



PLM Interest Group

PLM-ALM Standard and Reference Manual

Development of hybrid hardware/software products requires the integration of PLM and ALM. Modern "black box" software can fail to do this correctly.

This Standard shows PLM Managers the level of integration they must achieve; and is supported by a full technical analysis in the Reference Manual.

Version 3.0E

0 Executive Summary

The development and management of hybrid software/hardware products through their lifecycle has always been a complex problem.

This subject is now known as Application Lifecycle Management (ALM), but was originally referred to as Software Configuration Management (SCM).

Terminology has moved forward but solutions have not. Some vendors have provided applications that can link software development to PLM, but they tend to be "black box" solutions that hide their workings from the user.

What is worse for the PLM Team is that these applications focus on synchronising product development as it initially happens - in the "here and now". But PLM also has a requirement to maintain the complete product configuration as it existed at defined points in the past. Lack of attention to this stores up potentially serious problems for the future.

Hardware design and software design are different activities, and PDM and ALM are different systems. The most effective way of operating them together is to use the systems for the purposes for which they were intended, and to maintain a simple but rigorous interface between the two.

This document is based on a workshop run by the PLMIG in Gothenburg in 2005, when the subject was at the forefront of PLM debate. Participants included industrial companies, vendors of PLM and SCM solutions, and academic experts from the University of Mälardalen, who were the recognised leaders in the field.

The workshop produced an explicit methodology for the best way to manage the product data between Product Data Management and Software Configuration Management systems. This document brings everything up to date with a Standard, to show what should be done: and a Reference Manual to show why.

The [PLM-ALM Standard](#) enables the PLM Team to test that everything is set up correctly. It defines the requirements that the PLM implementation must meet if it is to successfully manage the software and hardware product lifecycle.

This document also contains a [PLM-ALM Reference Manual](#) that contains all of the theory and learnings from Gothenburg; and explains this logic in a way that any PLM Manager can understand. Its methodology removes the uncertainties about the best way to manage PLM and ALM data, and allows you to implement a single, practical solution with the minimum of time and effort.

Table of Contents

0	Executive Summary	2
PLM-ALM Standard		6
1	About This Standard	7
	1.1 Scope and Application.....	7
	1.2 History	7
	1.3 Revision Status.....	7
	1.4 Normative References	7
	1.5 Editing the Standard	8
	1.6 Printing the Standard.....	8
	1.7 PLMIG Q&A.....	8
2	Terminology	9
	2.1 PLM vs PDM.....	9
	2.2 ALM vs SCM.....	9
	2.3 PDM-SCM.....	9
3	Background.....	10
	3.1 Context of PLM-ALM.....	10
	3.2 The Importance of PLM	10
	3.3 The Purpose of this Standard	10
4	The PLM-ALM Standard	11
	4.1 Concurrent Integration	11
	4.2 Historical Integration.....	11
5	Applying the PLM-ALM Standard	12
	5.1 Holistic View	12
	5.2 PLM Application.....	12
	5.3 ALM Application.....	12
	5.4 Compliance	12
	5.5 Review and Further Action.....	13

PLM-ALM Reference Manual	14
6 Overview: The PDM-SCM Problem.....	15
6.1 What Is The Problem?.....	15
6.2 Why Is The Problem Misunderstood?.....	15
6.3 Underlying Differences Between Hardware and Software.....	16
6.4 Underlying Differences Between PDM and SCM.....	16
6.5 How PDM and SCM Can Be Integrated.....	17
7 What We Are Aiming For.....	18
8 Scope.....	19
8.1 PLM vs PDM.....	19
8.2 Product Hardware/Software Matrix.....	19
9 Definitions.....	20
9.1 Hardware vs Software.....	20
9.2 What is the Product?.....	20
9.3 Products vs Services.....	21
10 Principles and Assumptions.....	22
10.1 Discipline.....	22
10.2 The Importance of Terminology.....	22
10.2.1 Hardware vs Software Terminology.....	22
10.2.2 System Terminology: Baseline.....	22
10.2.3 Organisational Terminology.....	22
10.3 Processes vs Lifecycles.....	22
10.4 Not a Culture Project.....	23
10.5 Not a “Unified Theory”.....	23
11 Why Hardware and Software Environments are Different.....	24
11.1 Introduction.....	24
11.2 Different Product Components.....	24
11.3 Different Tasks.....	24
11.4 Different Lifecycles.....	25
11.5 Different Processes.....	26
11.6 Different Revision Effectivity.....	27
11.7 Different Roles.....	28
11.8 Different Tools.....	29
11.9 Different Data Management Needs.....	29

12	The Logical Solution.....	30
12.1	Theoretical Lifecycle.....	30
12.2	Decision Process	30
12.3	Design Process	31
12.4	PDM-SCM Process: Scope.....	32
12.5	PDM-SCM Process: Overview	33
12.6	PDM-SCM Process: Detail.....	34
12.6.1	PDM Process.....	34
12.6.2	SCM Process.....	35
12.7	Integrated PDM-SCM	35
12.7.1	Principles	35
12.7.2	Process in Detail - Graphic	36
12.7.3	Process in Detail - Description.....	37
12.8	Engineering Change	40
12.9	ERP	41
12.9.1	Pre-Production Release.....	41
12.9.2	Production Release	41
12.10	Archiving.....	41
13	The Practical Solution: How to Apply the Methodology	42
13.1	Overview	42
13.2	How to Use the Reference Manual	43
13.2.1	Understand the Theory	43
13.2.2	Back to Basics.....	43
13.2.3	Compare the Differences	43
13.2.4	Communication.....	43
13.2.5	Planning.....	43
13.3	Implementation	43
Annexes		44
14	Annexes	45
14.1	Academic and Logical Theory.....	45
14.1.1	Underlying Theory	45
14.1.2	Workshop Approach.....	45
14.1.3	Future Research.....	45
15	Terms and Conditions	46
15.1	Standard Ownership and Use.....	46
15.2	Reference Manual Ownership and Use	46
15.3	Disclaimer.....	47