SYSTEMS AND SOFTWARE CONSORTIUM

BUILDING BETTER SOLUTIONS TOGETHER

Architecting Standard Processes with SWEBOK® and CMMI®

Boris Mutafelija

Harvey Stromberg

Agenda

- Why this paper?
- Short intro to SWEBOK
- Mapping between SWEBOK and CMMI
- Postulate the OSSP architecture
- Introduce other “BOKs”
- Conclusion

Why This Paper?

When consulting in process improvement we often hear:

“CMMI requires …. “

instead of

“We do this because it helps us achieve our goals …”

Quotes:

“CMMI integrates bodies of knowledge that are essential when developing products, but have been addressed separately in the past … “ [2]

SWEBOK was developed by IEEE Computer Society “to promote the advancement of both theory and practice in this field (software engineering)” [1]

So why not use them both when developing an organizational set of standard processes (OSSP)?

What is SWEBOK? (1 of 2)

- "Every profession is based on a body of knowledge and recommended practices, although they are not always defined in a precise manner." [1]
- SWEBOK provides
  - a characterization of the bounds of the software engineering discipline
  - a topical access to the BOK supporting software engineering
- It is subdivided into 10 Knowledge Areas (KAs).
- Each KA contains references to key papers or books that present specific knowledge.
  - The BOK is strongly related to the normative literature (most notably IEEE and ISO/IEC) that provide proven principles that can be successfully implemented.
- Emphasis is less on the science or technology (e.g., computer science, information technology) but on the construction of useful software artifacts.

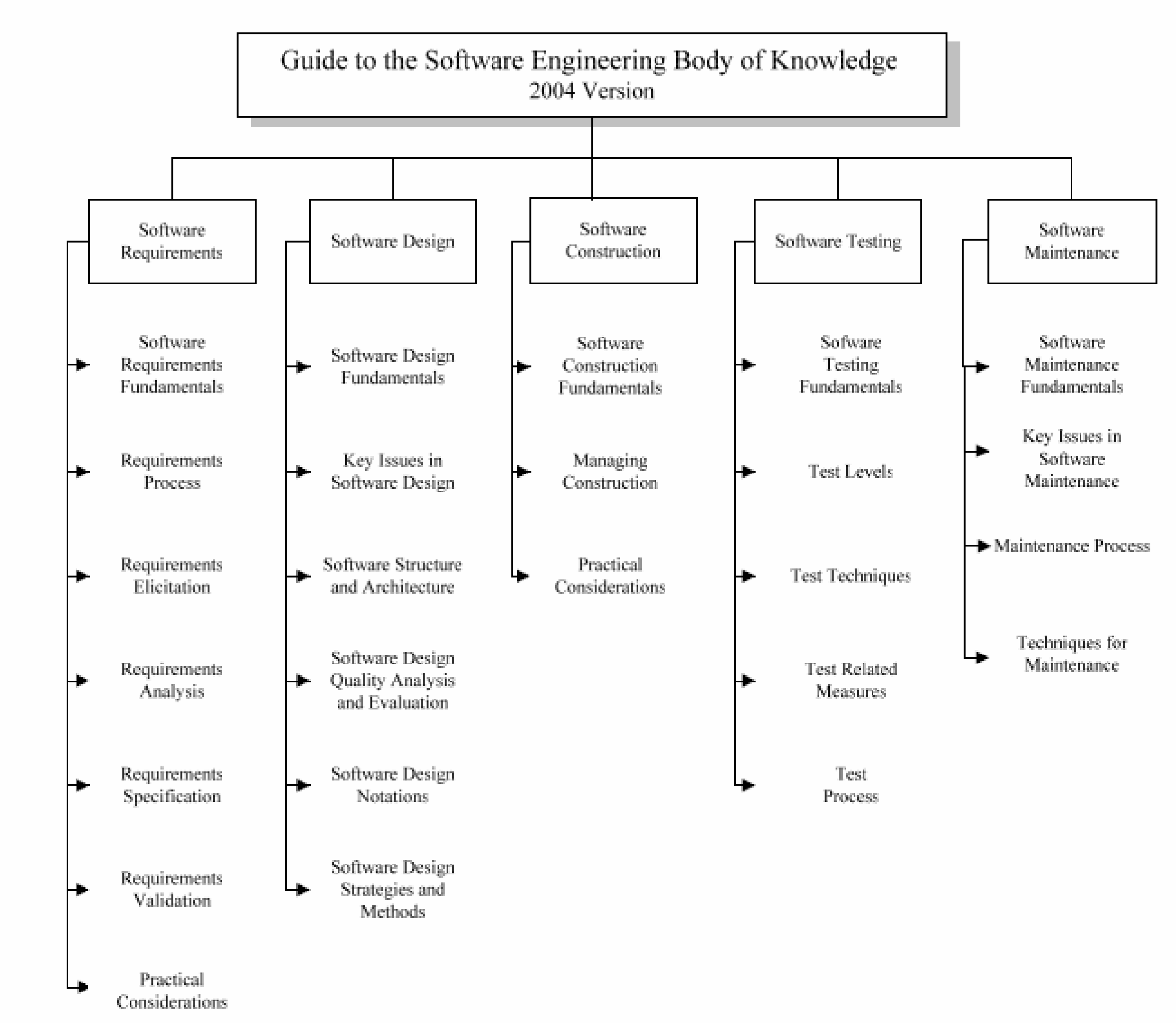
What is SWEBOK? (2 of 2)

- The knowledge described in the Guide is necessary but not sufficient for a software engineer since the practitioners will have to be knowledgeable, among the other things, about project management, systems engineering, computer science, etc.
- The BOK described in the Guide falls within the scope of software engineering and provides references to relevant information from other disciplines.
- Published as ISO Technical Report ISO/IEC TR 19759 ("international consensus on software engineering")

Structure of Knowledge Areas

- Hierarchical organization
  - Set of topics
  - Decomposed into sub-area, topic, and sub-topic
  - The breakdown does not presuppose application domains, business use, management philosophy, development methods, etc.
- Reference material
  - Provides best presentation and coverage of topics described
  - Limited to 500 pages of reference material
  - It is not comprehensive in its citations (distinguishes between “recommended” and “usual” references)
- Depth of treatment
  - Generally accepted knowledge applies to most projects most of the time and widespread consensus validates its value and effectiveness [3]

SWEBOK Structure (1 of 2)

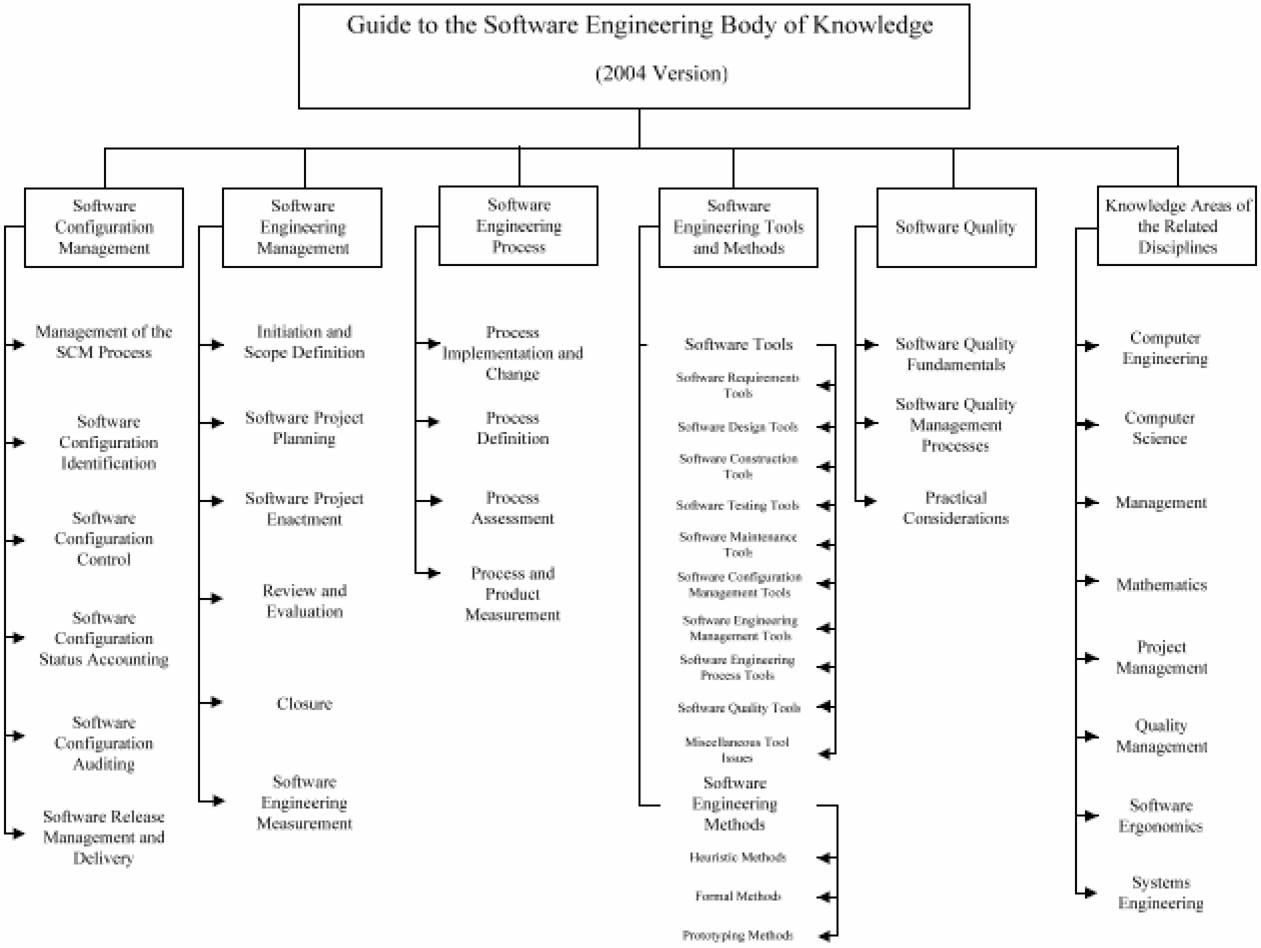


Guide to the Software Engineering Body of Knowledge
2004 Version

- Software Requirements
  - Software Requirements Fundamentals
  - Requirements Process
  - Requirements Elicitation
  - Requirements Analysis
  - Requirements Specification
  - Requirements Validation
  - Practical Considerations
- Software Design
  - Software Design Fundamentals
  - Key Issues in Software Design
  - Software Structure and Architecture
  - Software Design Quality Analysis and Evaluation
  - Software Design Notations
  - Software Design Strategies and Methods
- Software Construction
  - Software Construction Fundamentals
  - Managing Construction
  - Practical Considerations
- Software Testing
  - Software Testing Fundamentals
  - Test Levels
  - Test Techniques
  - Test Related Measures
  - Test Process
- Software Maintenance
  - Software Maintenance Fundamentals
  - Key Issues in Software Maintenance
  - Maintenance Process
  - Techniques for Maintenance

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

SWEBOK Structure (2 of 2)



Guide to the Software Engineering Body of Knowledge

(2004 Version)

- 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 Engineering Management
  - Initiation and Scope Definition
  - Software Project Planning
  - Software Project Enactment
  - Review and Evaluation
  - Closure
  - Software Engineering Measurement
- Software Engineering Process
  - Process Implementation and Change
  - Process Definition
  - Process Assessment
  - Process and Product Measurement
- Software Engineering Tools and Methods
  - Software Tools
    - Software Requirements Tools
    - Software Design Tools
    - Software Construction Tools
    - Software Testing Tools
    - Software Maintenance Tools
    - Software Configuration Management Tools
    - Software Engineering Management Tools
    - Software Engineering Process Tools
    - Software Quality Tools
    - Miscellaneous Tool Issues
  - Software Engineering Methods
    - Heuristic Methods
    - Formal Methods
    - Prototyping Methods
- Software Quality
  - Software Quality Fundamentals
  - Software Quality Management Processes
  - Practical Considerations
- Knowledge Areas of the Related Disciplines
  - Computer Engineering
  - Computer Science
  - Management
  - Mathematics
  - Project Management
  - Quality Management
  - Software Ergonomics
  - Systems Engineering

SWBOK Knowledge Areas (KAs) *

1. Software requirements
2. Software design
3. Software construction
4. Software testing
5. Software maintenance
6. Software configuration management
7. Software quality
8. Software engineering management
9. Software engineering process
10. Software engineering tools and methodologies

Engineering processes

Support processes

Management processes

Process Management processes

* Not in the same order as in the SWEBOK

Example: Requirements KA (1 of 2)

- Software Requirements Fundamentals
  - Definition of requirements (systems vs. software, product vs. process; functional vs. non-functional, etc.)
  - Importance of quantifiable requirements
- Requirements Process
  - Process modeling throughout the life cycle
  - Roles/participants/stakeholders in the requirements process
  - Resources needed
  - Process quality improvement
    - impact of requirements process on cost & timeliness of product delivery and customer satisfaction
- Requirements Elicitation
  - Determining requirements sources
  - Elicitation techniques
  - Building understanding of the problem and emerging solution
  - Importance of good communication

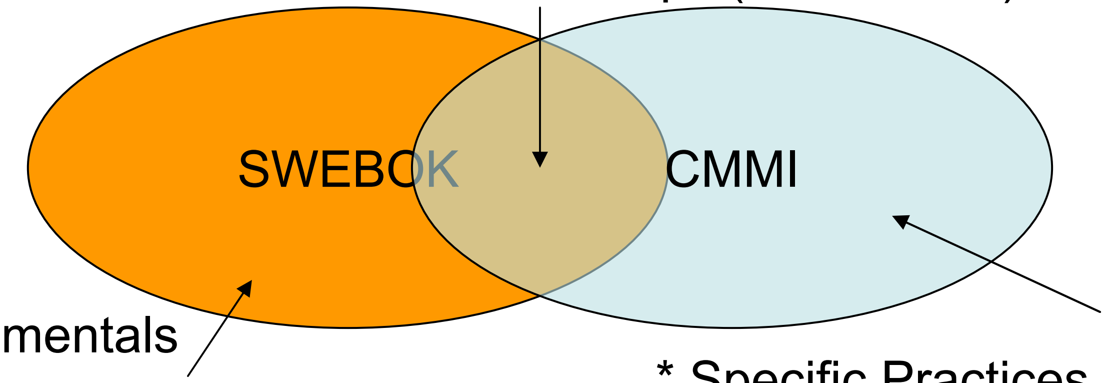
Example: Requirements KA (2 of 2)

- Requirements Analysis
  - Detect and resolve conflicts among requirements; elaborating systems requirements to better define software requirements
  - Requirements classification and model development
  - Architectural design; requirements allocation
  - Resolving conflicts and reaching consensus among the stakeholders
- Requirements Specification
  - Systems Definition Document; Systems Requirements Specification; Software Requirements Specifications
- Requirements Validation
  - Reviews; Prototyping; Model Validation; Acceptance Tests
- Practical Considerations
  - Difficulty in spanning the whole life cycle
  - Iterative, non-linear nature of the process
  - Change management
  - Traceability
  - Measurements

SWEBOK and CMMI

- SWEBOK has narrower scope but is more detailed
- CMMI is much broader but, in some instances, less detailed

* Roles / Responsibilities
* Resources
* Stakeholders
* Process Steps (most cases)



* Fundamentals
* How-to
* Practical Considerations
* References
* Further Reading

* Specific Practices = Process
* Generic Practices = Institutionalization
* OSSP vs. Project’s Defined Process
* What

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

Similarities – Differences

SWEBOK	CMMI
KA matches PA well (example: Software CM)	PA matches KA well (example: CM)
No match in CMMI (example: Software Construction)	No match in SWEBOK (example: DAR)
Single KA – similar detail (example: Software Requirements)	Multiple Process Areas – similar detail (mapping: RD, REQM)
Single KA (example: Software Engineering Management)	Multiple Process Areas – more detail in CMMI (mapping: PP, PMC, MA, QPM, SAM, RSKM)

SWEBOK – CMMI Synergy

- Where CMMI outlines steps (Specific Practices), SWEBOK provides process details, additional references, and further reading
- Where CMMI defines Typical Work Products, SWEBOK provides details of their content, some narrative, and additional references, and further reading
- Where SWEBOK lists roles, responsibilities, resources, etc., CMMI provides institutionalization structure in the form of Generic Practices

CMMI differentiates between the organizational standard and project's defined process, while SWEBOK acknowledges those differences in passing.

SWEBOK – CMMI Synergy

- CMMI requires definition of process elements and their relationships
- SWEBOK provides details

- What is needed:

  Something that relates process elements and associated details

> Process Architecture

or

> a Framework

Process Architecture

Definitions (CMMI):

- *OSSP:* A collection of definitions of the process that guide activities in an organization. These process descriptions cover the fundamental process elements (and relationships to each other, such as ordering of interfaces) that must be incorporated into the defined processes that are implemented in projects across the organization.
- *Process Element:* The fundamental unit of a process. Each process element covers a closely related set of activities.
- *Process Architecture:* The ordering, interfaces, interdependencies, and other relationships among the process elements in a standard process.

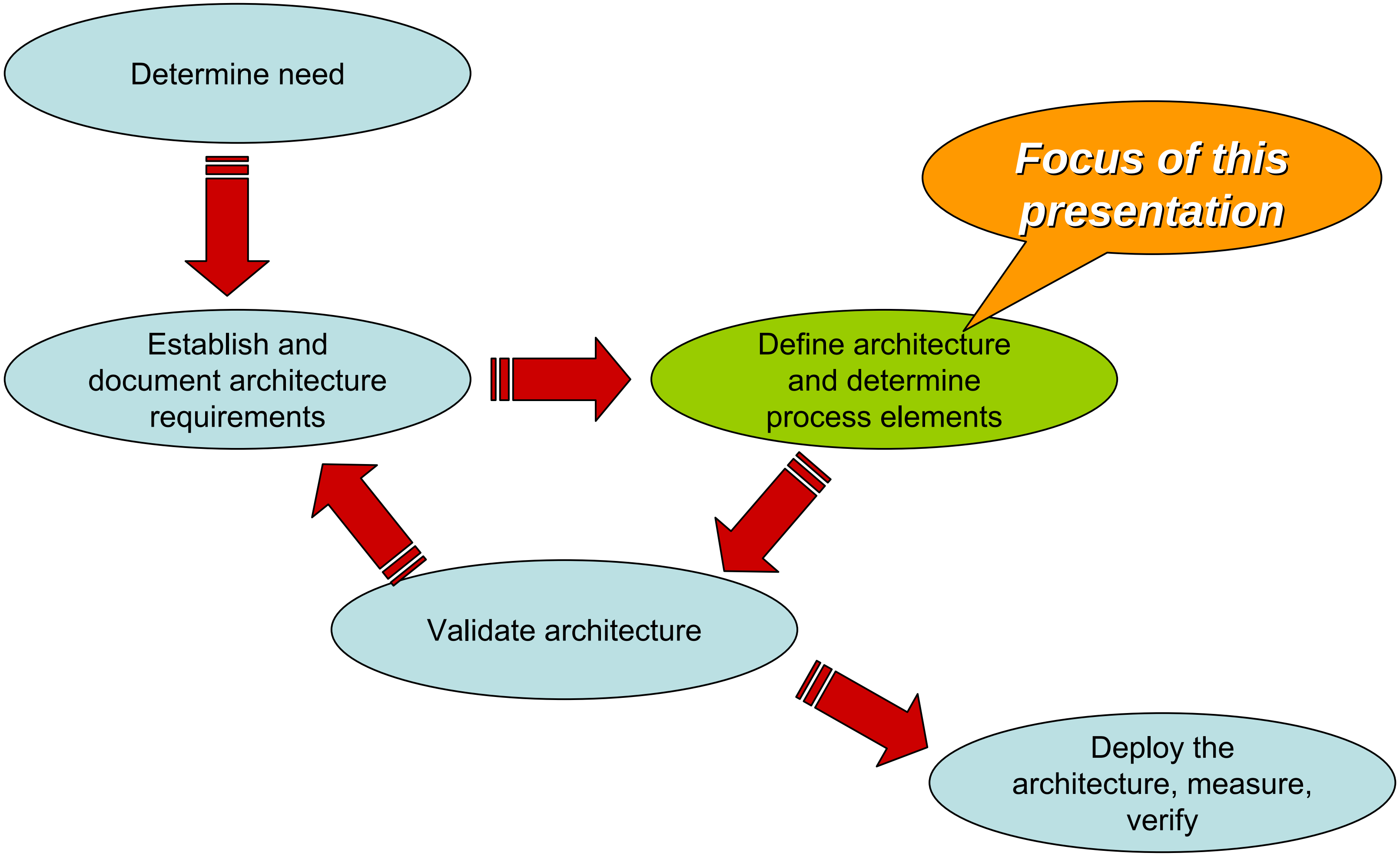
Process Architecture (cont'd)

Based on those definitions:

- OSSP is equivalent to a process architecture or a framework
- Process Elements that populate this architecture/framework are defined in such a way that they exhibit certain properties:
  - Abstraction
  - Modularity
  - Cohesion
  - Minimized coupling

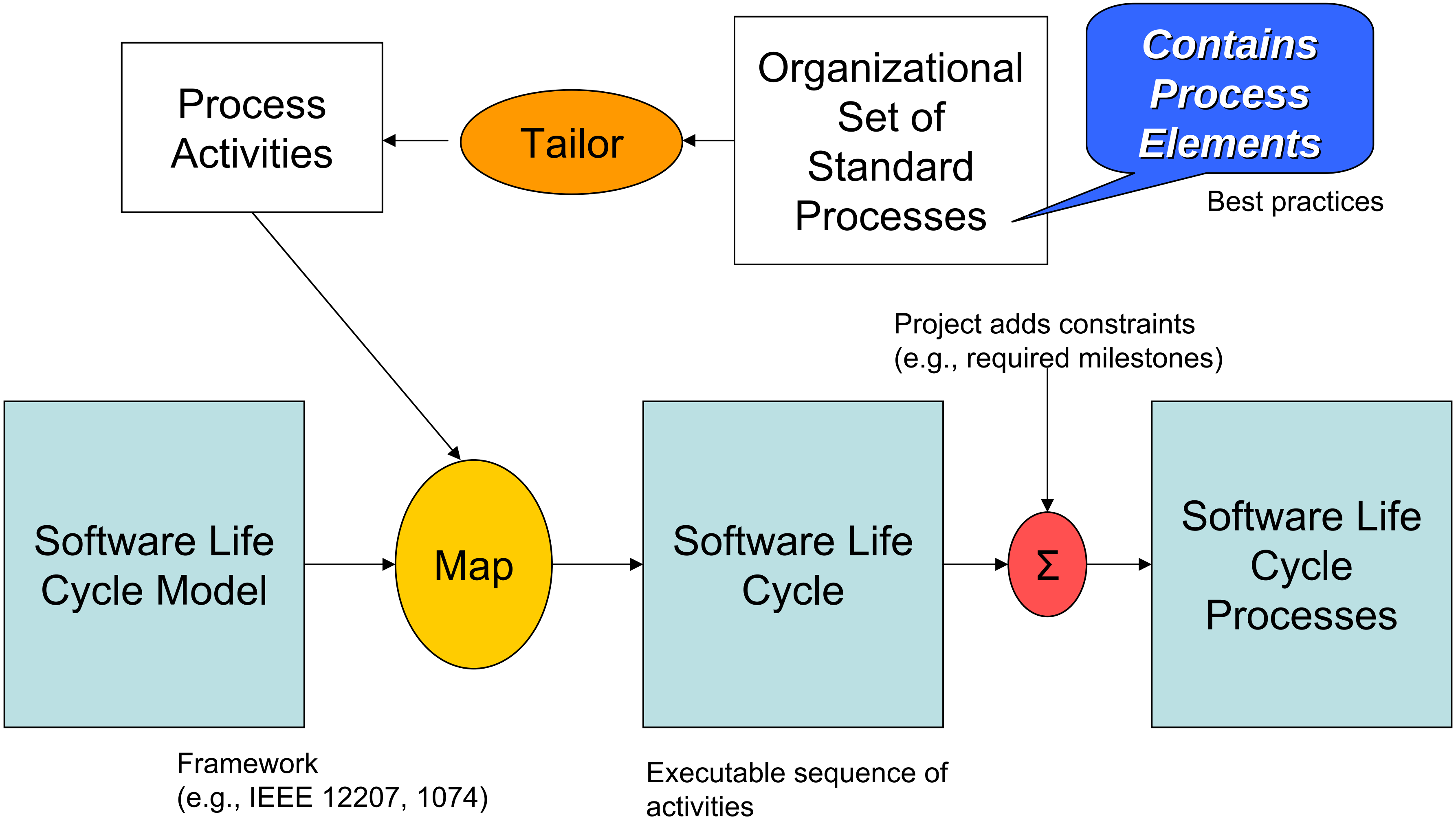
Problem:
Identify flexible and extendable Process Elements

Process Architecture Definition Process *


Determine need
Focus of this presentation
Establish and document architecture requirements
Define architecture and determine process elements
Validate architecture
Deploy the architecture, measure, verify

* Dana Bredemeyer, 2000

Generalized Process Approach


Process Activities
Tailor
Organizational Set of Standard Processes
Contains Process Elements
Best practices
Project adds constraints (e.g., required milestones)
Software Life Cycle Model
Map
Software Life Cycle
Σ
Software Life Cycle Processes
Framework (e.g., IEEE 12207, 1074)
Executable sequence of activities

Process Definitions – Questions to be answered*

- What tasks comprise the process?
- Who performs them?
- When do they take place?
- How are they implemented?
- Does the definition stand up to a formal audit?
- Do the definitions supports continuous process improvement?

Real World Constraints:
Schedule, Cost, Contractual Requirements
Organization size, maturity, products

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

*J. Henry, B. Blasewitz, 1992

Process Definitions – 3 Views

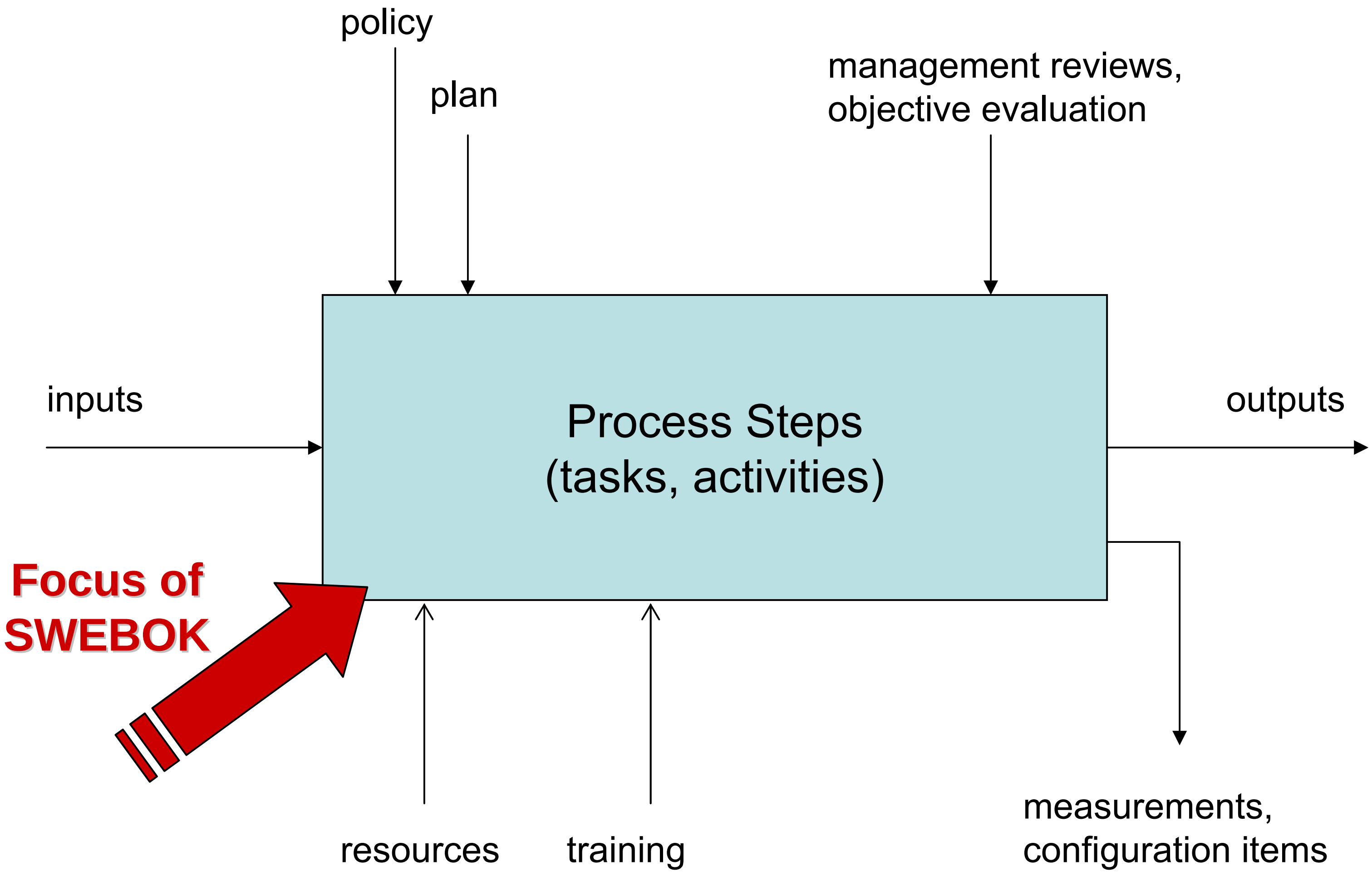
- Functional
  - Task descriptions
- Behavioral
  - When and how are tasks performed
- Organizational
  - Who performs the tasks

- Meta Architecture
  - Guiding principles and strategies
  - Basis for system decomposition and synthesis

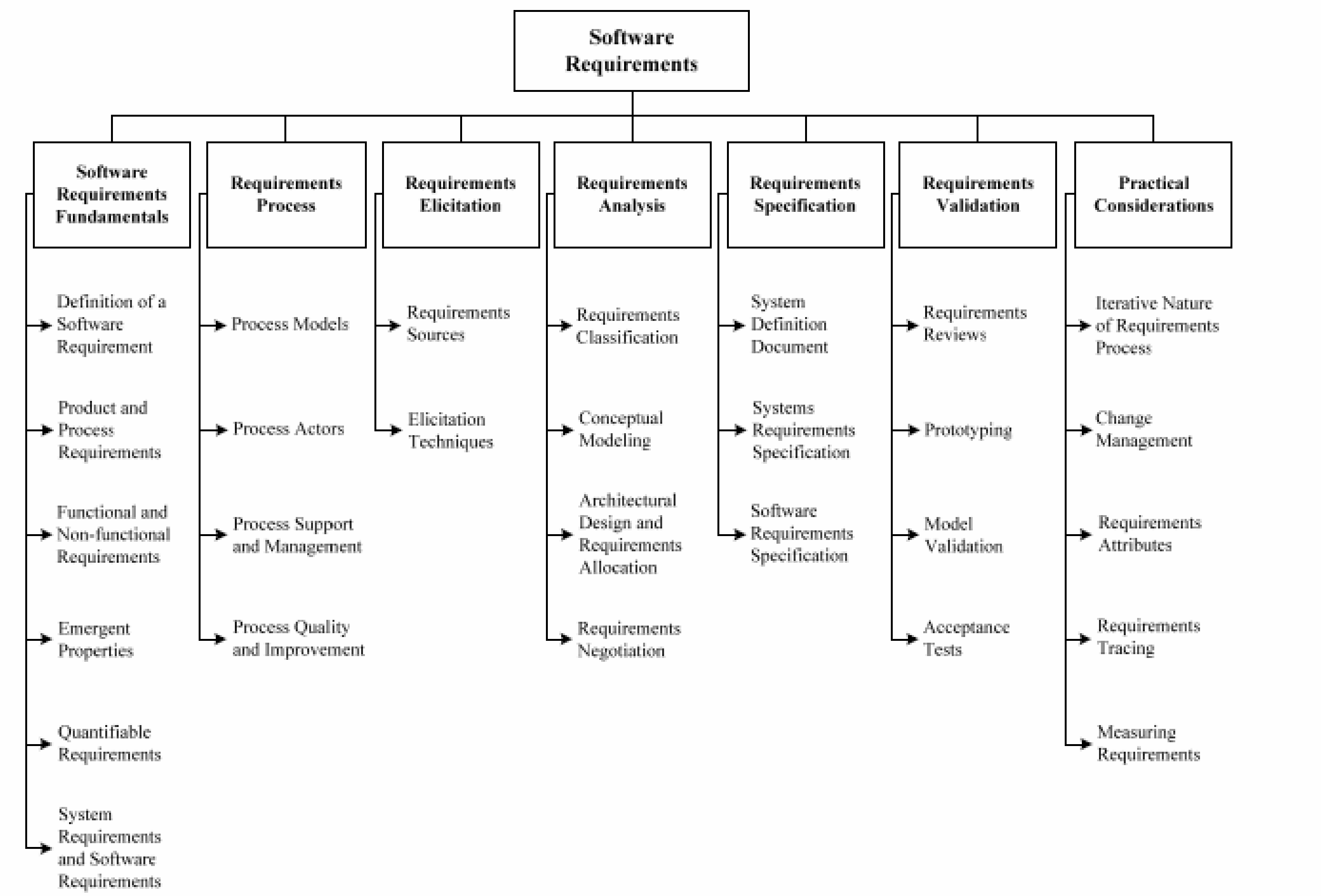
Process Element Description Components

- Entry and Exit Criteria
- Inputs and Outputs
- Activities
- Roles / Responsibilities
- Stakeholders
- Measurements
- Controls
  - Verification
  - Configuration management
- Related processes
- Tools / Standards / Training

Typical Process Element



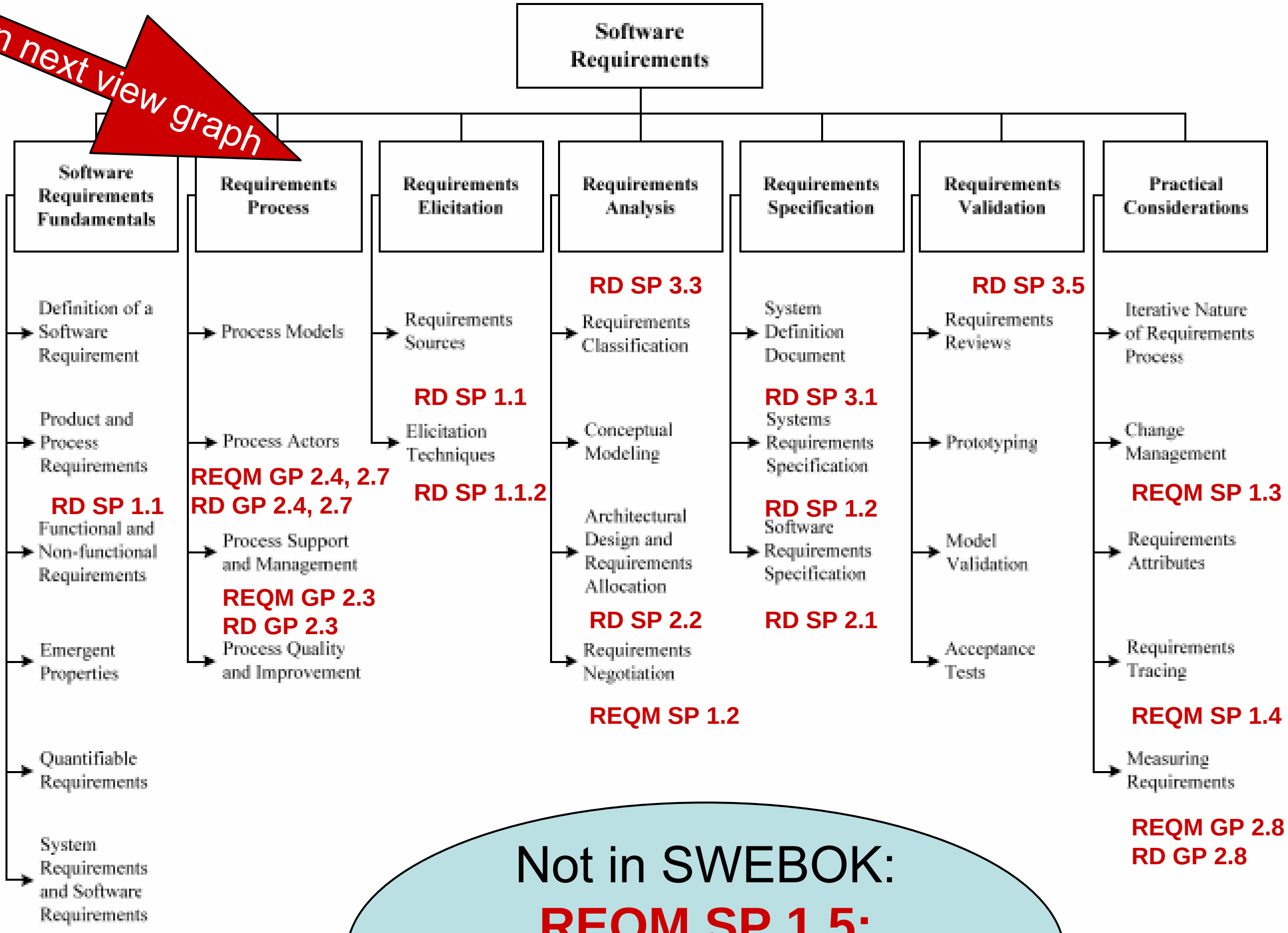
SWEBOK: S/W Requirements vs. CMMI


Software Requirements
Software Requirements Fundamentals
Definition of a Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Requirements Process
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Elicitation
Requirements Sources
Elicitation Techniques
Requirements Analysis
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
Requirements Specification
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Validation
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Practical Considerations
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

SWEBOK: S/W Requirements vs. CMMI

Details on next view graph

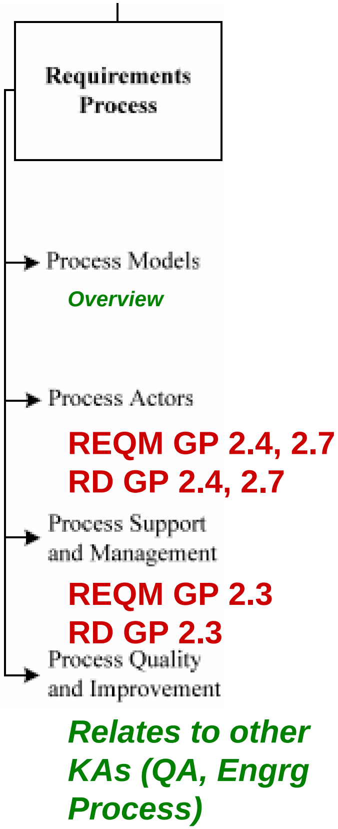


Software Requirements

- Software Requirements Fundamentals
  - Definition of a Software Requirement
  - Product and Process Requirements
  - RD SP 1.1
  - Functional and Non-functional Requirements
  - Emergent Properties
  - Quantifiable Requirements
  - System Requirements and Software Requirements
  - RD SP 1.1
- Requirements Process
  - Process Models
  - Process Actors
  - REQM GP 2.4, 2.7
  - RD GP 2.4, 2.7
  - Process Support and Management
  - REQM GP 2.3
  - RD GP 2.3
  - Process Quality and Improvement
- Requirements Elicitation
  - Requirements Sources
  - RD SP 1.1
  - Elicitation Techniques
  - RD SP 1.1.2
- Requirements Analysis
  - RD SP 3.3
  - Requirements Classification
  - Conceptual Modeling
  - Architectural Design and Requirements Allocation
  - RD SP 2.2
  - Requirements Negotiation
  - REQM SP 1.2
- Requirements Specification
  - System Definition Document
  - RD SP 3.1
  - Systems Requirements Specification
  - RD SP 1.2
  - Software Requirements Specification
  - RD SP 2.1
- Requirements Validation
  - RD SP 3.5
  - Requirements Reviews
  - Prototyping
  - Model Validation
  - Acceptance Tests
- Practical Considerations
  - Iterative Nature of Requirements Process
  - Change Management
  - REQM SP 1.3
  - Requirements Attributes
  - Requirements Tracing
  - REQM SP 1.4
  - Measuring Requirements
  - REQM GP 2.8
  - RD GP 2.8

Not in SWEBOK:
REQM SP 1.5;
GP 2.1, 2.2, 2.5, 2.6, 2.10, 3.1, 3.2

Requirements Process (multiple PA match)



Requirements Process

Process Models

Overview

Process Actors

REQM GP 2.4, 2.7
RD GP 2.4, 2.7

Process Support and Management

REQM GP 2.3
RD GP 2.3

Process Quality and Improvement

Relates to other KAs (QA, Engrg Process)

- This section provides process details
  - Understanding of the process (continues over several phases; marketing & feasibility studies)
  - Who are the stakeholders?
    - Users
    - Customers
    - Software engineers
    - Etc.
  - Resources required
  - Process quality and improvements
    - Cost & timeliness
    - Customer satisfaction
    - Standards
    - Interfaces with other KAs:
      - Software Quality
      - Software engineering

Process Element Example: Requirements Engineering

CMMI introductory and other notes contain information found in more detail in SWEBOK Chapter 2, Software Requirements, Section 1.0

Process Step	Description	CMMI	SWEBOK Chapter 2 Sections
1	Collect Needs	RD SP 1.1	3.1, 3.2
2	Develop Requirements		
	Develop customer requirements	RD SP 1.2, 3.1	4.1. 4.2 7.3
	Develop product requirements	RD SP 2.1	
	Allocate requirements	RD SP 2.2, 2.3	4.3
	Develop Interface requirements	RD SP 2.3	None
3	Document requirements	In Typical Work Products	All 5.0
4	Maintain Traceability	REQM 1.4	7.4 (weak)
5	Analyze and validate requirements	RD 3.3, 3.4, 3.5 REQM 1.1, 1.2	4.1, 4.2, 4.4 All 6.0
6	Manage requirements changes	REQM 1.3	7.2
7	Identify inconsistencies between requirements and work products	REQM 1.5	None

- **SWEBOK Section 2 presents overall requirements process**
- **CMMI presents Requirements Engineering in two PAs: RD and REQM**

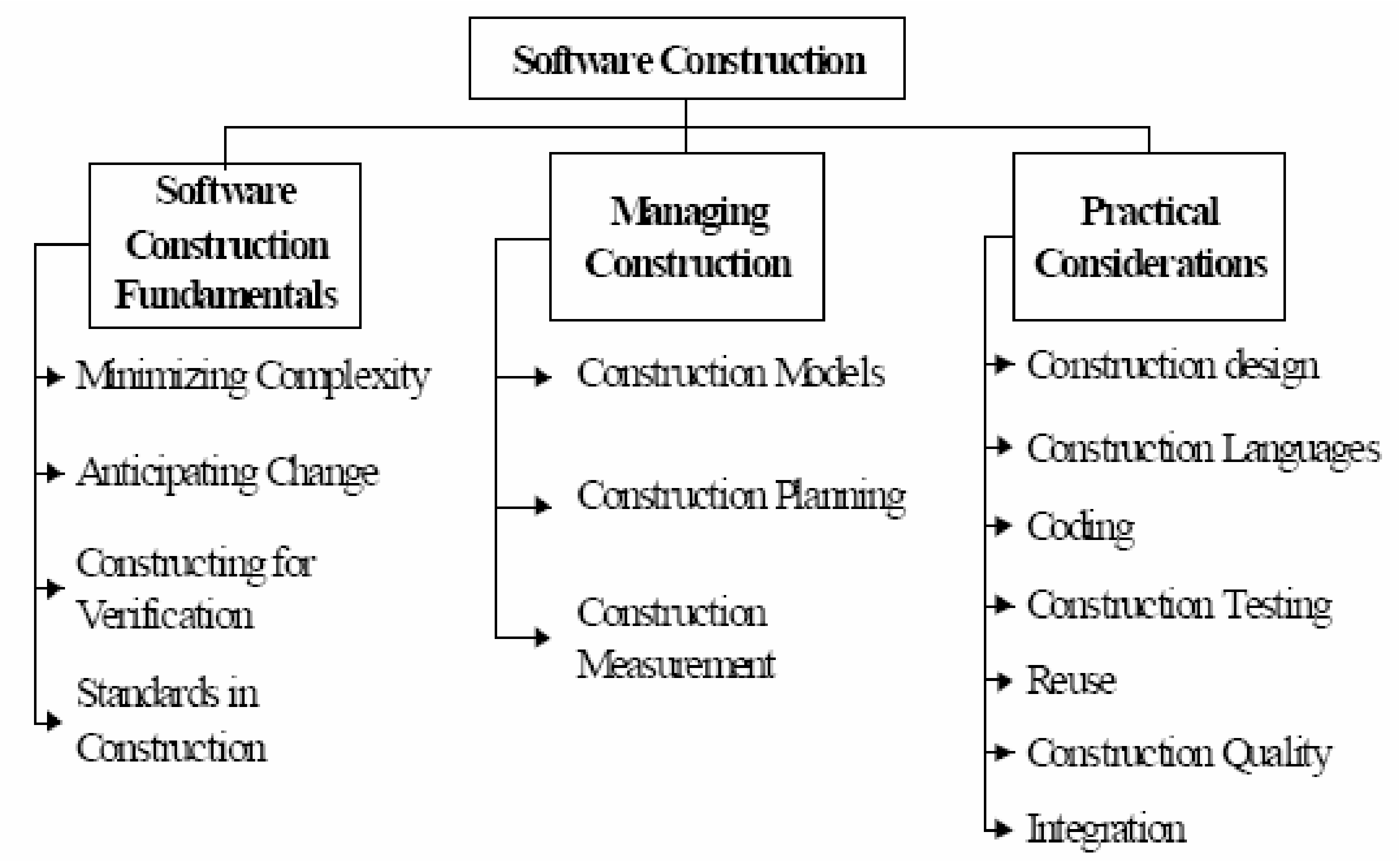
SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

Example: Requirements Engineering PE

Comments and Notes:

- SWEBOK is very weak in indicating institutionalization therefore some GPs are not addressed.
  - Exceptions: GP 2.3, 2.4, 2.7, and 2.8
- SWEBOK does not differentiate between the organizational standard process and the project's defined process
- SWEBOK has no notion of capability or maturity levels.
- The iterative nature of the requirement process addressed in SWEBOK Section 7.1 is explained in Chapter 4 of the CMMI book.
- Neither SWEBOK nor CMMI address process element interactions
  - CMMI provides very high level PA interaction diagrams
  - CMMI lists Typical Work Products for each SP that may be used as a guide for interfacing the process elements.

Software Construction Process (no direct match)



Software Construction

- Software Construction Fundamentals
  - Minimizing Complexity
  - Anticipating Change
  - Constructing for Verification
  - Standards in Construction
- Managing Construction
  - Construction Models
  - Construction Planning
  - Construction Measurement
- Practical Considerations
  - Construction design
  - Construction Languages
  - Coding
  - Construction Testing
  - Reuse
  - Construction Quality
  - Integration

Process Element Example: Software Construction

CMMI introductory notes to SP 3.1 allude to the iterative nature of product development.

Process Step	Description	CMMI	SWEBOK Chapter 4 Sections
1	Use effective methods to implement the design	TS SP 3.1 subpractice 1	1.1; 1.2; 1.3; 2.1
2	Adhere to applicable standards and criteria	TS SP 3.1 subpractice 2	1.4; 3.2; 3.3; 3.5
3	Construct software component	none	none
4	Conduct peer review of the selected components	TS SP 3.1 subpractice 3	3.6
5	Perform unit testing of the component as appropriate	TS SP 3.1 subpractice 4	3.4
6	Revise the component as necessary	TS SP 3.1 subpractice 5	none
7	Integrate separately developed components	PI SP 3.2	3.7

Example: Software Construction PE

Comments and Notes:

- SWEBOK provides detailed guidelines for software construction; CMMI is by its nature (systems engineering, software engineering, IPPD, SS) more generic, however some examples listed in the practice amplifications are more elaborate
- CMMI (in the Technical Solution PA) provides possible process steps that are amplified by SWEBOK
- SWEBOK provides some guidelines for planning (Section 2.7) that are typically addressed in GP 2.2
- SWEBOK identifies potential measurements (Section 2.3) which are more software-specific than those listed in GP 2.8
- In SWEBOK peer reviews are not specifically mentioned
- SWEBOK has no notion of capability or maturity levels.
- Institutionalization is not indicated in SWEBOK

SWEBOK – to – CMMI: High-level Maps *

SWEBOK	CMMI
Software Requirements	RD, REQM
Software Design	TS
Software Construction	TS – level of detail is less
Software Testing	VER, VAL are much more general
Software Maintenance	Requires interpretation of CMMI PAs
Software Configuration Management	CM
Software Engineering Management	REQM, PP, PMC, SAM (ISM), MA, RSKM, QPM
Software Engineering Process	OPF, OPD, MA, QPM
Software Quality	PPQA, VER, VAL, MA

Conclusions

- CMMI and SWEBOK are synergistic
  - CMMI provides more structure
  - SWEBOK provides more detailed “tutorial” information
  - Using SWEBOK when performing process improvement enables organizations to move from “doing CMMI” to “implementing best practices”
- Process architecture is required to define process interactions and interfaces
  - Some of the international standards may be consulted when developing such process architecture (e.g., IEEE Std 1074: 1997; ISO/IEC 12207:1995 including Amd. 1: 2002; etc.) – they provide another, more procedural, look at the process element interaction and implementation

Other BOKs

- A Guide to Project Management BOK (PMBOK® Guide)
  - A National Standard (ANSI/PMI 99-001-2000)
  - Divided into set of Knowledge Areas
    - Project Integration Management
    - Project Scope Management
    - Project Integration Management
    - Project Time Management
    - Project Cost Management
    - Project Quality Management
    - Project Human Resource Management
    - Project Communication Management
    - Project Risk Management
    - Project Procurement Management

Other Emerging BOKs

- Systems Engineering BOK (SEBoK)
  - Developed by INCOSE
  - Defines SE broadly in terms of its concepts, processes, skills required, and capability
  - Virtually no new information, but extensive references, organization of material, and context

  [Leibrandt, et al, *A Guide to the SE Body of Knowledge*, INCOSE]
- Software Measurements KA (BoK)
  - Measurements is a common theme in SWEBOK
  - Proposal breakdown of this new KA with analysis of empirical support for a new KA

  [Buglione, L., A. Abram, ETS Montreal, CA]

Other Emerging BOKs

- Information Systems Engineering BoK
  - More organic approach
  - Fully integrated and completely consistent for the Information Systems Engineering
  - "compile and structure a library filled with a collection of significant, and loosely coupled knowledge items …"

  [Proper, H. A., et al, *Towards an Information Systems Engineering BoK*, Proceedings of the 2nd National Architecture Congress, Amsterdam, 2000]
- Software Assurance BoK
  - Initial effort concentrated on achieving and assuring security properties and functionality
  - Contains common concepts and principles required across acquiring, developing, and sustaining secure software

  [Redwine, Jr., S. T., www.stc.hill.af.mil, Oct 2005 – BoK to be published in December 2005]

References

1- IEEE Computer Society, *SWEBOK, Guide to the Software Engineering Body of Knowledge,* 2004 Version

2 – Chrissis, M.B., at al., *CMMI, Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2003

3 – Project Management Institute Standards Committee, *A Guide to the Project Management Body of Knowledge (PMBOK),* Project Management Institute, 2000

4 – Gray, L., An OSSP Architecture Based on IEEE/EIA 12207 and CMMI-SE/SW/IPPD/SS, SEPG Conference, 2002, Phoenix, AZ

5 – Buglione, L., A. Abran, *The Software Measurement Body of Knowledge*, Software Measurement European Forum, Rome, 2004

6 – Bredemeyer, D., *Software Architecture Action Guide*, Bredemeyer Consulting, 2000

7 – Henry, J., B. Blasewitz, *Process Definition Theory and Reality*, IEEE Software, November 1992

8 – Land, S.K., Jumpstart CMM/CMMI Software Process Improvement using IEEE Software Engineering Standards, Wiley-Interscience, 2005

9 – Sherer, W., S. Thrasher, *Contrasting CMMI and the PMBOK*, 8th Annual NDIA Systems Engineering Conference, October 2005

Authors Addresses

Systems and Software Consortium, Inc.
2214 Rock Hill Rd.
Herndon, VA 20170

mutafelija@systemsandsoftware.org
stromberg@systemsandsoftware.org