Architecting Standard Processes with SWEBOK® and CMMI®

Boris Mutafelija
Harvey Stromberg

SWEBOK® is an official service mark of the IEEE.
CMMI® is registered in the US Patent and Trademark Office by Carnegie Mellon University.
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”)

Available Free on the Web
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)
SWEBOK Structure (2 of 2)
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

* 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
Similarities – Differences

**SWEBOK**
- KA matches PA well (example: Software CM)
- No match in CMMI (example: Software Construction)
- Single KA – similar detail (example: Software Requirements)
- Single KA (example: Software Engineering Management)

**CMMI**
- PA matches KA well (example: CM)
- No match in SWEBOK (example: DAR)
- Multiple Process Areas – similar detail (mapping: RD, REQM)
- 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
- 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

- Software Life Cycle Model
  - Map
  - Software Life Cycle
    - Executable sequence of activities

- Project adds constraints (e.g., required milestones)

- Framework (e.g., IEEE 12207, 1074)
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 support continuous process improvement?

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

*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

Focus of SWEBOK

Policy

Plan

Management reviews, objective evaluation

Inputs

Process Steps (tasks, activities)

Outputs

Resources

Training

Measurements, configuration items
SWEBOK: S/W Requirements vs. CMMI
SWEBOK: S/W Requirements vs. CMMI

Software Requirements

Requirements Process
- Process Models
  - Process Support and Management
    - Process Actors
      - REQM GP 2.4, 2.7
      - RD GP 2.4, 2.7
    - Process Quality and Improvement
      - REQM GP 2.3
      - RD GP 2.3

Requirements Elicitation
- Requirements Sources
  - RD SP 1.1
    - Elicitation Techniques
      - REQM SP 1.2
    - Conceptual Modeling
    - Architectural Design and Requirements Allocation
      - RD SP 2.2

Requirements Analysis
- RD SP 3.3
  - Requirements Classification
    - System Definition Document
    - RD SP 3.1
      - Systems Requirements Specification
      - RD SP 1.2
        - Software Requirements Specification
        - RD SP 2.1
          - Requirements Validation
            - Acceptance Tests
            - RD SP 3.5
              - Requirements Reviews
                - Prototyping
                - Model Validation

Requirements Specification
- RD GP 2.8
  - Measuring Requirements
    - REQM GP 2.8
    - RD GP 2.8

Requirements Validation
- RD GP 2.1
  - Requirements Validation
    - Acceptance Tests
    - RD GP 2.2
      - Requirements 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

Not in SWEBOK:
- REQM SP 1.5;
  - GP 2.1, 2.2, 2.5, 2.6, 2.10, 3.1, 3.2
- RD GP 2.1
  - Requirements Validation
    - Acceptance Tests
    - RD GP 2.2
      - Requirements Validation
        - Acceptance Tests

Details on next view graph

System Requirements and Software Requirements
- RD SP 1.1
  - Qualifiable Requirements
  - Emergent Properties
  - Functional and Non-functional Requirements
    - RD SP 1.1
      - 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
        - REQM SP 1.2
  - Process Models
    - RD SP 1.1
      - Elicitation Techniques
        - REQM SP 1.2
      - Conceptual Modeling
      - Architectural Design and Requirements Allocation
        - RD SP 2.2
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

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

- SWEBOK Section 2 presents overall requirements process
- CMMI presents Requirements Engineering in two PAs: RD and REQM
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)
Process Element Example: 
Software Construction

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

<table>
<thead>
<tr>
<th>Process Step</th>
<th>Description</th>
<th>CMMI</th>
<th>SWEBOK Chapter 4 Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use effective methods to implement the design</td>
<td>TS SP 3.1 subpractice 1</td>
<td>1.1; 1.2; 1.3; 2.1</td>
</tr>
<tr>
<td>2</td>
<td>Adhere to applicable standards and criteria</td>
<td>TS SP 3.1 subpractice 2</td>
<td>1.4; 3.2; 3.3; 3.5</td>
</tr>
<tr>
<td>3</td>
<td>Construct software component</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>4</td>
<td>Conduct peer review of the selected components</td>
<td>TS SP 3.1 subpractice 3</td>
<td>3.6</td>
</tr>
<tr>
<td>5</td>
<td>Perform unit testing of the component as appropriate</td>
<td>TS SP 3.1 subpractice 4</td>
<td>3.4</td>
</tr>
<tr>
<td>6</td>
<td>Revise the component as necessary</td>
<td>TS SP 3.1 subpractice 5</td>
<td>none</td>
</tr>
<tr>
<td>7</td>
<td>Integrate separately developed components</td>
<td>PI SP 3.2</td>
<td>3.7</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>SWEBOK</th>
<th>CMMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Software Requirements</td>
<td>• RD, REQM</td>
</tr>
<tr>
<td>• Software Design</td>
<td>• TS</td>
</tr>
<tr>
<td>• Software Construction</td>
<td>• TS – level of detail is less</td>
</tr>
<tr>
<td>• Software Testing</td>
<td>• VER, VAL are much more general</td>
</tr>
<tr>
<td>• Software Maintenance</td>
<td>• Requires interpretation of CMMI PAs</td>
</tr>
<tr>
<td>• Software Configuration Management</td>
<td>• CM</td>
</tr>
<tr>
<td>• Software Engineering Management</td>
<td>• REQM, PP, PMC, SAM (ISM), MA, RSKM, QPM</td>
</tr>
<tr>
<td>• Software Engineering Process</td>
<td>• OPF, OPD, MA, QPM</td>
</tr>
<tr>
<td>• Software Quality</td>
<td>• PPQA, VER, VAL, MA</td>
</tr>
</tbody>
</table>
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)
  - 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

- 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 …”

• 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
References

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