Office of the Secretary of Defense
Software Sustaining Engineering Information Architecture
Information Model - Revision 3.0

13 July 2011

Software Sustainment Engineering Information Architecture

The Software Sustaining Engineering Information Architecture is comprised of a set of integrated information, process, and analysis models that provide an objective structure for characterizing, evaluating, and managing software inventories that are within the sustaining engineering phase of their life cycles. A primary component of the architecture is the information model, which provides a tailorable framework for collecting, correlating, and evaluating software sustainment attribute and performance data. The Software Sustainment Information Model (SSIM) is comprised of two complementary structures. The first is a data taxonomy that provides an outline for identifying and collecting software sustainment attribute data. The second is a performance parameter typology that provides a structure for categorizing and relating those factors that influence and impact the achievement of software sustainment technical and management performance objectives. The data taxonomy and the performance parameter typology together support the characterization of the software sustainment product environment as well as the analysis of observed technical, management, and performance factors and interactions. In addition, the SSIM will drive the design of applicable databases and data management structures for software sustaining engineering data.

Software Sustainment Information Model Assumptions

In the design of the SSIM, the following assumptions are pertinent:

- The specific definitions of a software product and associated attributes are variable across different programs and sustainment organizations.
- Each software product is defined within its own context. Context is largely determined by the nature of the source project and the development/operational environments.
- Software attribute data is collected at the lowest possible level. This is usually at the product level, as defined within an associated organizational context.
- Lower level software attribute data can be aggregated, within defined measurement constraints, at multiple decision levels as applicable. (Low level data should not be aggregated, modified, or “pre-normalized” to an enterprise defined context.)

- Software sustainment data is time dependent. Software sustainment time chronology is tied to the release baselines of defined operational software products.
- Different categories software sustainment attribute data must be correlated - derived from the same product within the same measurement period. Defined operational software releases provide a mechanism to correlate the software attribute data. Many product and environmental (organizational/context) attributes change for a given product from release to release.
- The SSIM is structured to include the following categories of data:
  - Software product attributes and characteristics, including related systems level data.
  - Environmental attributes and characteristics, including organizational characteristics and data.
- The structures of both the SSIM attribute and performance models are intended to be mapped and correlated.

Software Sustainment Information Model - Data Taxonomy

The SSIM-Data Taxonomy (SSIM-DT) is a structured listing of pertinent Software Sustainment attribute data. It is a static information model that identifies the data required to characterize software sustainment products and their associated sustainment environments. The SSIM-DT is applicable to Weapons System, Command and Control System, and Information System software.

The SSIM-DT provides a data structure that supports:

- The characterization and description of software sustainment products and inventories,
- Development and calibration of software sustainment cost estimating relationships (CERs) across a wide scope of programs and products

The SSIM-DT structure is defined as follows:

Software Sustainment Information Model - Data Taxonomy Outline

1.0 Software Sustainment Product

1.1 Program Identification

1.1.1 Program Decision Level (ACAT)

1.1.2 Program Decision Authority

1.1.3 Program Responsibility (OSD/NII/etc.)

1.2 System Identification
- 1.2.1 System of Systems
- 1.2.2 System
  - 1.2.2.1 System element
    - 1.2.2.1.1 System sub-element (component)

1.3 Baseline Identifier
- 1.3.1 Baseline Identifier
  - 1.3.1.1 Initial Product Baseline
  - 1.3.1.2 Incremental Released Baselines (Multiple)
  - 1.3.1.3 Projected Release Baselines (Multiple)
- 1.3.2 Maintenance Stage
- 1.3.3 Design Attributes
- 1.3.4 Technology
- 1.3.5 Product Baseline Change Profile
  - 1.3.5.1 Release Type
  - 1.3.5.2 Release Process
  - 1.3.5.3 Release Content
    - 1.3.5.3.1 Corrective
      - 1.3.5.3.1.1 Defect Type
      - 1.3.5.3.1.2 Priority
    - 1.3.5.3.2 Enhancement
      - 1.3.5.3.2.1 Requirements type
      - 1.3.5.3.2.2 Priority
- 1.3.6 Product Change Backlog
  - 1.3.6.1 Number/size/type of prospective changes
  - 1.3.6.2 Change assessment
  - 1.3.6.3 Investment requirements
  - 1.3.6.4 Priorities
- 1.3.7 Functional Content
- 1.3.8 Resource Profile
  - 1.3.8.1 Budget/Dollars
    - 1.3.8.1.1 Source
    - 1.3.8.1.2 Type
  - 1.3.8.2 People
    - 1.3.8.2.1 Labor
    - 1.3.8.2.2 Effort (By Task)

1.3.8.3 Facilities
1.3.9 Schedule Profile
1.3.10 Work Profile (Applied Tasks/Activities - WBS Allocations)
1.3.10.1 Maintenance - Applied Labor Categories/Source
1.3.10.2 Sustaining Engineering - Applied Labor Categories/Source

1.4 Product Functional Attributes (Each Baseline)
1.4.1 Application Type
1.4.1.1 Weapons
1.4.1.2 Command and Control
1.4.1.3 Information System
1.4.2 Operating environment
1.4.3 Application domain
1.4.4 Criticality
1.4.5 Volatility
1.4.6 User Profile

1.5 Product Design Attributes (Each Baseline)
1.5.1 Platform
1.5.1.1 Host hardware platform/operating system
1.5.1.2 Target hardware platform/operating system
1.5.2 System
1.5.2.1 Architecture
1.5.2.2 Technology (Maturity)
1.5.2.3 Change Drivers
1.5.2.4 External Interfaces
1.5.3 Software
1.5.3.1 Structure
1.5.3.2 Requirements/Function/Type
1.5.3.2.1
1.5.3.3 Language
1.5.3.4 Size
1.5.3.5 Complexity
1.5.3.6 Derivation
1.5.3.6.1 COTS/GOTS
1.5.3.6.2 Reuse
1.5.3.6.3 New
1.5.3.6.4 Modified
1.5.3.7 Age

2.0 Software Sustainment Product Environment

- 2.1 Organizational Definition
  - 2.1.1 Charter/Scope
  - 2.1.2 Structure
    - 2.1.2.1 Internal Organization
    - 2.1.2.2 External Interfaces
    - 2.1.2.3 Decision Structure/Responsibility
  - 2.1.3 Customer Base
- 2.2 Technical Processes
  - 2.2.1 Type
  - 2.2.2 Capability
    - 2.2.2.1 Requirements
    - 2.2.2.2 Design
    - 2.2.2.3 Implementation
    - 2.2.2.4 Integration
    - 2.2.2.5 Test
    - 2.2.2.6 Project management
    - 2.2.2.7 Stakeholder/User Interface
- 2.3 Business Processes
  - 2.3.1 Sustainment strategy
  - 2.3.2 Resourcing strategy (LOE, Product driven, etc.)
  - 2.3.3 Project Management
  - 2.3.4 Information Management/Communications
- 2.4 Resources
  - 2.4.1 Budget/Dollars
    - 2.4.1.1 Source
    - 2.4.1.2 Type
  - 2.4.2 Staff
    - 2.4.2.1 Number
    - 2.4.2.2 Experience/Competency
    - 2.4.2.3 Source (Organic/Contractor)
  - 2.4.3 Facilities
    - 2.4.3.1 Hardware
    - 2.4.3.2 Tools
    - 2.4.3.3 Capacity/Utilization
- 2.5 Work Profile
  - 2.5.1 Volume
  - 2.5.2 Project Diversity/Scope of Work

Software Sustainment Information Model - Performance Typology

The SSIM-Performance Typology (SSIM-PT) is a structured framework used to identify and assess the factors that influence the performance of a software sustainment product and/or organization. The SSIM-PT is a dynamic information model that provides a basis for defining the occurrence and relationships of these factors with respect to stated performance objectives and expected outcomes. (Note: Performance is defined in terms of efficiency, effectiveness, timeliness, and quality.)
The SSIM-PT establishes a many-to-one relationship model that provides a consistent approach for identifying and assessing existing software sustainment performance factors. These factors are of two types:

1. Software sustainment enabling factors: defined as product, process, or environment attributes that enhance the achievement of desired software sustainment outcomes, or

2. Software sustainment product, process, or environment issues: defined as risks, problems, uncertainties, or concerns that constrain performance.

The SSIM-PT assessment issue structure is currently being defined and expanded, and is expected to include the following top-level performance information categories:

- Environment: What is the regulatory, labor, reform, and political environment in which the program operates?

- Mission Requirements: How complex is the development? Can it defeat the threat arrayed against it? Is the operational requirement reasonable? Does the developer have the expertise, plant, and equipment to successfully develop the required system? What are the critical dependencies between other systems?

- Financial: Is funding sufficient, timely, and stable? Is there enough flexibility in funds management to deal with program issues?

- Resources: Do the PM and the developer have the personnel, facilities, tools, and training to complete successful development?

- Management: Do the PM and the developer have the capability to plan, resource, control, and monitor the effort? This includes the acquisition strategy, project planning, contracting and subcontracting, and communication processes.

- Technical Process: Does the developer possess the capability to implement the technical processes needed to manage and conduct the development and ensure process conformance? Is that process appropriate? Is it applied to the program?

- Technical Product: How well do the products and services being produced conform to the requirement? This issue considers product lines, testing requirements, quality, human factors, and safety.

- Schedule: Is the schedule realistic? Does the schedule reflect resource usage and availability? Does the schedule track progress and dependencies?

- User/Customer: Is the end user or customer of the product appropriately supported? This issue includes customer satisfaction and new equipment training or transition support.

- Project Specific: Are there any project-unique issues that cannot be mapped into one of the previous nine issue areas?

- The Program Issue Structure provides a consistent starting point for identifying, assessing, and correlating program issues. An overview of Version 2.3 of the structure is provided in Figure 1, and the full structure is found in Appendix A. It is intended that the structure is augmented and tailored as the program situation, characteristics and issues dictate.

Software Sustainment Information Model - Development Approach

Both components of the SSIM are initially derived from the findings of ongoing U.S. Army, Air Force, and OSD software sustaining engineering analysis and data collection efforts across the DOD program base. These efforts are currently being expanded to

include prime and support contractor software sustaining engineering and maintenance efforts. This will provide a characterization of software sustaining engineering from both the government and industry perspectives. A critical part of this characterization is the collection and structuring of software sustaining engineering planning and performance data. This data will be used to characterize the operational software environments, and to support the development of value added software sustaining engineering measurement processes and cost estimating relationships and models.