2017 S1000D User Forum, Amsterdam

S1000D Requirements for Product Lifecycle Management (*PLM*): *Use Cases, Functional Requirements, and Evaluation Methods*

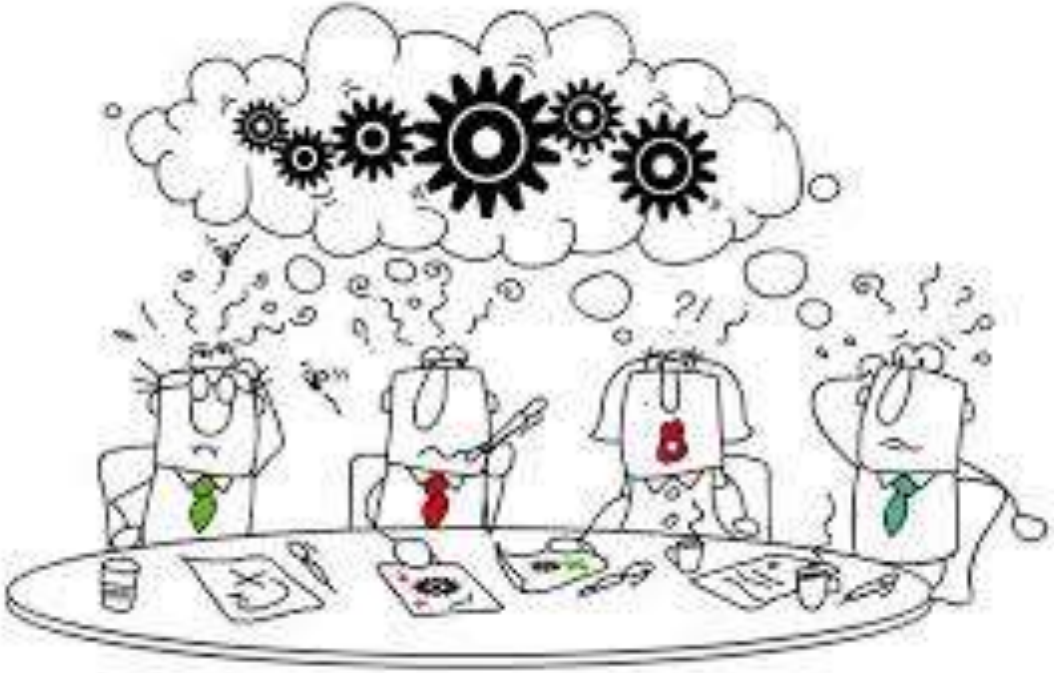
Wayne Gafford:
U.S. NAVSEA

Svante Ericsson:
Psericon

June 13, 2017

Why Are We Here?

Wayne



How to measure the 'Level of Support" for S1000D requirements in a Product Lifecycle Management (PLM) system):

First We Look at - The Product Lifecycle Conceptual Architecture

Second – Analysis Structure for Choosing the Optimal PLM System to Meet Your S1000D Requirements.

Presentation Caveat

Svante

S1000D, technical content management, and publishing are only *several factors* for choosing a PLM system.

Translation:

This presentation is not about how to select a PLM system, but how to factor in optimal S1000D support when evaluating a PLM system.

Other PLM Evaluation Considerations Include:

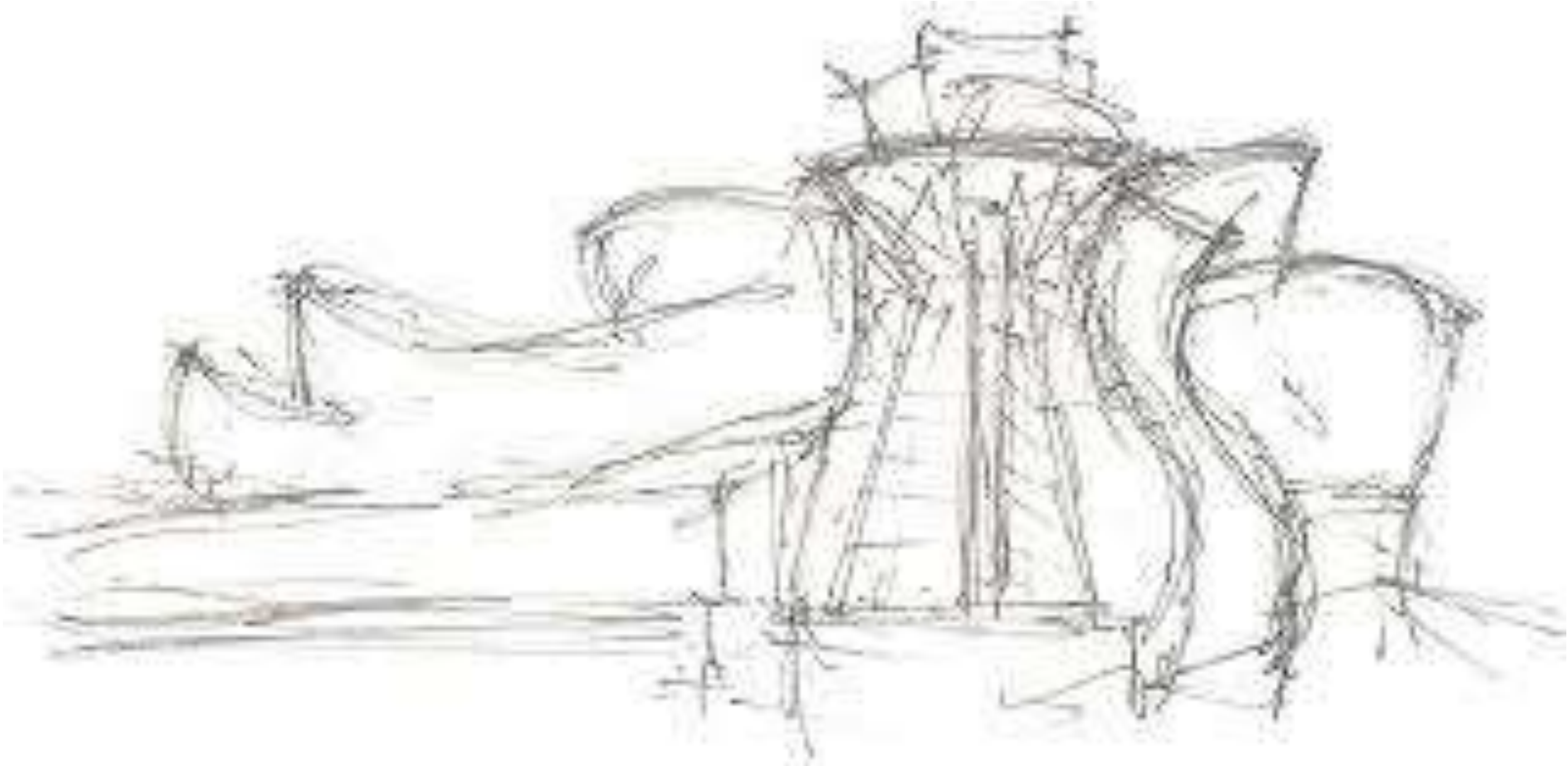
Svante

- Controlling, accessing, and sharing product data
- Tracking engineering product development
- The Digital Thread (*No Orphaned Data*)
- System performance requirements linked to supportability analyses
- Vendor support

Those considerations are for another presentation.

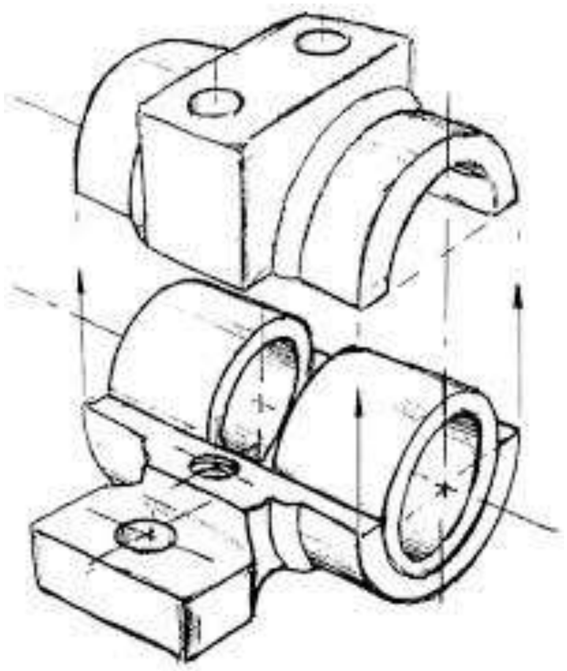
Wayne

Let’s Start By Looking at The PLM Conceptual Architecture



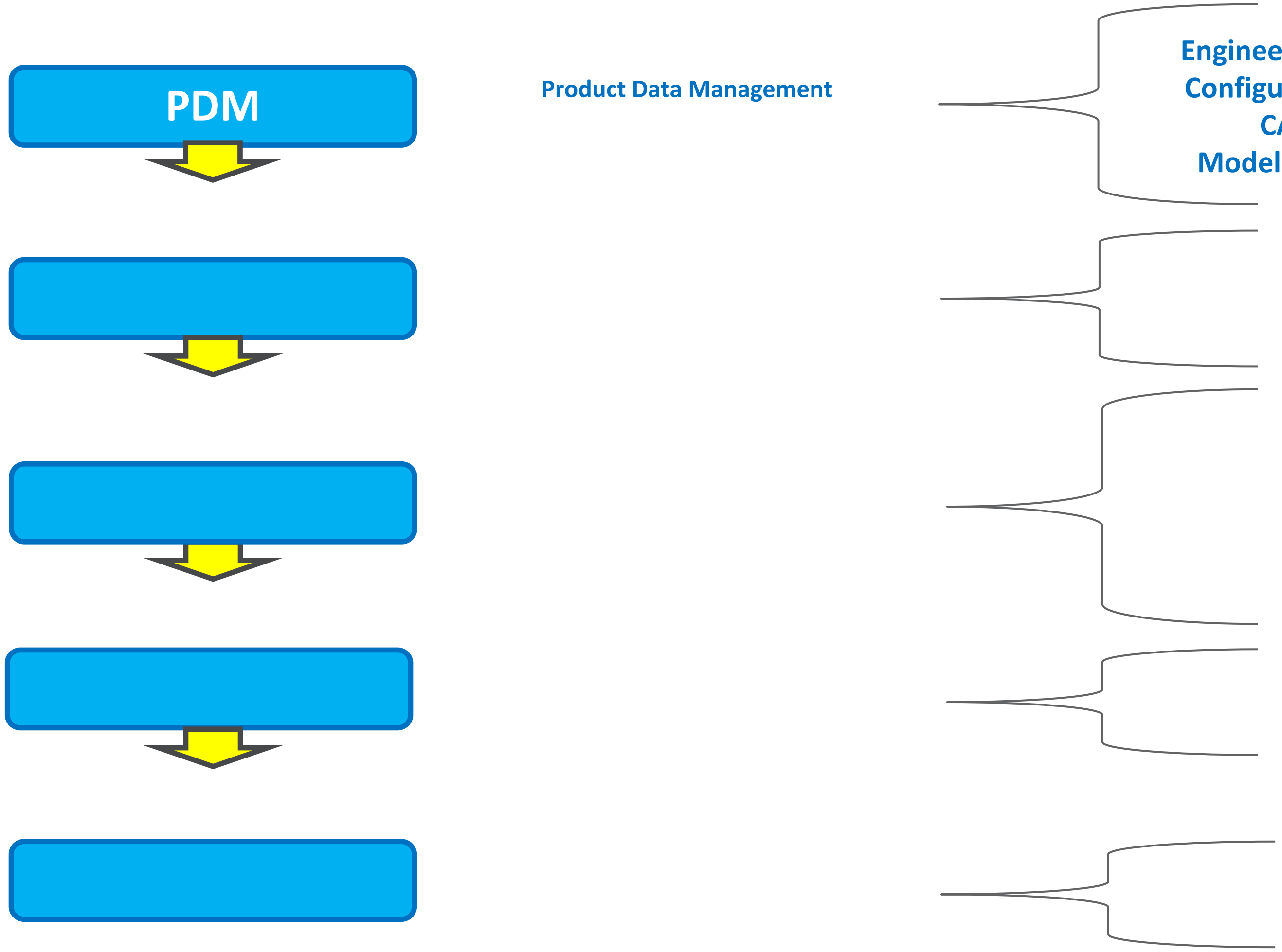
PLM Conceptual Architecture:
Product Data Management Requirements

Wayne



PLM Tools & Components

PLM Functional Activities



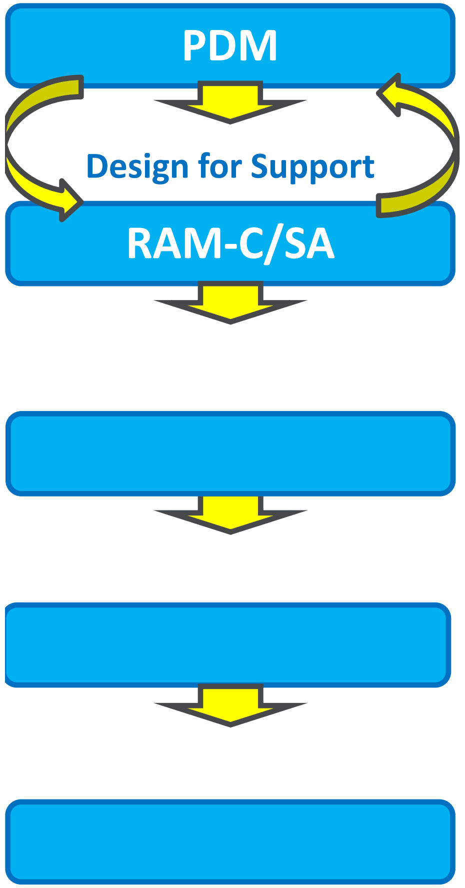
Product Data Management

Engineering Bill of Materials
Configuration Management
CAD Integration
Model-Based Engineering

PLM Conceptual Architecture:
Performance Requirements

Wayne

PLM Tools & Components



Product Data Management

Reliability, Availability, Maintainability, Cost/ Supportability Analysis

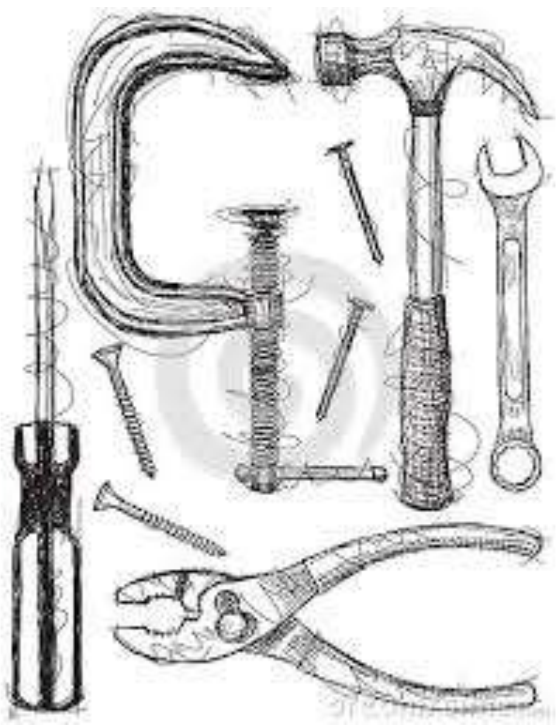
PLM Functional Activities

Engineering Bill of Materials
Configuration Management
CAD Integration
Model-Based Engineering

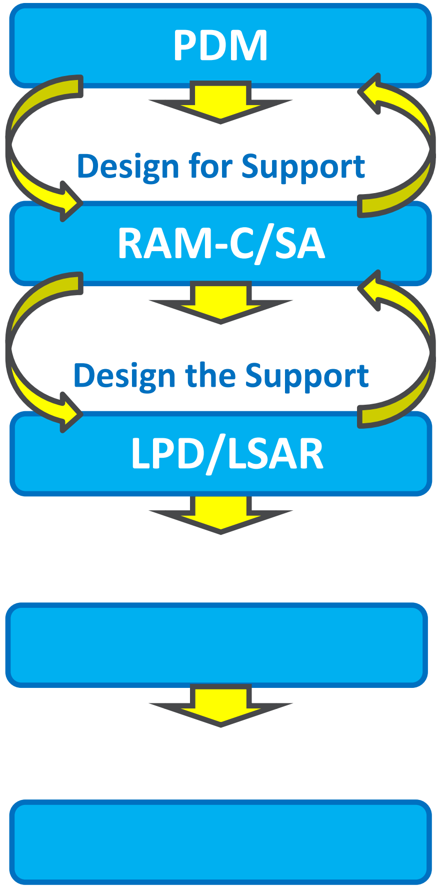
Failure Analysis
Modeling and Simulation
Reliability Block Diagram

PLM Conceptual Architecture:
Supportability Requirements

Wayne



PLM Tools & Components



PLM Functional Activities

Product Data Management

- Engineering Bill of Materials
- Configuration Management
- CAD Integration
- Model-Based Engineering

Reliability, Availability, Maintainability, Cost/ Supportability Analysis

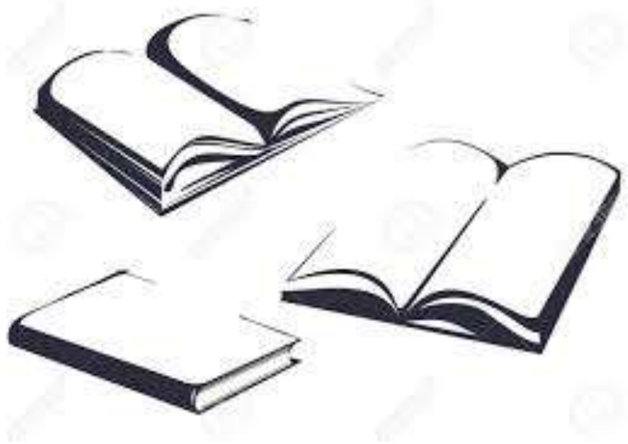
- Failure Analysis
- Modeling and Simulation
- Reliability Block Diagram

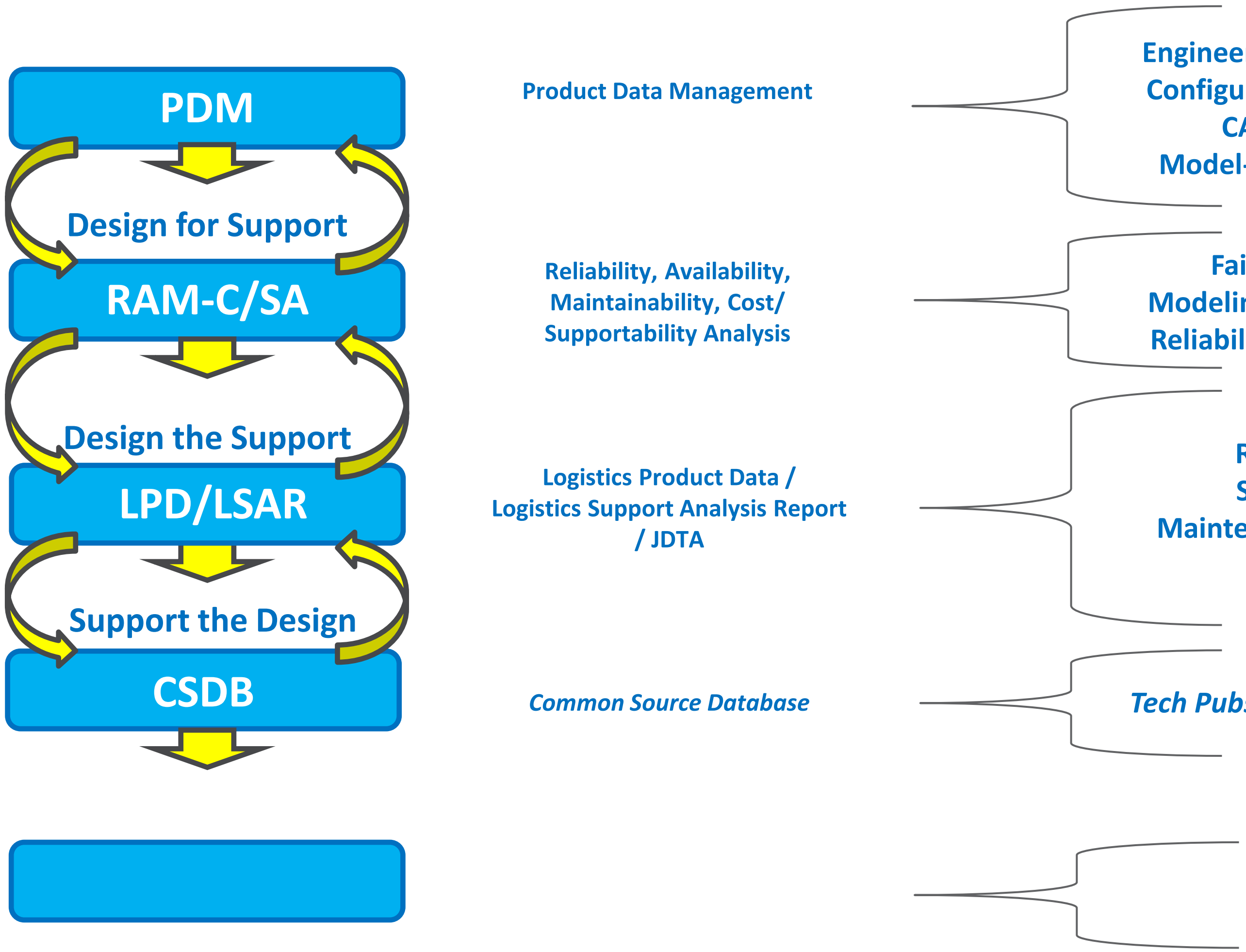
Logistics Product Data / Logistics Support Analysis Report / JDTA

- Readiness Optimization
- Service Bill of Materials
- Maintenance Task Analysis (S3000L)

PLM Conceptual Architecture:
Product Support Requirements

Wayne





PLM Tools & Components

PLM Functional Activities

PDM

Product Data Management

Engineering Bill of Materials
Configuration Management
CAD Integration
Model-Based Engineering

Design for Support

RAM-C/SA

Reliability, Availability,
Maintainability, Cost/
Supportability Analysis

Failure Analysis
Modeling and Simulation
Reliability Block Diagram

Design the Support

LPD/LSAR

Logistics Product Data /
Logistics Support Analysis Report
/ JDTA

Readiness Optimization
Service Bill of Materials
Maintenance Task Analysis (S3000L)

Support the Design

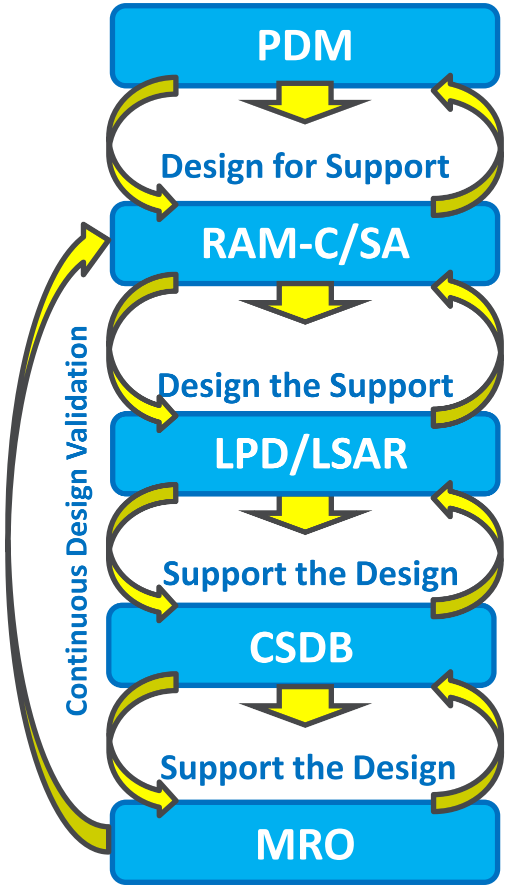
CSDB

Common Source Database

Tech Pubs Dev. (S1000D)

PLM Conceptual Architecture:
Readiness Capture Requirements

Wayne



Continuous Design Validation

PDM

Design for Support

RAM-C/SA

Design the Support

LPD/LSAR

Support the Design

CSDB

Support the Design

MRO

PLM Tools & Components		PLM Functional Activities
PDM	Product Data Management	Engineering Bill of Materials Configuration Management CAD Integration Model-Based Engineering
RAM-C/SA	Reliability, Availability, Maintainability, Cost/ Supportability Analysis	Failure Analysis Modeling and Simulation Reliability Block Diagram
LPD/LSAR	Logistics Product Data / Logistics Support Analysis Report / JDTA	Readiness Optimization Service Bill of Materials Maintenance Task Analysis (S3000L)
CSDB	*Common Source Database*	*Tech Pubs Dev. (S1000D)*
MRO	*Maintenance Repair & Operation*	*Readiness Metrics*

Evaluating The S1000D Thread in PLM:
Three Essential Components to Your Study

Svante

1. S1000D Use Cases

A list of S1000D-based events defining the interactions between a role and a system to achieve a goal.

2. S1000D Functional Requirements

A list of inputs and outputs within an event to enable a S1000D use case.

3. S1000D Evaluation Methodology

A tool to help understand the level of support for a S1000D functional requirement.

S1000D Use Cases for PLM Evaluation

Svante

Our S1000D Use Cases Divided Into Two Categories

- **Functional Use Cases**

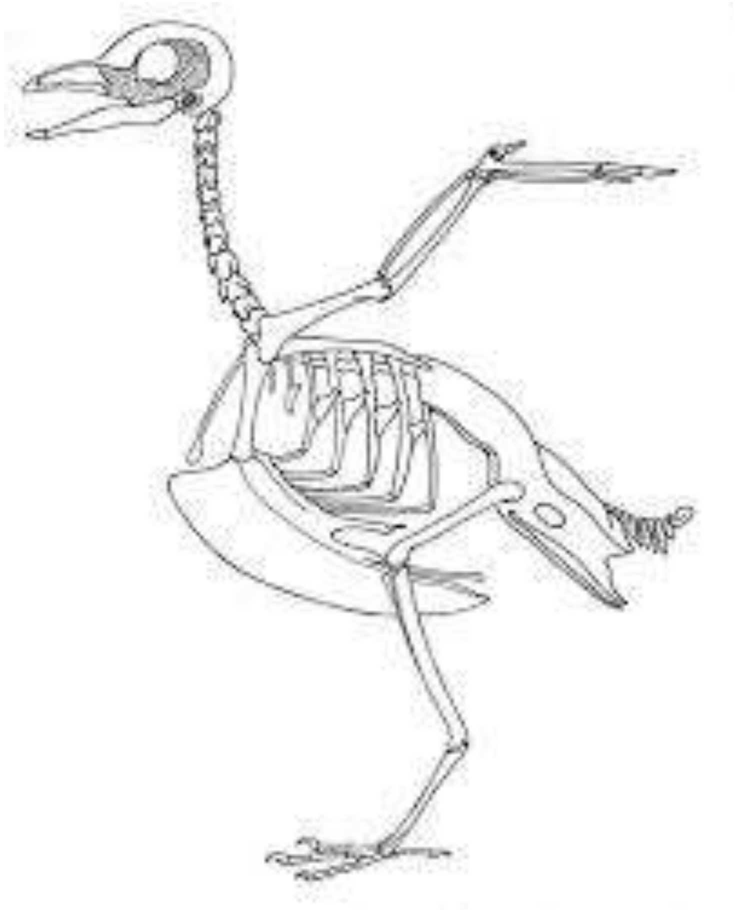
  PLM processing events that build data modules and create traceability.

- **Informational Use Cases**

  PLM processing events that collect, structure, and output information from data modules into a published form.

S1000D Functional Use Cases

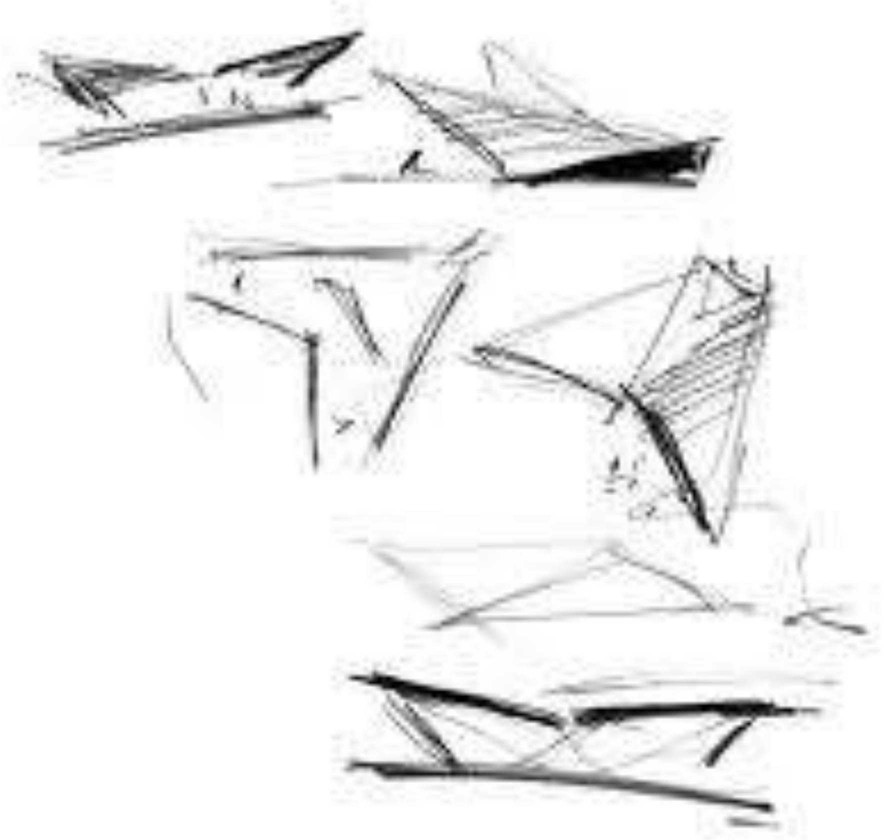
Svante



- DMRL: Create Data Module Requirements List (DMRL) in PLM
- Templates: Automatically build publication and data module templates in CSDB
- Changes: Trace system design changes to impacted CSDB Objects
- Faults: Identify Fault Modes and Fault Modes Resolution in CSDB Objects

S1000D Informational Use Cases

Svante



- Create a Parts Lists
- Create Operating Instructions
- Create Maintenance Instructions
- Create Human Competency Models and Technical Learning Content

Use Case Structure: Profile Example

Svante

Use Case Name: *Identify Fault Modes and Resolution in CSDB Objects*

Integrated Product Support Element : Sustaining Engineering, Technical Data

Actors: Manufacturer, Maintenance Engineer, Maintainer, Logistics Management Specialist, Tech Writers, PLM

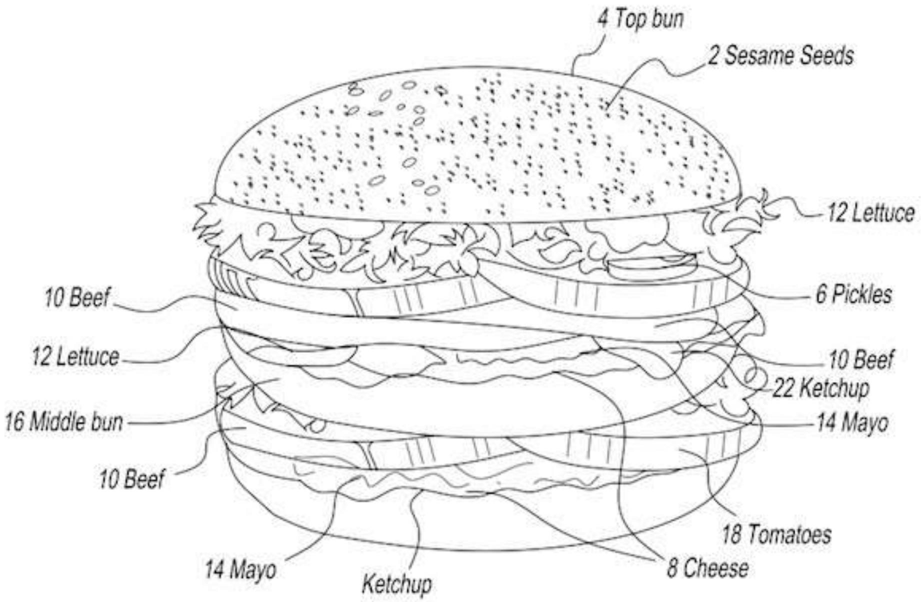
Goal: Effectively troubleshoot, identify failure modes and system failures in CSDB Objects.

Pre-condition: Operational checklist, fault codes identified, troubleshooting procedures.

Post-condition: Functional Link exist between fault modes and CSDB objects

S1000D Functional Requirements and Desired Data Exchanges

Wayne



Every Use Case Needs Functional Requirements

(Functional requirements lead to software code)

Functional requirements represent the *desired data exchange* between engineering data, S1000D technical data, and end user activities

Use Case + Functional Requirements = Good Recipe

Wayne

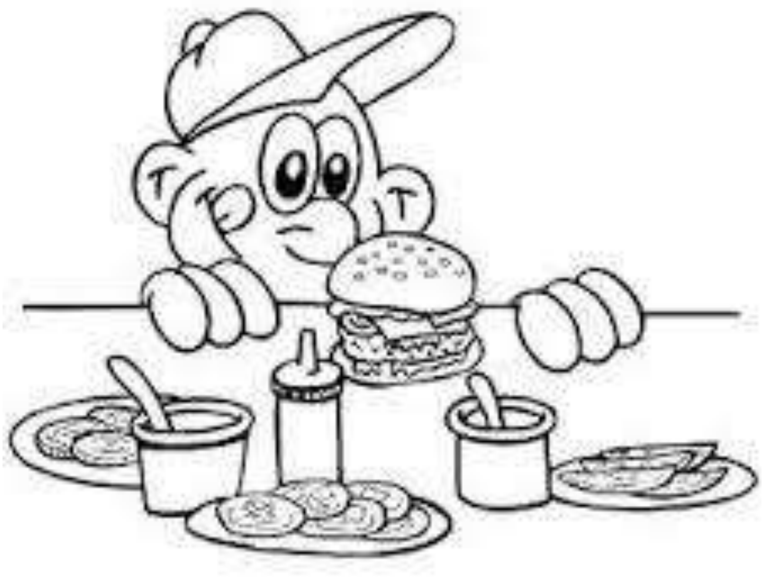
FOUR FUNCTIONAL S1000D USE CASES *WITH* FUNCTIONAL REQUIREMENTS

Functional Use Case ONE *With* Functional Requirements

Wayne

Use Case Name: Create Data Module Requirements List (DMRL) in PLM.

Functional Requirements



MAP AND POPULATE: The system shall fully map to and populate all structures, including contents of the Data Module Code element (<DMCODE>), in an S1000D data module requirements list on a per-project basis.

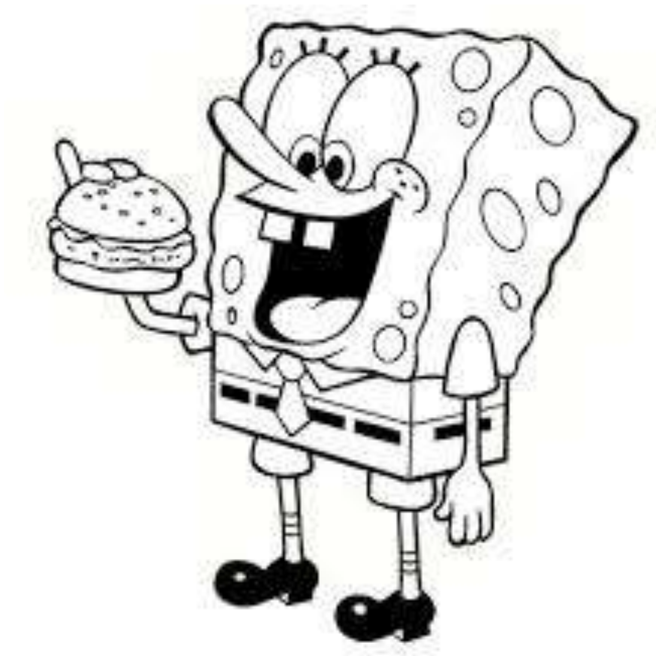
IDENTIFY SNS FROM CHAP 8: The system shall identify top-down breakdown structures of a system and disassembly data using standard numbering systems documented in chapter 8 of the S1000D specification.

Functional Use Case *TWO With* Functional Requirements

Wayne

Use Case Name: Automatically build publication and data module templates in CSDB

Functional Requirements



EXPORT TEMPLATES: The system shall support the ability to export S1000D publication and data module templates to a CSDB.

POPULATE PM CODES: The system shall populate the publication module template with publication module codes.

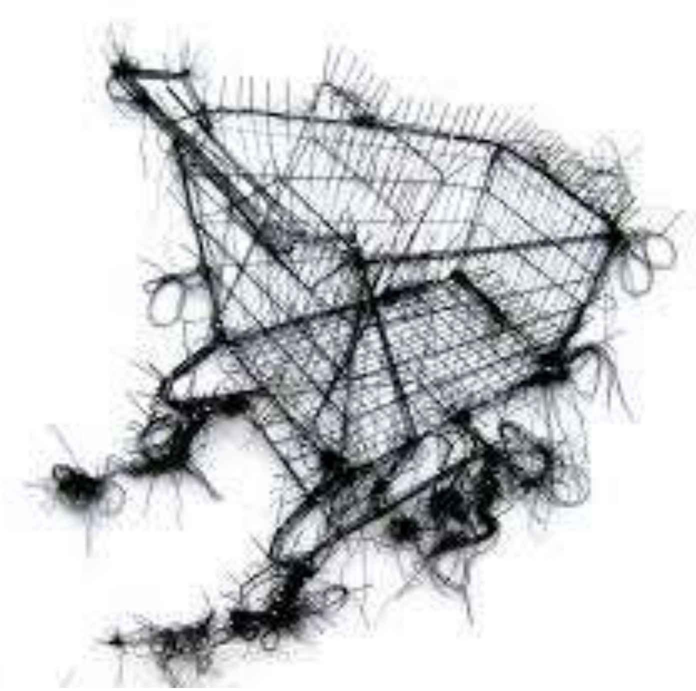
POPULATE DM CODES: The system shall populate the data module templates with defined data module codes.

POPULATE IDENT & STATUS: The system shall identify, map and populate the Identification and Status portion of the templates with appropriate project metadata.

Functional Use Case THREE *With* Functional Requirements

Wayne

Use Case Name: Trace system design changes to impacted CSDB Objects



Functional Requirements

IDENTIFY CSDB OBJECTS: The system shall identify all CSDB objects impacted by system, parts, assemblies and configuration changes.

ESTABLISH APPLICABILITY: The system shall support the means to establish applicability to CSDB objects.

Functional Use Case Four *With* Functional Requirements

Wayne

Use Case Name: *Identify Fault Modes and Resolution in CSDB Objects*

Functional Requirements

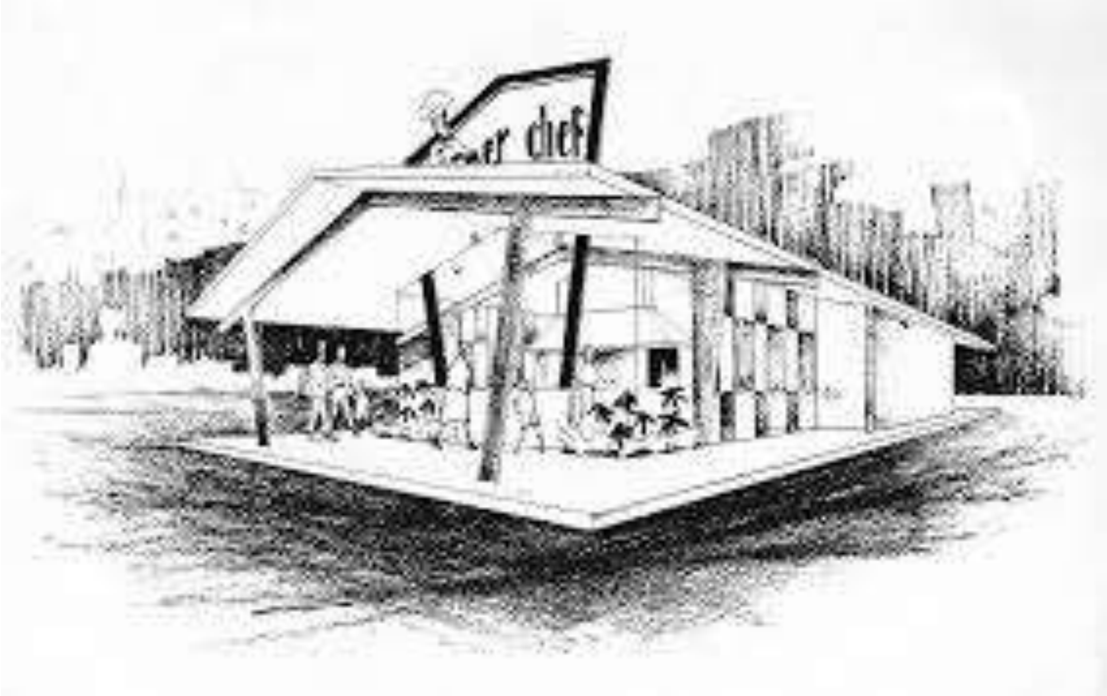
ASSIGN FAULT CODES: The system shall assign a fault code to each failure mode and populate the appropriate CSDB object.

ORGANIZE FAILURE MODES: The system shall organize the failure modes by separate systems, subsystems, items, parts, assemblies or process steps in the appropriate CSDB objects.

MAINTAIN LINKS: The system shall maintain links between failure modes in PLM and CSDB objects over the life of the system.

Use Case + Functional Requirements = Good Recipe

Svante



FOUR INFORMATIONAL S1000D USE CASES *WITH* FUNCTIONAL REQUIREMENTS

Informational Use Case One *With* Functional Requirements

Svante

Use Case Name: Create a Parts Lists

Functional Requirements

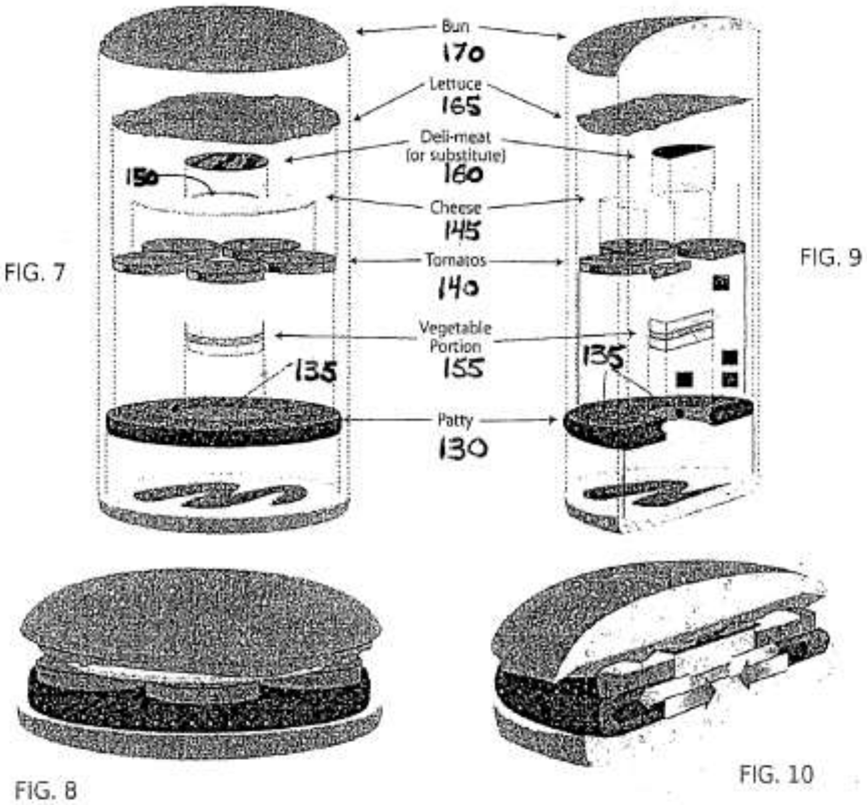


FIG. 7

FIG. 9

FIG. 8

FIG. 10

PARTS *TO* DMs: The system shall support the ability to develop, maintain and export a parts list to the appropriate S1000D CSDB objects.

PARTS *TO* GRAPHICS: The system shall support the association of parts lists and part list items with the appropriate interactive graphics.

PARTS *IN* GRAPHICS: The system shall export interactive graphics associated with parts lists and part list items to the appropriate CSDB object.

Informational Use Case Two *With* Functional Requirements

Svante

Use Case Name: Create Operating Instructions

Functional Requirements

DEVELOPMENT: The system shall support the development of system operating instructions based on project content requirements in S1000D CSDB objects.

LINKING: The system shall maintain links between operating instructions in CSDB objects and related data fields in a Product.

Informational Use Case Three *With* Functional Requirements

Svante

Use Case Name: Create Maintenance Instructions

Functional Requirements

DEVELOPMENT: The system shall support the development of system maintenance instructions based on project content requirements in S1000D CSDB objects.

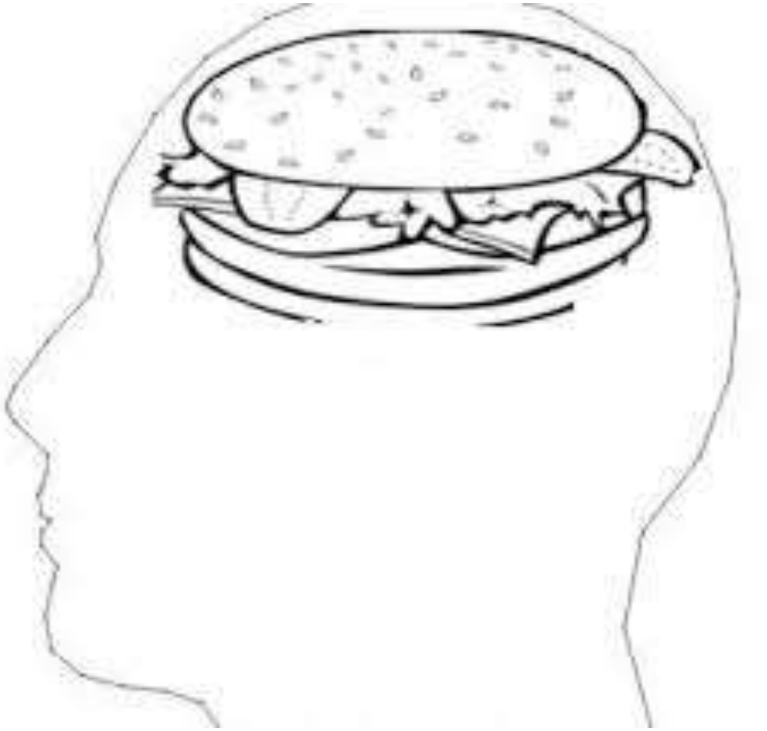
CORRECTIVE ACTION: The system shall support the development of procedures for identifying faults that lead to a corrective action, such as a maintenance or troubleshooting procedure, in a CSDB object.

LINKING: The system shall maintain links between maintenance instructions in CSDB objects and related data fields in ePLM.

Informational Use Case Four *With* Functional Requirements

Use Case Name: Create Human Competency Models and Technical Learning Content

Wayne



Functional Requirements

DEVELOP COMPETENCIES: The system shall support the development of human competency models.

ALIGN COMPETENCIES: The system shall align human competency models to system performance and supportability requirements.

DEVELOP CURRICULUM: The system shall support the development of learning objectives, course planning, training information, assessments and human performance content in S1000D CSDB objects.

LINK COMPENTENCIES TO CURRICULUM: The system shall link human competency models to learning, training and human performance data modules.

LINK ALL CURRICULUM TO SYSTEM OBJECTS: The system shall maintain links between learning and training content in CSDB objects and related data fields in the Product Lifecycle Management (PLM) solution.

EVALUATION METHODOLOGY

Wayne

USE CASES

+

FUNCTIONAL REQUIREMENTS

+

EVALUATION METHODOLOGY

=

S1000D SUPPORT DECISION

Basis for Evaluation

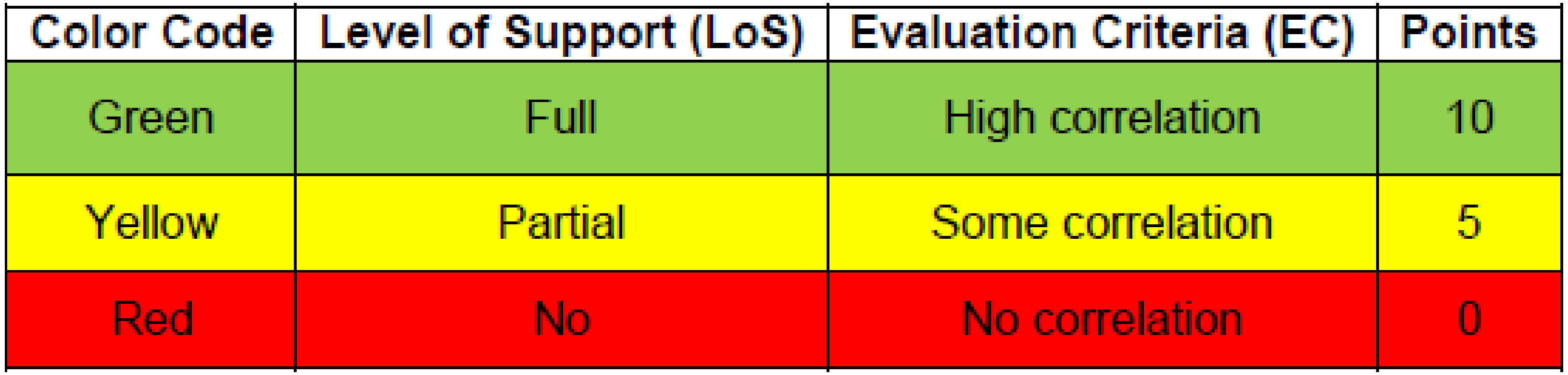
Wayne

Written Responses – ***How does your product support each functional requirement?***

Demonstrations – ***Show how your product support each functional requirement?***

Evaluation Methodology Scoring

Wayne



Color Code	Level of Support (LoS)	Evaluation Criteria (EC)	Points
Green	Full	High correlation	10
Yellow	Partial	Some correlation	5
Red	No	No correlation	0

Level of Support Analysis

Each *Functional Requirement* is Scored. It is Given:

1. **A Color**
2. **Points**
3. **Evaluation Criteria**

The system *fully, partially, or does not* support the *desired data exchange between engineering data and S1000D technical data* expressed in a use case and in its functional requirements.

Level of Support Scoring Calculation: Total Functional Requirement Score (TFRS)

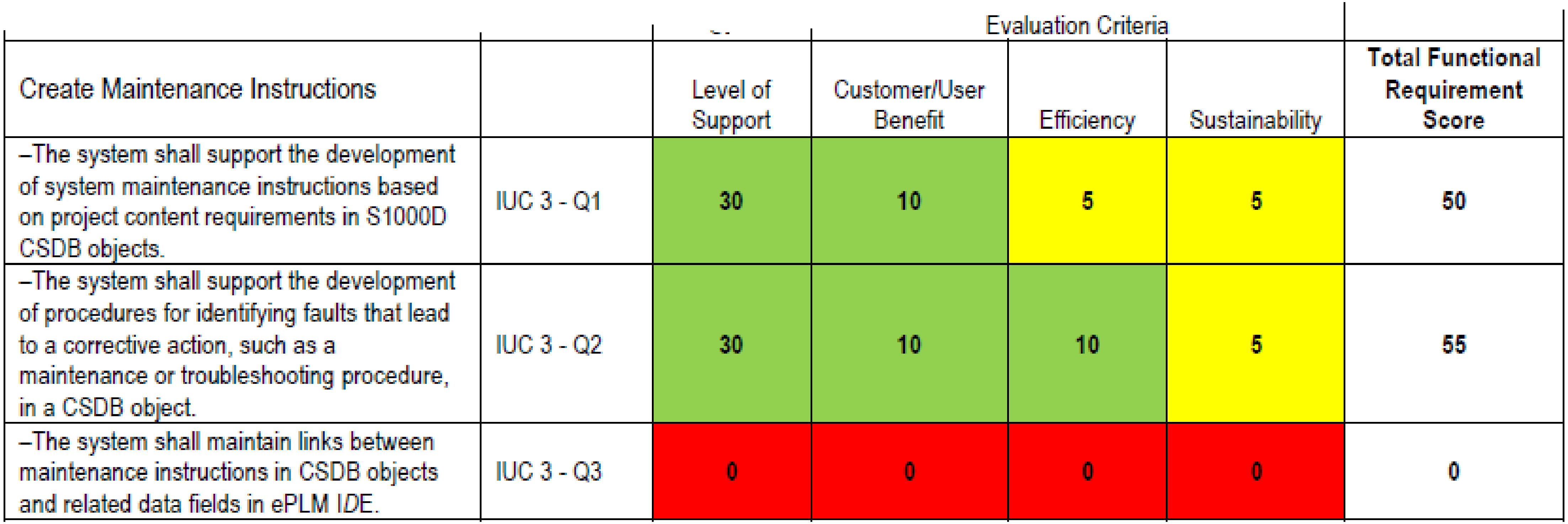
The **Total Functional Requirement Score (TFRS)** is the ***Level of Support (LoS)*** multiplied by a ***weight of three*** plus the sum of the three ***Evaluation Criteria (EC) scores***. The weight is calculated by adding the quantity of ECs. Therefore, the weight is three because there are three ECs.

Wayne

TFRS = (LoS*3) + (EC1+EC2+EC3)

Total Possible TFRS Score: 60

Total Possible Grand Score (sum of all 24 TFRS): 1440



			Evaluation Criteria			
Create Maintenance Instructions		Level of Support	Customer/User Benefit	Efficiency	Sustainability	**Total Functional Requirement Score**
–The system shall support the development of system maintenance instructions based on project content requirements in S1000D CSDB objects.	IUC 3 - Q1	**30**	**10**	**5**	**5**	**50**
–The system shall support the development of procedures for identifying faults that lead to a corrective action, such as a maintenance or troubleshooting procedure, in a CSDB object.	IUC 3 - Q2	**30**	**10**	**10**	**5**	**55**
–The system shall maintain links between maintenance instructions in CSDB objects and related data fields in ePLM IDE.	IUC 3 - Q3	**0**	**0**	**0**	**0**	**0**

Evaluation Criteria Definitions

Wayne

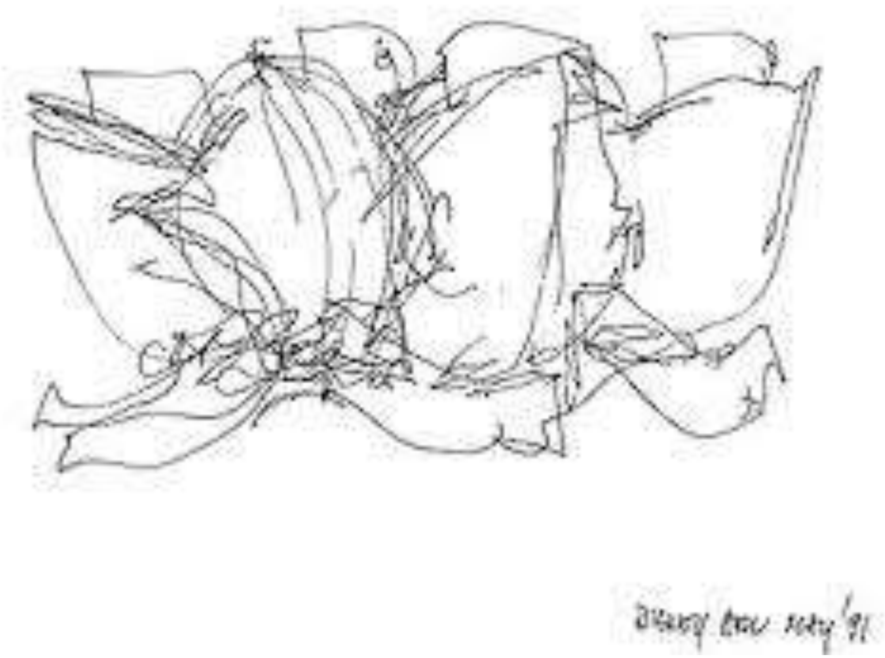
CUSTOMER OR USER BENEFIT – Ease of use, accessibility, how well the item fulfills the use case requirement for a user.

EFFICIENCY – Consideration for the operational footprint, power and required technical personnel including personnel training. How well (automatically, partially, or not at all) the PLM deposits upstream engineering data into an S1000D data module stored in the CSDB.

SUSTAINABILITY – Initial and lifecycle system costs. The forecast for product improvements via expected technology refresh cycle times. Whether the capability is available in the baseline system or requires a separate license. Projected software maintenance costs.

Document Structure

Wayne



EXECUTIVE SUMMARY

Feasibility Study Participants

1 Background

2 Purpose and Scope

3 Feasibility Study Team

4 Use Cases

5 Conceptual Process Architecture

6 Assumptions

7 Constraints

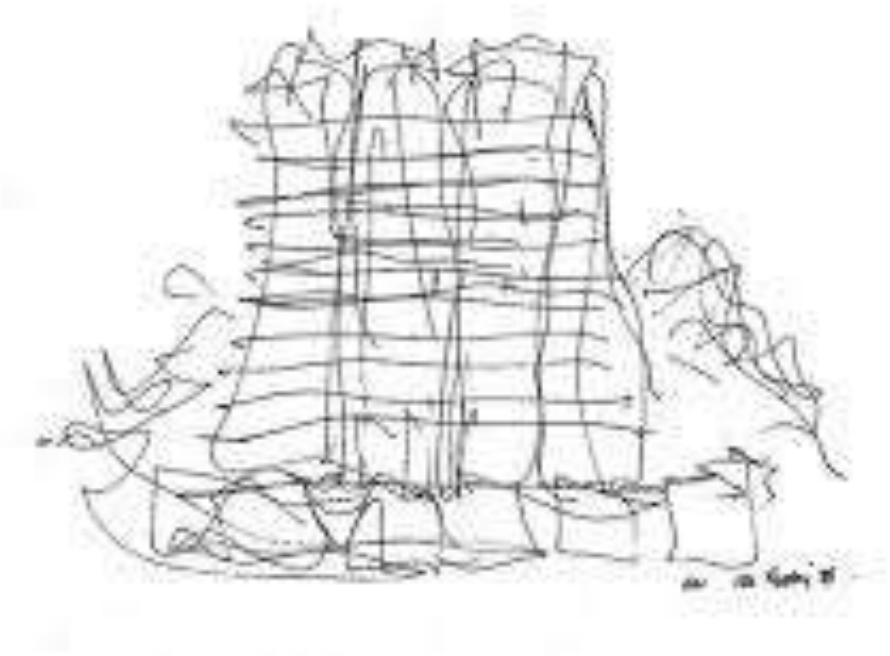
8 Evaluation Methodology

8.1 Evaluation Methodology Development Process

8.2 Apply Evaluation Method and Calculate LoS Scores

Document Structure

Wayne



9 Courses of Action (COA)

9.1 COA #1

9.1.1. Evaluation for COA #1

COA #1, Functional Use Case #1

COA #1, Functional Use Case #2

COA #1, Functional Use Case #3

COA #1, Functional Use Case #4

COA #1, Informational Use Case #1

COA #1, Informational Use Case #2

COA #1, Informational Use Case #3

COA #1, Informational Use Case #4

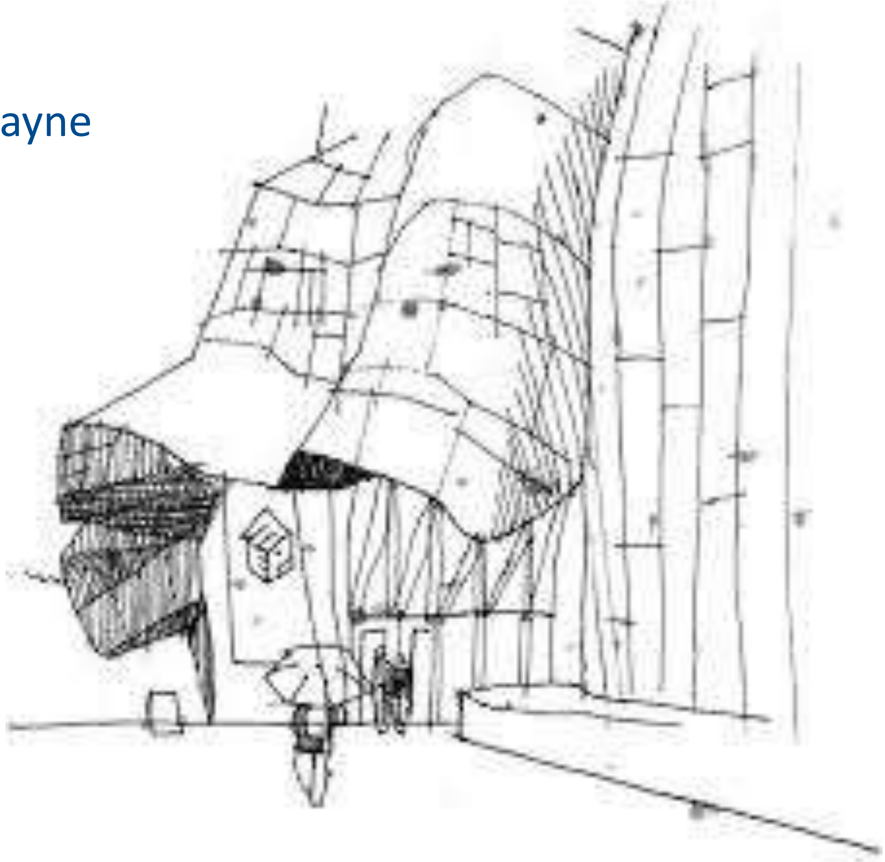
9.2 COA #2 9.2.1 Evaluation for COA #2

COA #2, Functional Use Case #1

ETC…

Document Structure

Wayne



10 Evaluation Summary

10.1 Scoring Summary for COA #1

10.2 Scoring Summary for COA #2

10.3 Clarification of Assumptions

11 Recommendation, Conclusion, and Way Ahead

APPENDICES

APPENDIX A - Acronyms

APPENDIX B - S1000D-PLM Use Cases and Functional Requirements for Level of Support

APPENDIX C - Consolidated Evaluation Scores

APPENDIX D - COA Responses (Limited Distribution)

Conclusion and Summary

Measure Level of Support for the desired data exchange in PLM between 1) engineering data, 2) S1000D technical data, and 3) end user activities.

Wayne/Svante

Use Cases

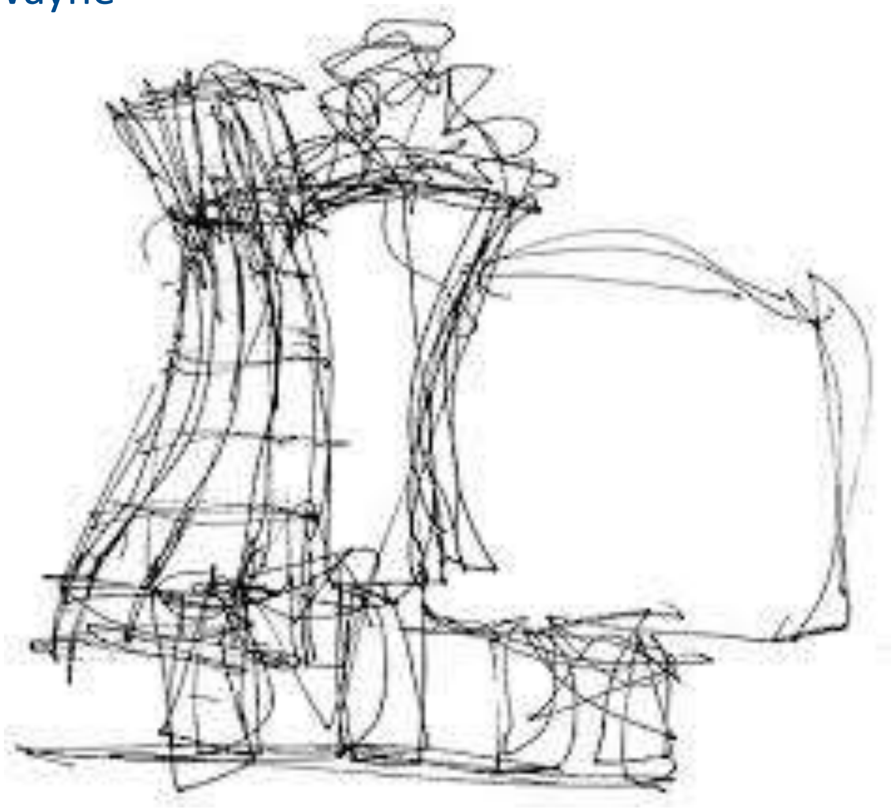
Functional Requirements

Evaluation Methodology

Document Structure

Document Structure

Wayne



List of Tables

List of Figures

2017 S1000D User Forum, Amsterdam

S1000D Requirements for
Product Lifecycle Management (*PLM*):
Use Cases, Functional Requirements, and
Evaluation Methods

Wayne Gafford, U.S. NAVSEA – wayne.gafford@navy.mil
Svante Ericsson, Psericon – svante@psericon.com