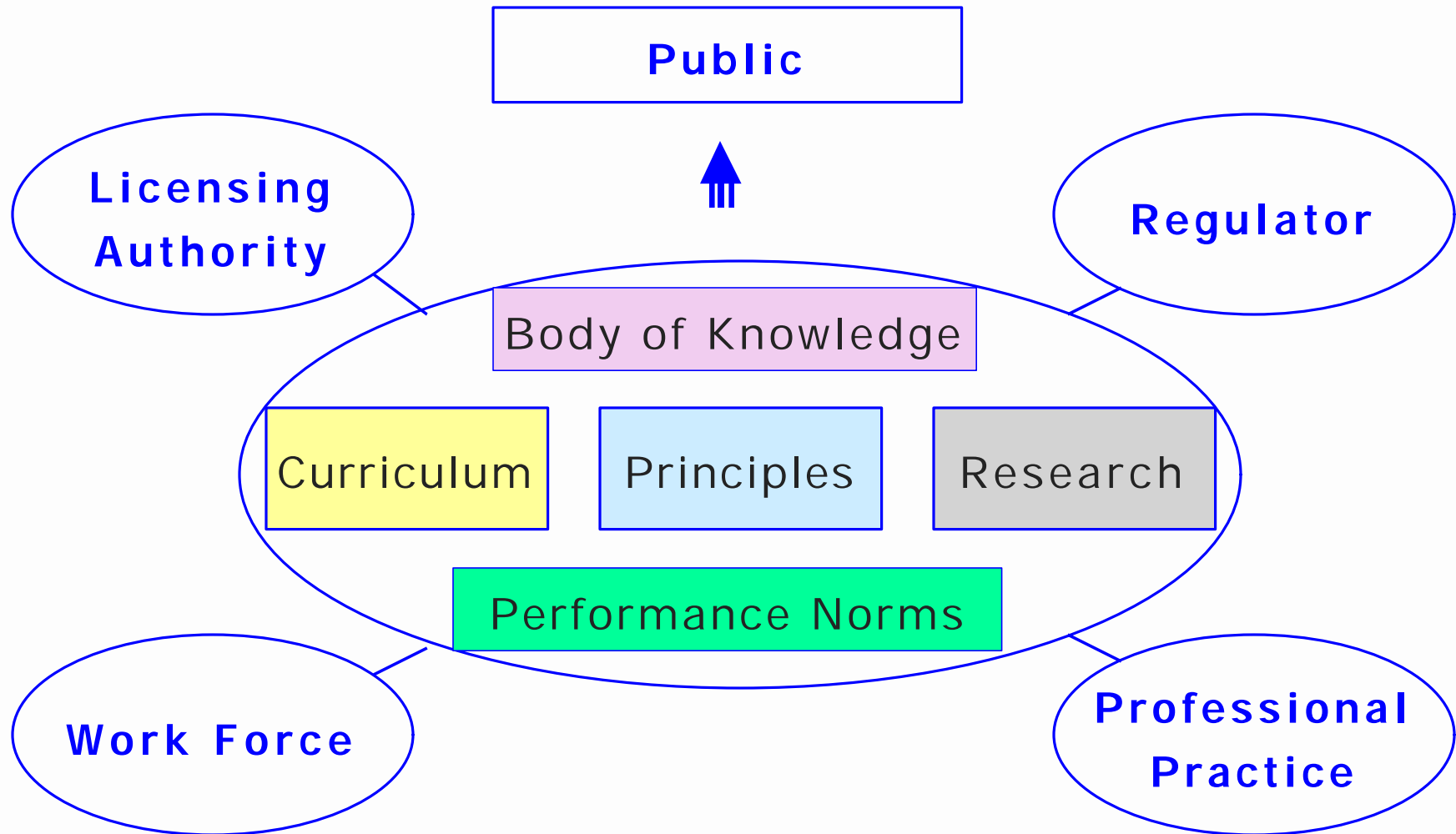
Software engineering. (1) The application of a systematic disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

(IEEE Std 610.12)

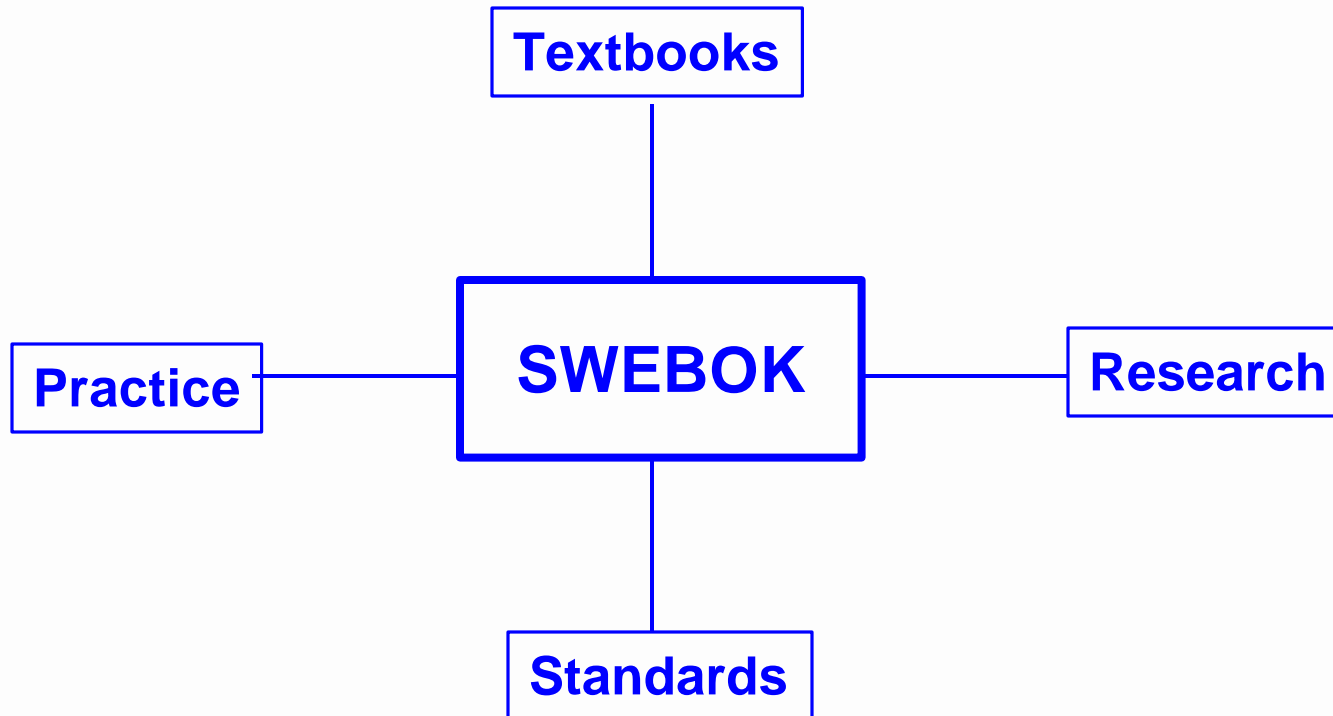
Professional Development



Elements of a Profession



Body of Knowledge Sources



IEEE Software Engineering Certification Program - Requirements

- **Education**
Baccalaureate degree
- **Experience**
9000 hours in 6 of 11 knowledge areas
- **Proof of Professionalism**
Participant in professional society or recommended by 2
Computer Society members
- **Examination**
Passing score on 3 ½ hour examination
- **Assumed knowledge**
Competent in computing fundamentals
- **Maintain professional currency**

Certification Program - Project Results

- **Examination**

- 420 Initial test item pool
- 300 Test item pool additions

- **Candidate Preparation**

- Examination Preparation Guide
- 3 day Training Workshop
- Sample Examination

- **Re-certification**

- Catalog of professional development activities

- **Policies and Procedures**

- Examination Development
- Examination Administration
- Training Development

Cert Program - Development Milestones

Aug 1999	Job Analysis
Sep 1999	Job Analysis Survey
Jan 2000	Test Specification
Apr 2000	Item Writing Workshop
Jul 2000	Item Review Workshop
Nov 2000	Exam Review Workshop
Jun 2001	Exam Beta Test
Aug 2001	Passing score determined
Nov 2001	IEEE approval to offer exam

Guide to Body of Knowledge Project

- **Project Objectives**
- **Intended Audience**
- **Underlying Principles of the Project**
- **Project Approach**
- **Initial Set of Knowledge Areas**

Project Objectives

- **Characterize the contents of the Body of Knowledge**
- **Provide a topical access structure**
- **Promote a consistent view of software engineering worldwide**
- **Clarify the place of, and set the boundary with respect to other disciplines**
- **Provide a foundation for education and licensing**

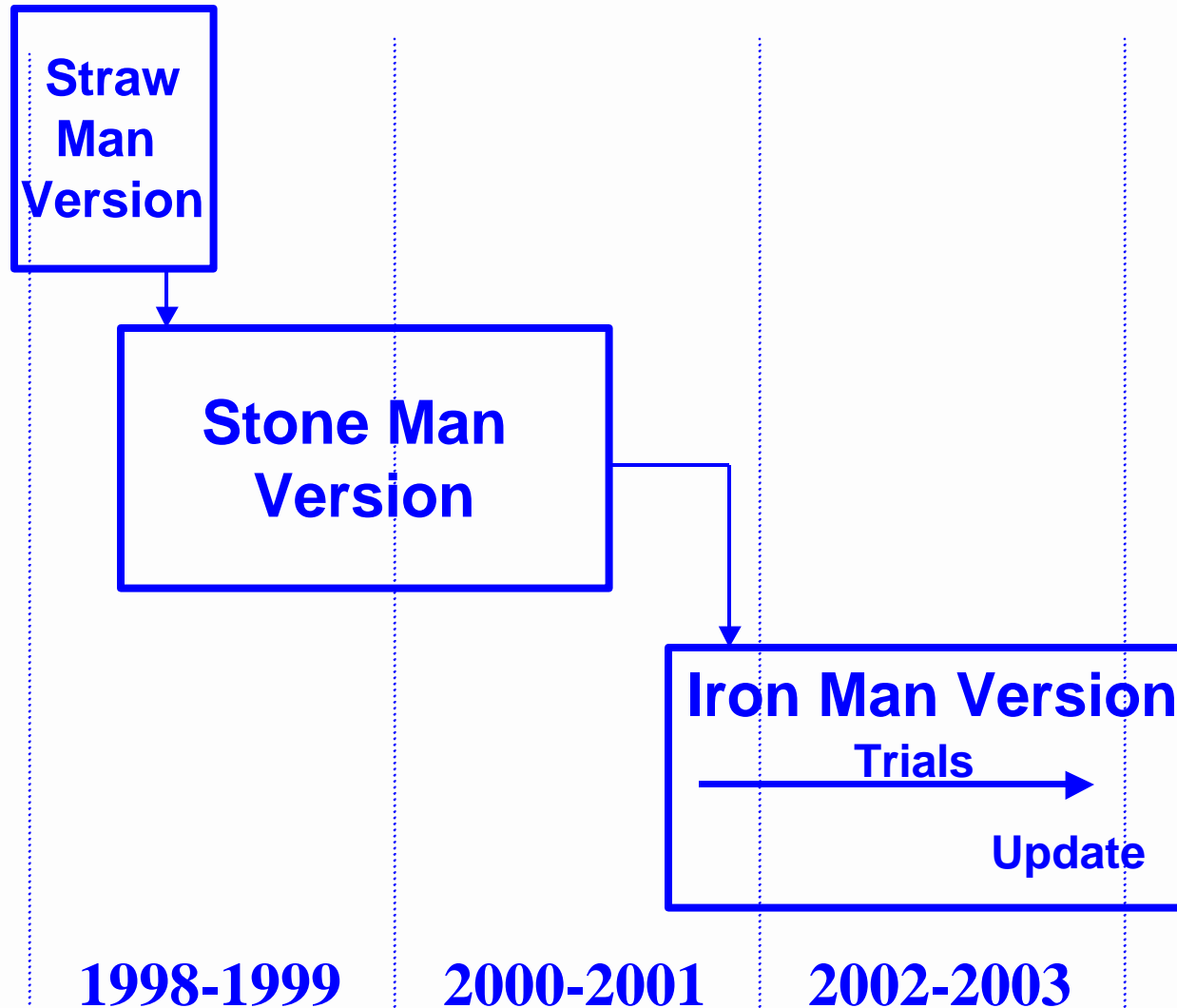
Intended Audience

- **Public and private organizations**
- **Practicing software engineers**
- **Makers of public policy**
- **Professional societies**
- **Software engineering students**
- **Educators and trainers**

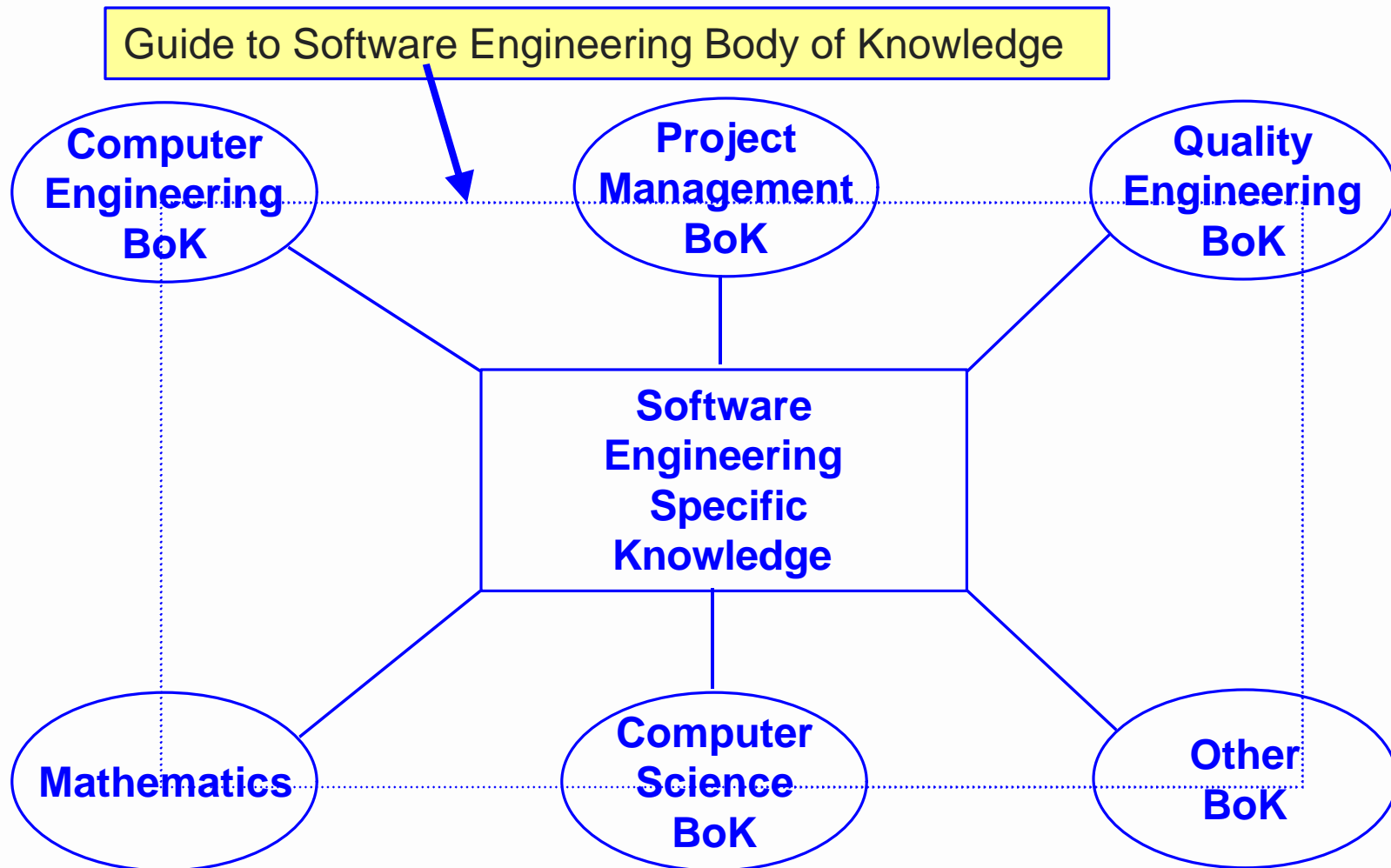
Project Principles

- *Transparency*: the development process is itself published and fully documented
- *Consensus-building*: the development process is designed to build, over time, consensus in industry, among professional societies and standards-setting bodies and in academia

Project Approach



Project Context



SWEBOK -- What It Is

- **Software Engineering Body of Knowledge (SWEBOK)**
 - **“An inclusive term that describes the sum knowledge within the profession of software engineering**
 - **Guide to the Software Engineering Body of Knowledge,**
published by the IEEE Computer Society, identifies and describes “that subset of software body of knowledge which is generally accepted . . . the knowledge and practices described are applicable to most applications most of the time, and that there is widespread consensus about their value and usefulness.”

SWEBOK -- How It is Used

- **The SWEBOK --**
 - **Provides a common taxonomy or lexicon**
 - **Establishes the boundary of, software engineering with respect to other disciplines**
 - **Provides a consistent structure for accreditation of degree-granting educational programs in software engineering**
 - **Provides a foundation for individual certification and licensing material**

SWEBOK - What It Defines

- **Software:** Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.
(Source: IEEE Std 610.12)
- **Software Engineering:**
 - (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
 - (2) The study of approaches as in (1).
(Source: IEEE Std 610.12)

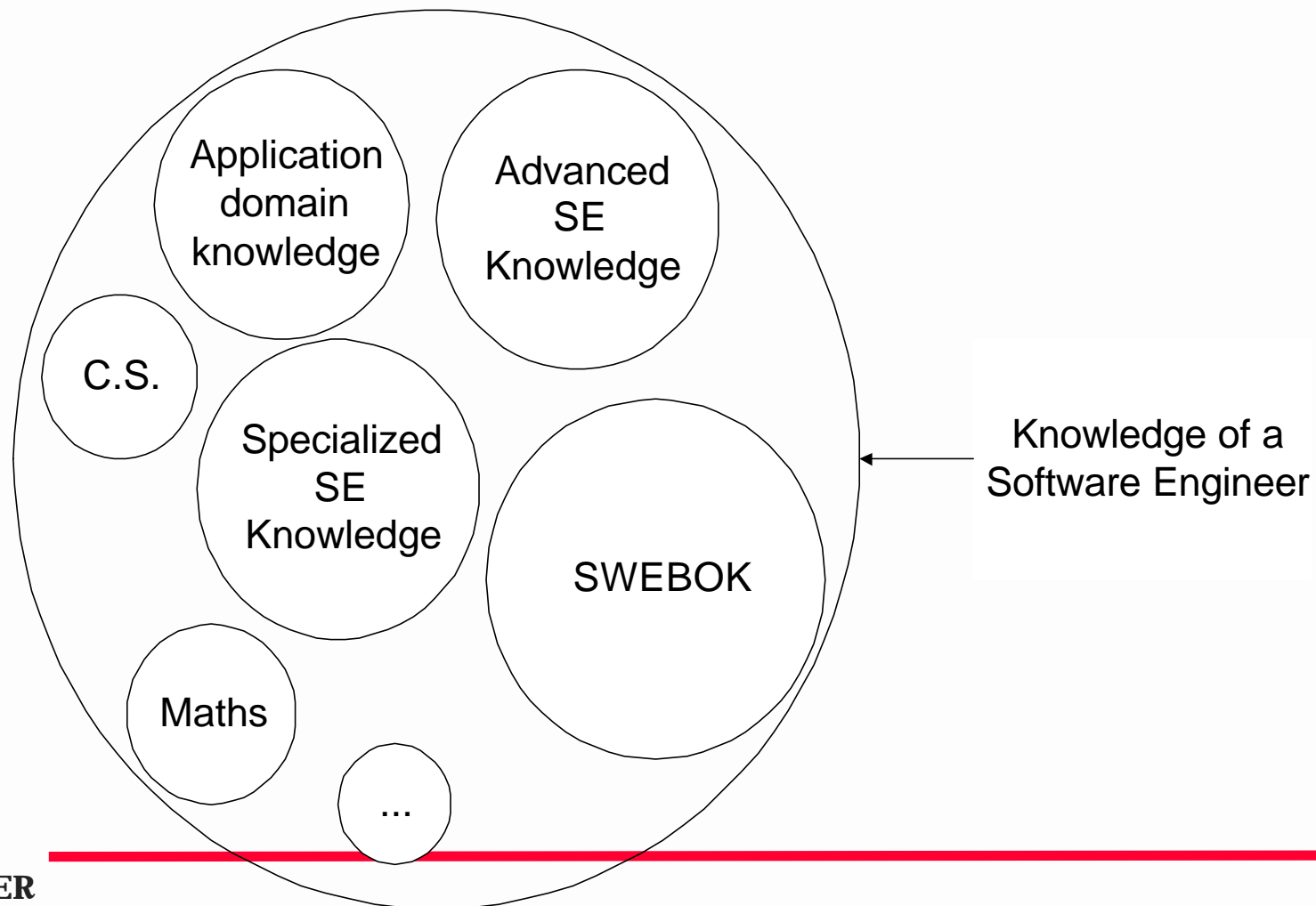
SWEBOK – Development Summary

- **Project sponsor – IEEE Computer Society**
- **Project duration – 1998 – May 2001**
- **Editorial team – Alain Abran, James Moore, Robert Dupuis, Pierre Bourque**
- **Corporate sponsors – ACM, Boeing, CCPE, Construx Software, MITRE, NIST, National Research Council Canada, Rational, Raytheon, SAP Labs**
- **Project manager – University of Quebec at Montreal**
- **Review process – three review cycles with over 500 participants from 41 countries produced nearly 10, 000 comments. All comments and their associated dispositions are available at the following URL: www.swebok.org**

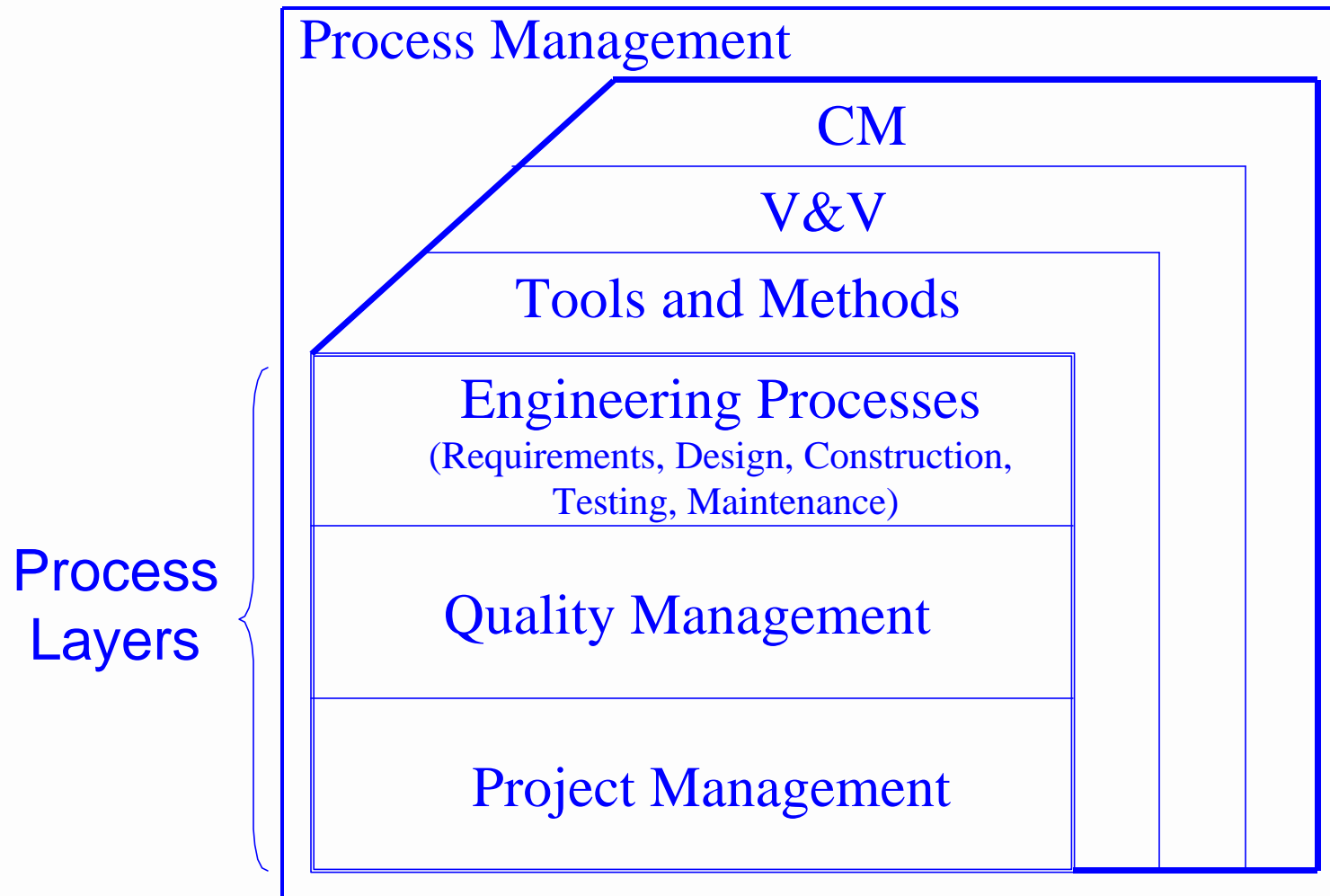
SWEBOK - Content Organization

- A chapter corresponds to a knowledge area
- The 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.
- The Guide treats the selected topics in a manner compatible with major schools of thought and with breakdowns generally found in industry and in software engineering literature and standards.
- The extent of each topic's description is only that needed to understand the generally accepted nature of the topics and for the reader to successfully find reference material.
- Materials were selected because they are written in English, readily available, easily readable, and—taken as a whole—provide coverage of the described topics.

SWEBOK - Its Place within Software Engineer's Knowledge



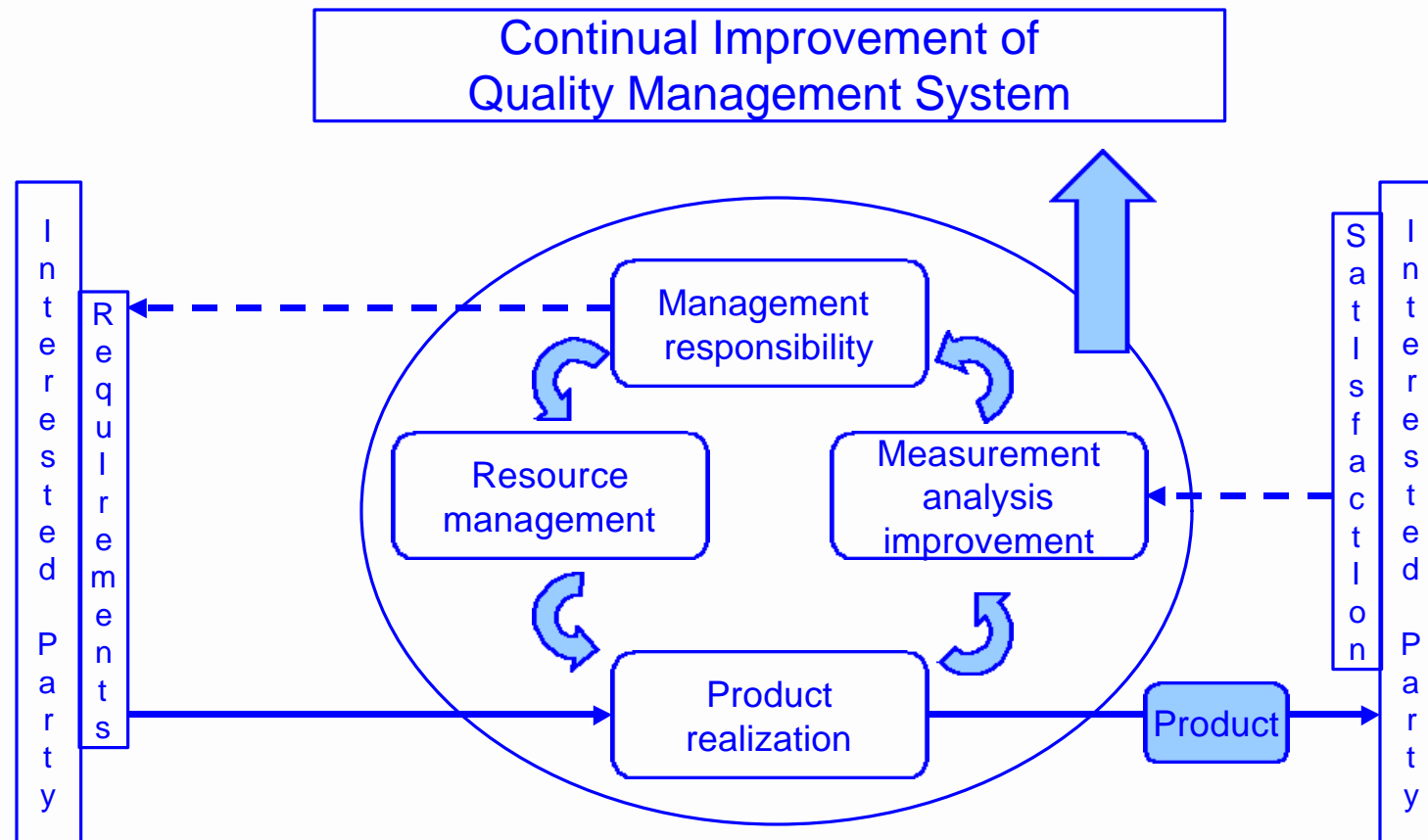
SWEBOK - Process Model



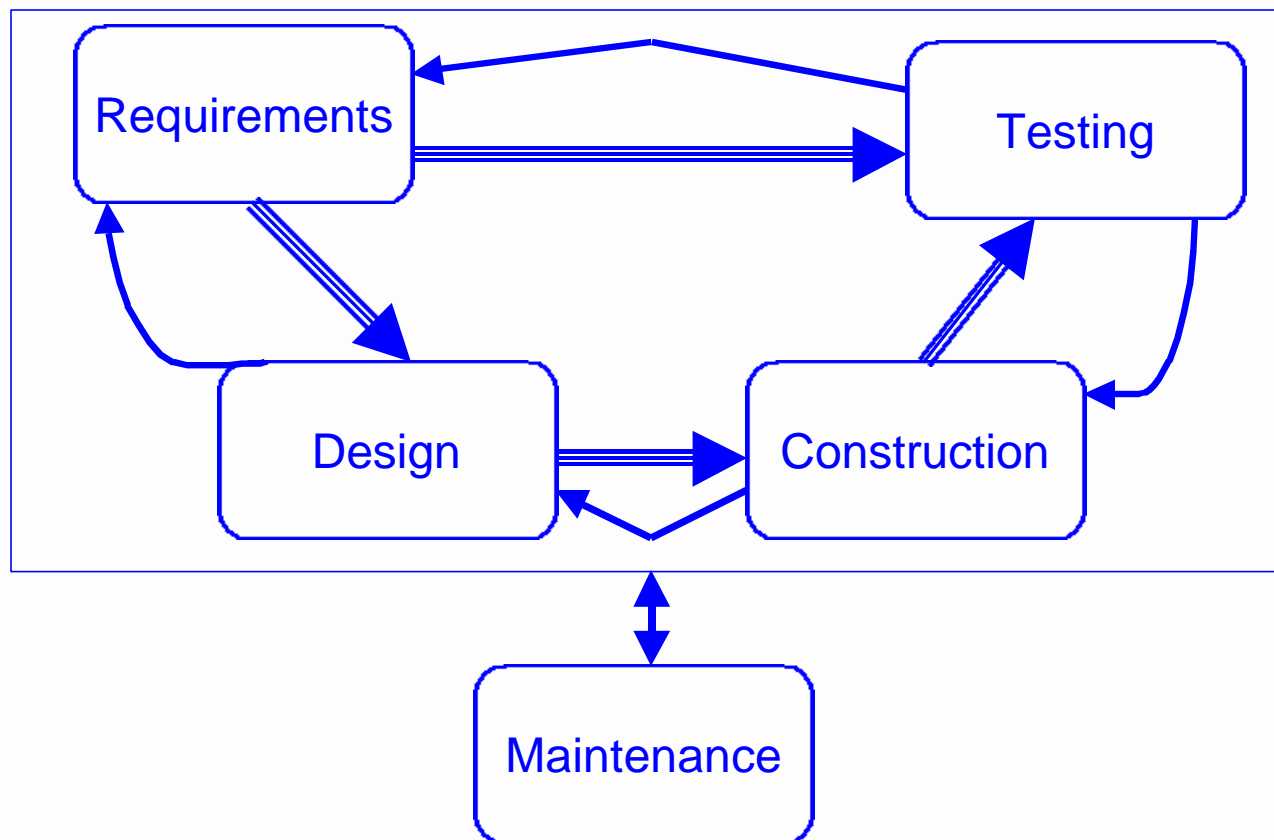
SWEBOK - Project Management Layer

Process Group	Area of Knowledge									
		Integration	Scope	Time	Cost	Quality	HR	Communication	Risk	Procurement
	Initiating		Initiation							
	Planning	Project Plan Development	Scope Planning	Activity Definition	Resource Planning	Quality Planning	Organizational Planning	Communication Planning	Risk Identification	Procurement Planning
			Scope Definition	Activity Sequencing	Cost Estimating		Staff Acquisition		Risk Quantification	Solicitation Planning
				Activity Duration Estimating	Cost Budgeting				Risk Response Development	
				Schedule Development						
	Executing	Project Plan Execution	Scope Verification			Quality Assurance	Team Development	Information Distribution		Solicitation
Source Selection										
									Contract Administration	
Controlling	Overall Change Control	Scope Change Control	Schedule Control	Cost Control	Quality Control		Performance Reporting	Risk Response Control		
Closing							Administrative Closure		Contract Close-Out	

SWEBOK - Quality Management Layer



SWEBOK - Engineering Process Layer



SWEBOK -- How It Is Organized

- **10 Knowledge Areas (KA)**
 - Software Requirements
 - Software Design
 - Software Construction
 - Software Testing
 - Software Maintenance
 - Software Configuration Management
 - Software Quality
 - Software Engineering Tools & Methods
 - Software Engineering Process
 - Software Engineering Management
- **Interfaces to Related Disciplines**
 - Computer Science
 - Project Management
 - Computer Engineering
 - Systems Engineering
 - Management and Management Science
 - Cognitive Sciences and Human Factors

SWEBOK -- Requirements KA

- **Requirements engineering process**
- **Requirements elicitation**
- **Requirements analysis**
- **Software requirements specification**
- **Requirements validation**
- **Requirements management**

SWEBOK - Design KA

- **Software design concepts and notions**
- **Software design issues**
- **Structure and architecture**
- **Software design quality analysis and evaluation**
- **Software design notations**
- **Software design strategies and methods**

SWEBOK - Construction KA

- **Software construction principles**
 - reduction of complexity
 - anticipation of diversity
 - structuring for validation
 - use of external standards
- **Software construction styles/methods**
 - linguistic
 - formal
 - visual

SWEBOK - Testing KA

- **Testing concepts**
- **Test levels**
- **Test techniques**
- **Test-related measures**
- **Test process management**

SWEBOK - Maintenance KA

- **Maintenance concepts**
- **Software maintenance process**
- **Software maintenance issues**
 - **technical**
 - **management**
 - **cost and estimating**
- **Techniques for maintenance**
 - **program comprehension**
 - **re-engineering**
 - **reverse engineering**
 - **impact analysis**

SWEBOK - Configuration Management KA

- **Management of SCM process**
- **Software configuration identification**
- **Software configuration control**
- **Software configuration status accounting**
- **Software configuration auditing**
- **Software release management**

SWEBOK - Software Engineering Management KA

- **Organizational management**
- **Process/project management**
- **Software engineering measurement**

SWEBOK - Software Engineering Process KA

- **Process concepts**
- **Process infrastructure**
- **Measurements specific to software engineering process**
- **Process definition**
- **Qualitative process analysis**
- **Process implementation and change**

SWEBOK - Tools and Methods KA

- **Software tools**
 - (tools for other 9 KAs)
 - infrastructure for tools
 - miscellaneous tools
- **Software methods**
 - heuristic methods
 - formal methods
 - prototyping methods
 - miscellaneous methods

SWEBOK - Software Quality KA

- **Software quality concepts**
- **Purpose and planning of SQA and V&V**
- **SQA and V&V activities and techniques**
- **Measurement applied to SQA & V&V**

Software Engineering Standards

- **Characteristics**
- **Importance of Standards**
- **IEEE SESC**
- **Strategic Model**
- **1999 IEEE SESC Collection**
- **ISO/IEC 12207**
- **Standards in Five Years**

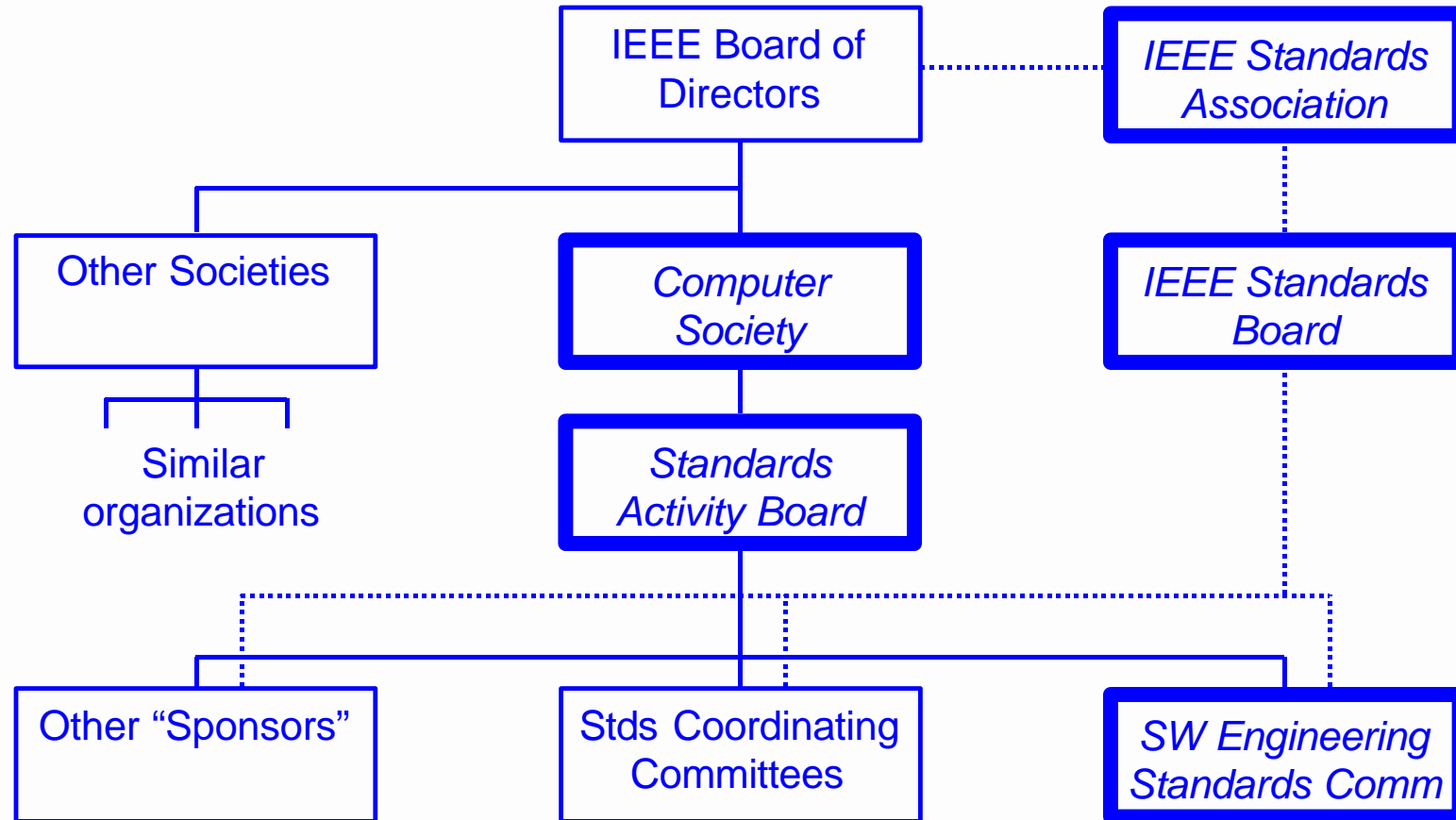
Characteristics

- Concern the *responsible practice* of software engineering
- Usually deal with *process* rather than product ...
- ... although they sometimes deal with generic product characteristics or supporting resources
- Deal with subjects like *Configuration Management, Quality Assurance, Verification, and Validation*

Importance of Standards

- They consolidate existing technology into a firm basis for introducing newer technology
- They increase professional discipline
- They protect the business
- They protect the buyer
- They improve the product

IEEE SESC



IEEE Software Engineering Std

- Scope

Standardization of processes, products, resources, notations, methods, nomenclatures, and techniques, for the engineering of software and systems dependent on software.

- Membership

1500 software practitioners in over 30 countries

- Progress

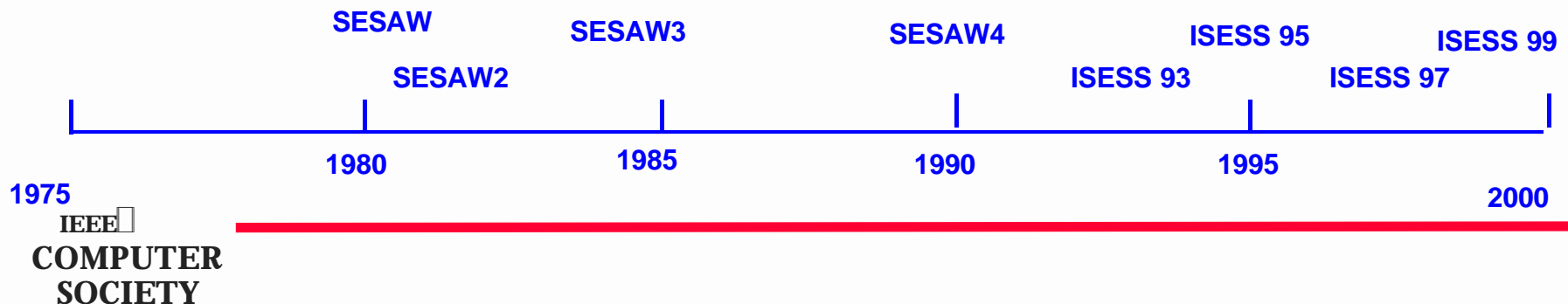
Year	Number of Stds	Number of Pages
1984	5	139
1987	11	293
1993	19	690
1994	27	990
1997	38	1580
1999	44	2400

SESC History

729 Terminology	830 Rqmts Spec
730 QA	982 Reliable SW
828 CM	
829 Test	

1008 Unit Test	1058 PM
1012 V&V	1061 SW Quality
1016 Design Rep	1074 LC
1028 SW Reviews	1219 SW Maintenance

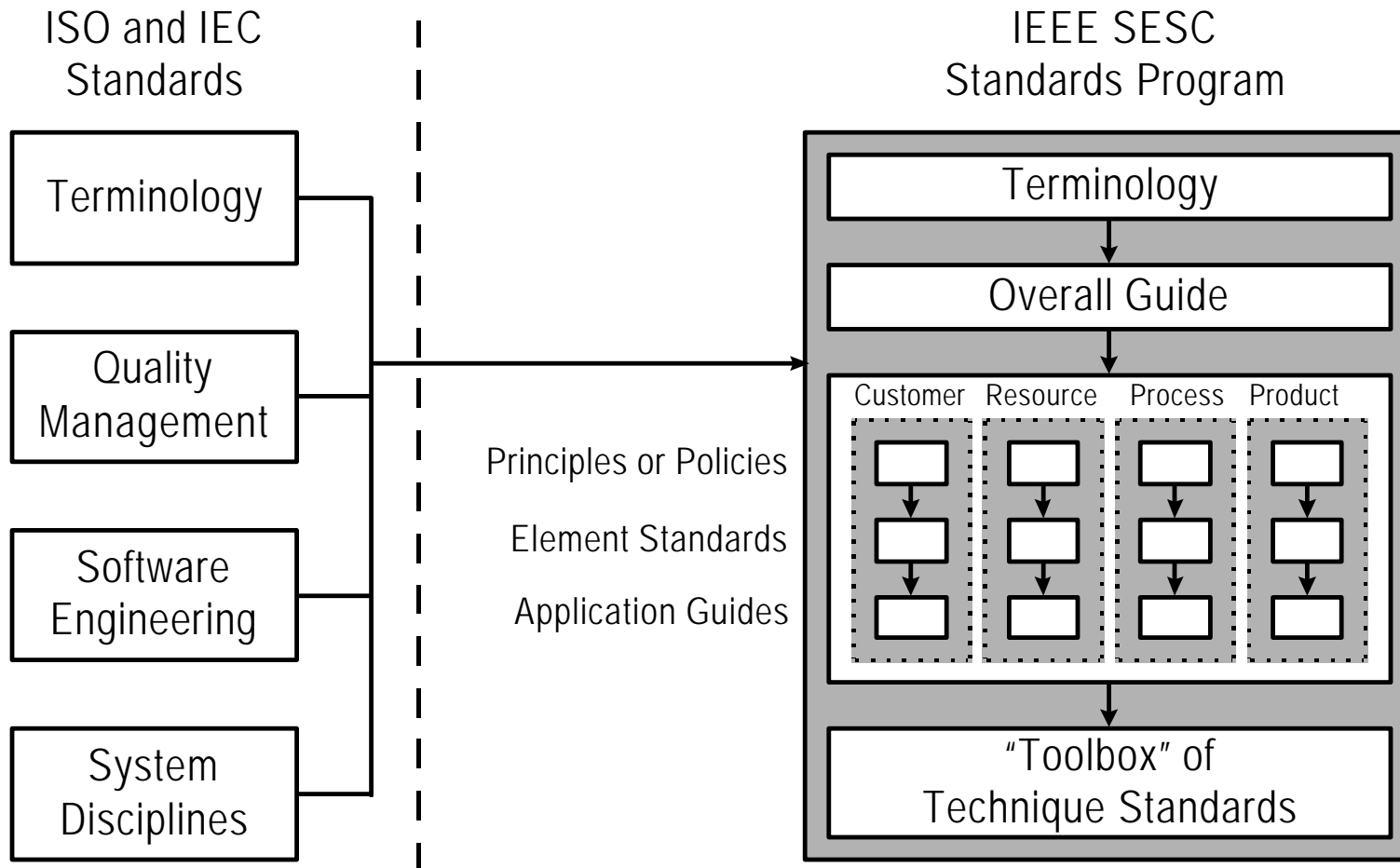
1044 SW Anomalies	1233 System Spec
1045 Prod Measures	1348 CASE Tool
1075 Tool Intercon	1420 Software Reuse
1220 Sys Eng Proc	12207.0 SW LC



Features of 1999 Collection

- Four volumes including 2400 pages
- Organized around standards model
- Alignment with international standards
- Harmonized process and data standards
- Established relationships with related disciplines

Strategic Model for SESC Program



Overall SESC Collection

*Adapted from
[Moore97]*

Terminology

610.12
IEEE glossary

729
SESC glossary

1002
Taxonomy of
standards

Overall Guide

[Moore97]

Principles

Element Standards

See figure
below for
Customer
standards

See figure
below for
Process
standards

See figure
below for
Product
standards

See figure
below for
Resource
standards

Application Guides and Supplements

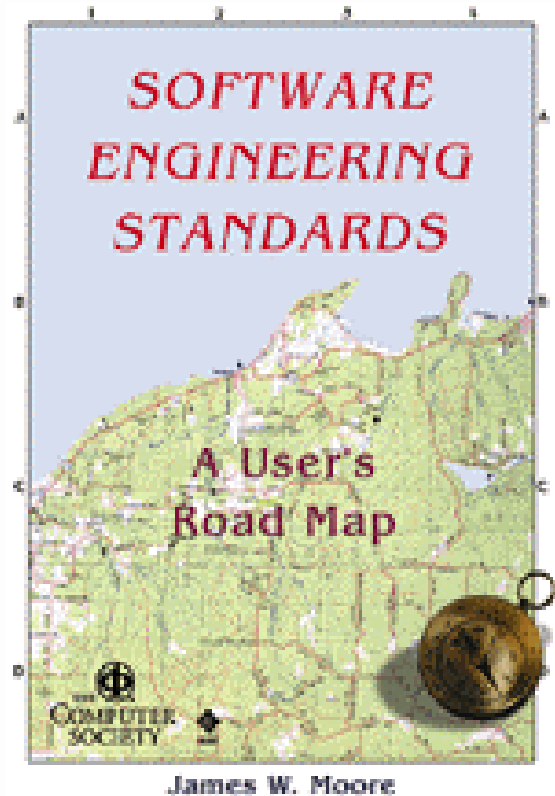
“Toolbox” of Techniques

1044
Classification of
anomalies

1044.1
Guide to
1044

Overall Guide to Collection

- Rather than developing a consensus document, IEEE SESC “commissioned” the writing of an overall guide to the collection.
- Relevant standards are “sliced and diced” by:
 - Context
 - Object
 - Normative intent
 - Provider and subject
- Relationships are explained, even between standards from different providers.
- Published by IEEE Computer Society Press. 328 pages. 6" x 9" Softcover. ISBN 0-8186-8008-3. \$40.00 List



SESC Resource Stack

*Adapted from
[Moore97]*

Principles

Element Standards

Application Guides and Supplements

Principles			
Data Storage & Interchange	Notation	Reuse Libraries	Tools & Environments
1175 Tool inter- connection	<i>P1320.x</i> IDEF	1420.x Data model for reuse lib interop	1209 Selection of CASE tools
830 SW rqmts specifications	<i>P1471</i> Architectural description		1348 Adoption of CASE tools
1016 SW design descriptions			
829 SW test documentation			
1016.1 Guide to 1016		1430 Guide to 1420.x	

SESC Product Stack

Principles

Element
Standards

Application
Guides and
Supplements

*Adapted from
[Moore97]*

Principles			
Charac- teristics	Product Measurement	Product Evaluation	End Item Specification
1061 Software quality metrics methodology	1012 SW V & V plans	1362 Concept of operations doc	
	730 SW QA plans		1233 Guide--System reqmts spec
	982.1 Measures for reliable SW		1063 SW user documentation
			1228 SW safety plans
	982.2 Guide to 982.1	1059 Guide--SW V & V plans	
	730.1 Guide--SW QA planning		

SESC Process Stack

*Adapted from
[Moore97]*

Principles

12207.0
Software life cycle (SWLC) processes

Element Standards

General Processes	Primary Processes	Supporting Processes	Process Measurement
----------------------	----------------------	-------------------------	------------------------

1220 Systems eng. process	J-Std-016 Acq/sup agreement	730 SW QA plans	1045 SW product- ivity metrics
1362 Concept of operations doc	830 SW require- ments spec	1298 SW quality mgmt system	
1233 Guide--System rqmts spec	1008 SW unit testing	828 SW CM plans	
1058.1 SW project mgmt plans	829 SW test documentation	1012 SW V & V plans	
1074 Developing SWLC proc	1219 SW maint.	1028 SW reviews and audits	

Application Guides and Supplements

12207.1 Guide--SW life cycle data		730.1 Guide--SW QA planning	
12207.2 Guide--SWLC process		1042 Guide- SW CM	
1074.1 Guide to 1074		1059 Guide--SW V & V plans	

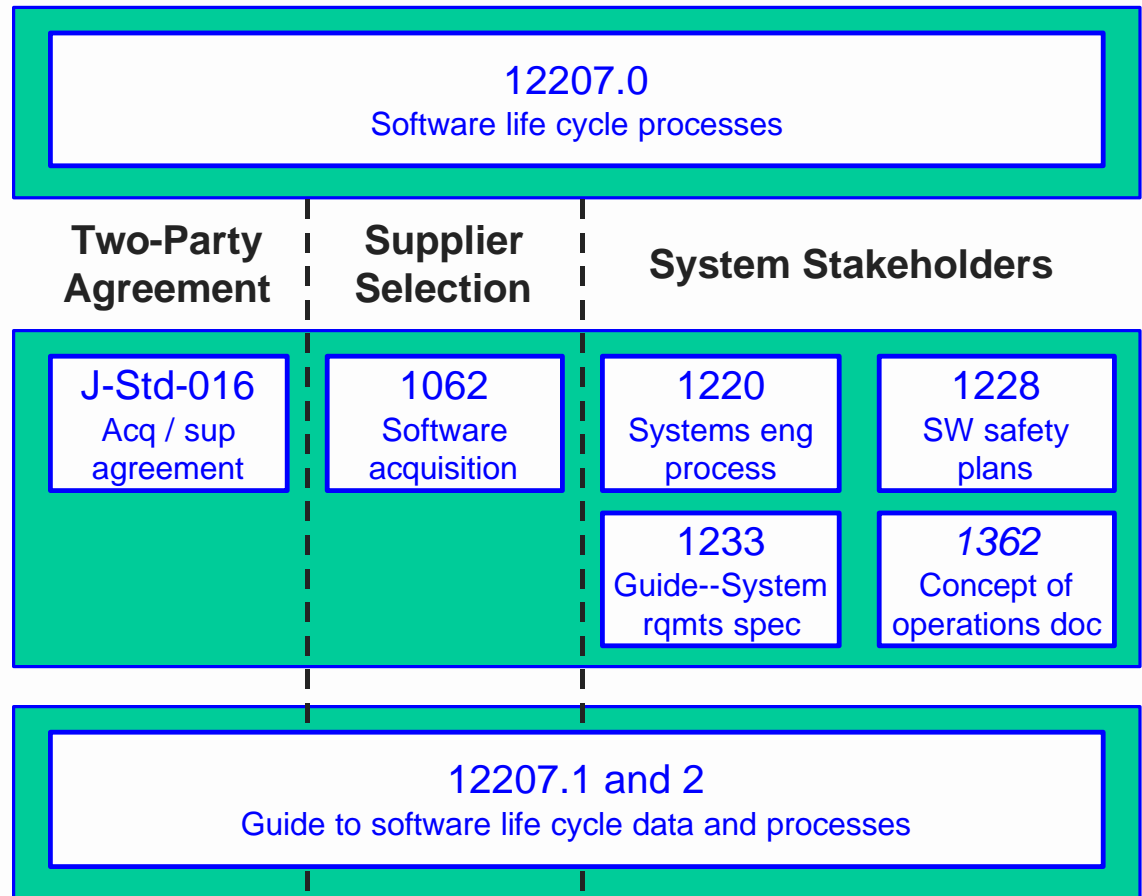
SESC Customer Stack

*Adapted from
[Moore97]*

Principles

**Element
Standards**

**Application
Guides and
Supplements**



History of ISO/IEC 12207

- **Produced by ISO/IEC JTC1/SC7**
- **Participants:**
 - **National bodies: Australia, Canada, Denmark, Finland, France, Germany, Ireland, Italy, Japan, Korea, Netherlands, Spain, Sweden, UK, USA**
 - **Convener: James Roberts (USA)**
 - **Project Editor: Raghu Singh (USA)**
- **Schedule:**
 - **June 1988-August 1995**
 - **4 working drafts, 2 committee drafts, 1 DIS**
 - **17,000 labor-hours of effort [project editor's estimate]**

*This chart is
adapted from a
presentation by
Raghu Singh.*

Purpose of 12207

- **To establish a common framework for the life cycle of software**
 - **Acquire, supply, develop, operate, and maintain software**
 - Undertakes broader scope than previous standards
 - **Manage, control, and improve the framework**
 - Recognizes that software is part of a system and that a project is part of an enterprise
- **To establish a basis for world trade in software**

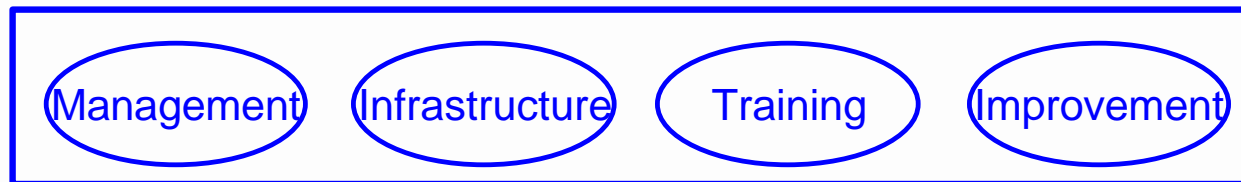
Features of IEEE/EIA 12207

- Defined at level of *processes* ... rather than procedures
- Prescribes *continuing responsibilities* ... rather than steps to be executed
- Prescribes *functions to be performed* ... rather than organizations to execute them
- Prescribes a *minimum systems context* ... rather than treating software in isolation

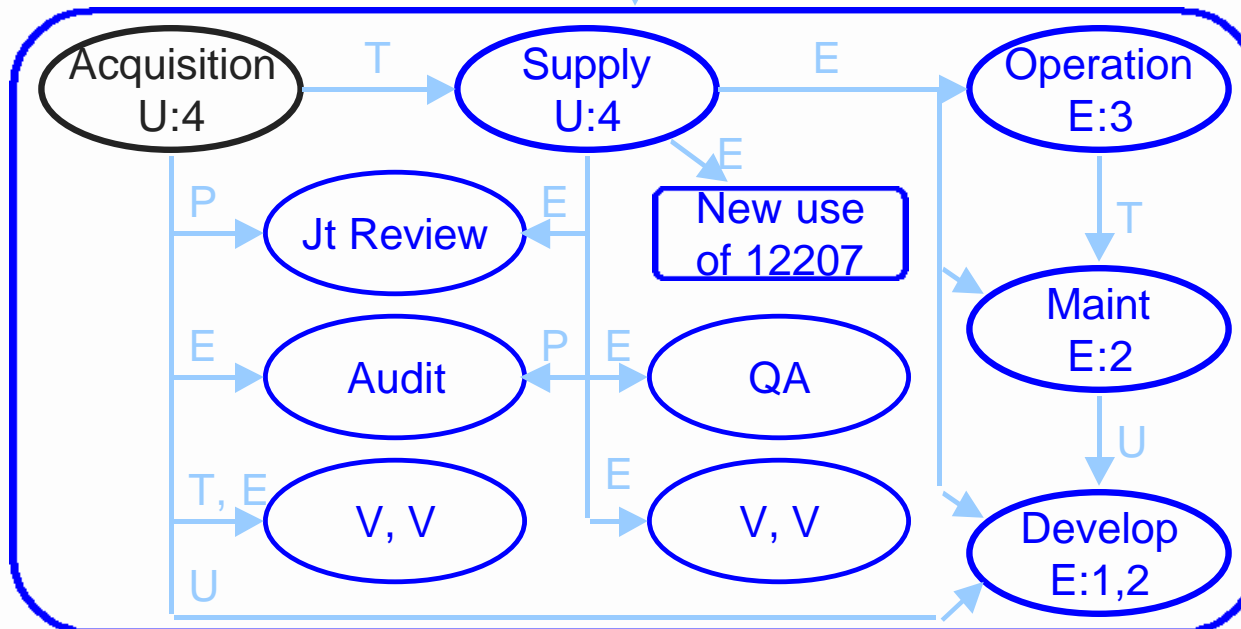
ISO/IEC 12207 processes

Adapted
from:
IEEE/EIA
12207.0
Annex E, by
permission

Organ-
ization



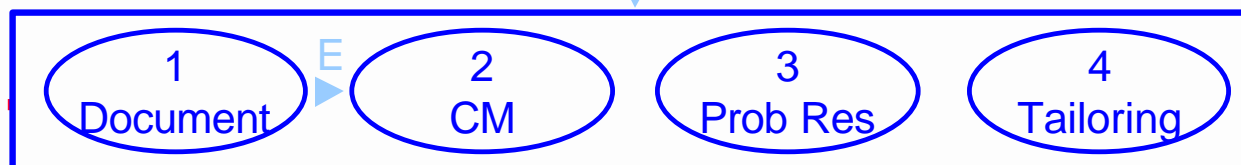
Project



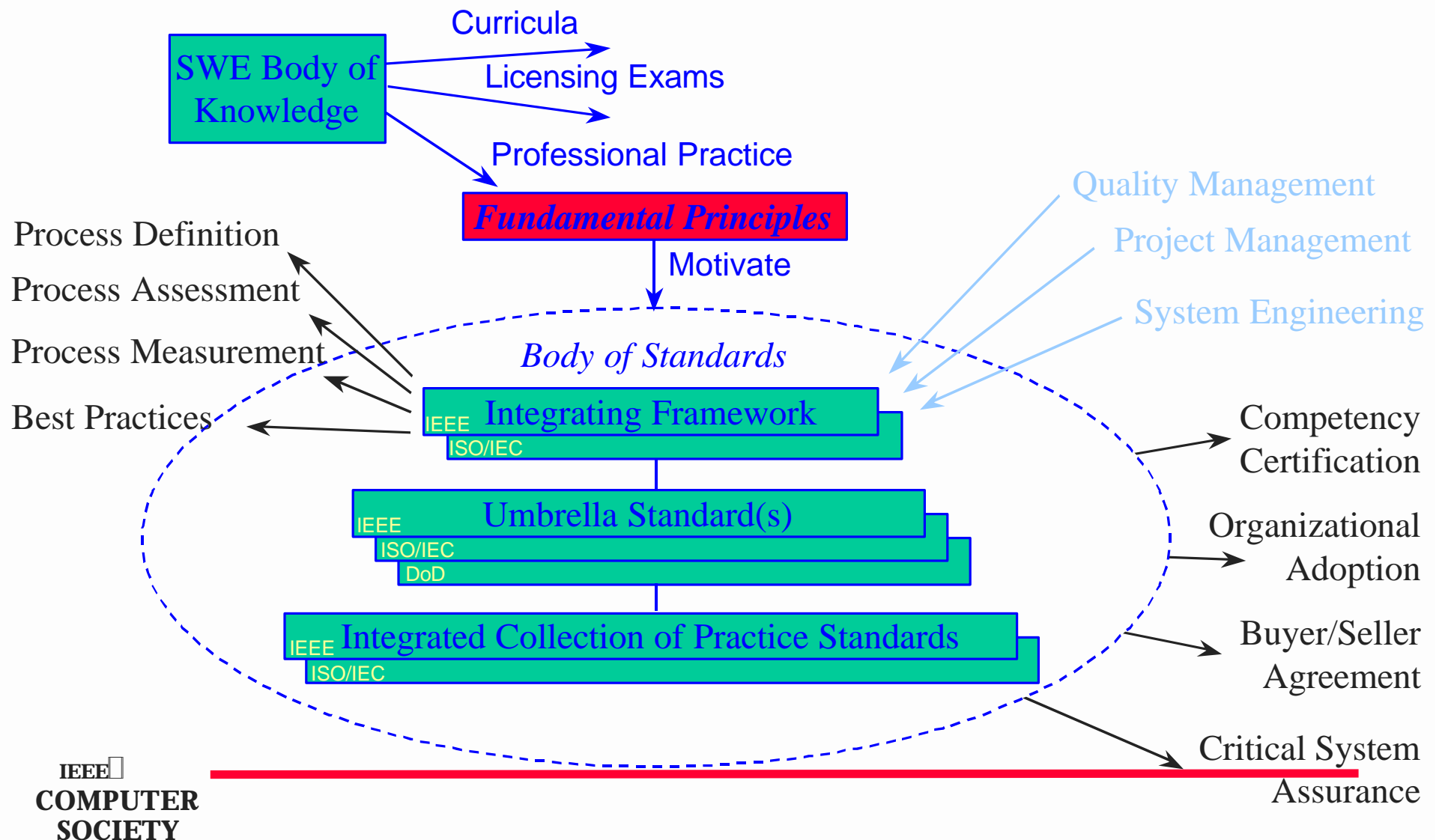
Legend

E: Execute
M: Manage
P: Participate
T: Task
U: Use
Feedback is not
depicted.

Most show E:3.



SW Engineering Standards Five Years from Now



Concluding Remarks

- **Engineering of software is possible**
- **Wide disparity in existing practice of software engineering**
- **Social aspect of software complicates its engineering**
- **Standards are necessary but not sufficient to do software engineering**

References

- [Jabir97] Jabir [group], "A Search for Fundamental Principles of Software Engineering," *Computer Standards and Interfaces*, forthcoming.
- [Moore97] James W. Moore, *Software Engineering: A User's Road Map*, IEEE Computer Society Press, Los Alamitos, CA, 1997.
- [SESC93] SESC Long Range Planning Group, *Master Plan for Software Engineering Standards*, Version 1.0, December 1, 1993.
- [SESC95] SESC Business Planning Group, *Vision 2000 Strategy Statement*, v1.0, SESC/BPG-002, August 20, 1995.
- [SESC98] SESC, *1998 Annual Report*, January 1999.
- [Vincenti] What Engineers Know and How They Know It, Johns Hopkins University Press, 1990.