



**HAL**  
open science

# Set Based PLM Implementation, a Modular Approach to PLM Process Knowledge, Management and Automation

Bas Koomen

► **To cite this version:**

Bas Koomen. Set Based PLM Implementation, a Modular Approach to PLM Process Knowledge, Management and Automation. 14th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2017, Seville, Spain. pp.3-12, 10.1007/978-3-319-72905-3\_1 . hal-01764158

**HAL Id: hal-01764158**

**<https://inria.hal.science/hal-01764158>**

Submitted on 11 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Set based PLM implementation, a modular approach to PLM process knowledge, management and automation.

Bas Koomen

Cadmes b.v., 's-Hertogenbosch, The Netherlands

[bas@cadmes.com](mailto:bas@cadmes.com)

**Abstract** In many cases PLM implementations are halted in the first phases of larger projects. On average, implementation projects take longer, cost more than planned and not all goals are achieved despite modern software implementation methods like Agile or Scrum. This paper proposes another approach, in which the implementation method is inspired by product development methods in general and set based concurrent engineering in particular. The method is structured in five major steps alongside a method of knowledge management and reuse to support the implementation method. The five steps deal with scope and maturity level, requirements analysis, process mapping, rationale based solution selection and system consolidation. The element of knowledge reuse makes this method also accessible for small and medium sized companies, generally reluctant to conduct a fundamental process analysis before starting a software implementation. From there this knowledge can evolve towards a product configuration framework for PLM implementation. The paper outlines the method in theory and proposes further steps to investigate each step in more detailed research and case studies.

**Keywords.** PLM implementation · Knowledge management · PLM process management · SME.

## 1 Introduction

Over 15 years of field experience in industry have led to the realization that often organizations regard PLM mainly as software that should solve their business challenges. They rely on expertise from vendors or external experts from who they expect to explain to them what is the best software to select and how PLM should be implemented. First, they start with software implementation and then, during the project, they find out that they actually needed a definition of processes, procedures,

structures, strategies, etc. after all and then need to retrace their steps in the project to save the investment made. This is especially the case with SME organizations where a high amount of time and money is lost on problematic implementations.

Research on PLM maturity level [1] shows that many organizations do not evolve beyond basic level of PDM (Product Data Management). In literature, different approaches have been laid out to overcome this problem. Section 2 elaborates more on this.

A number of organizations that have been worked with are manufacturers of machinery. These organizations are working in an engineering-to-order structure, but are looking to change to a configure-to-order structure. At some point in time, around 2014, the idea came up that at a high level of abstraction, PDM/PLM software implementations could be regarded the same way. A first literature review was performed to look for existing research on this idea of which the most important findings are described in section 2.

From literature review and insights from actual projects, a framework was defined for a new method to implement PLM, optimized for knowledge reuse and therefore suitable for SME. The framework is a middle way between a “blank canvas” PLM implementation and a fully standardized “off-the-shelf” or “industry template” implementation. The content of this framework is described in section 3.

Before it can be confirmed or denied that this framework indeed can lead to a better world for PLM in SME, more research has to be performed, as further described in section 4.

## 2 Background/related work

### 2.1 PLM

**Holistic PLM.** This paper builds on the concept of holistic PLM. In this concept, PLM is not just the software that is referred to as PLM software, but the combination of strategy, culture and people, structures, process and interaction patterns and IT architecture.

Processes and interaction patterns are defined by strategy, supported by (product and knowledge) structure and they use culture, people and IT architecture as a resource [2]. This implies that an implementation of PLM includes a transformation or at least some kind of assessment or formalization of all elements in the holistic concept of PLM. Also the IT infrastructure includes all software, hardware and network infrastructure relevant to the processes and interaction patterns.

The benefits of PLM are not discussed in this paper and assumed as significant for most organizations.

**Implementation.** Literature exists about different approaches to improve the implementation strategy for SME. Batenburg et al. [1] point out the importance of

maturity level before implementing software and how to measure it with an alignment framework. Also they propose to do a step-wise implementation instead of a full scope implementation at once. What is not covered is how to achieve the desired maturity level.

Silventoinen et al. [3] propose a roadmap approach where the emphasis is on growing awareness of the benefits of PLM. From there a company derives motivation to implement processes and software. They also recognize the need for a certain process maturity level in order to be successful.

Schuh et al. [4] have defined a process framework with macro level process definitions for industries and the corresponding PLM software requirements. This research mainly looks for a match between macro level processes and reference models for IT-systems serving these processes.

Navarro et al. [5] have investigated the application of lean principles in PLM implementation projects. They suggest that when lean principles (customer value, creating flow, continuous improvement) are applied, PLM implementation challenges could be minimized.

Bokinge and Malmqvist have done a deeper review of existing literature on PLM implementation guidelines [6]. They have summarized it into twenty guidelines in four categories (Project process, goals, system and process design, organization). Main conclusion is that the pre-work before installing software is at least as or more important than the technical implementation.

**Maturity level.** Another area of research that has been looked into is PLM maturity level [1-3] and organization capability level. Two conclusions can be drawn from this research: 1) Most surveyed companies do not evolve much beyond basic PDM processes when implementing PLM. 2) A solid PLM process definition is a critical success factor for a lasting PLM software implementation.

A question is whether conclusion 1 is a consequence of conclusion 2 and companies struggle to formalize and (re)define their processes. Apparently it is difficult for organizations which do not have in-house business process analysts to analyze their processes in a way that can be used effectively for a PLM software implementation.

## 2.2 Product development

The main idea of this paper is to apply product development principles to the implementation of PLM. Therefore, literature has been reviewed about the following product development methods.

**QFD.** Quality Function Deployment [7] is a methodology in which different aspects of product development are related to each other. It offers a number of matrix diagrams in which product characteristics, customer requirements, process characteristics, competition characteristics, etc. can be checked for correlation.

**SBCE.** Set based concurrent engineering [8-11] is a modular design methodology in which different areas of a complex product can be developed in parallel. For each subsystem, a broad set of feasible solutions is defined based on proven technology.

### 2.3 Product configuration and modularization

Another area of product specification that has been reviewed is product configuration. Many companies are investigating product configuration as a mean to lower development costs but increase the flexibility to customize products for their customers. The most common way to do this is with a modular approach. [12]. Modules are developed in variants. The final product is a combination of modules and parts based on customer requirements.

## 3 Proposed implementation approach

The main focus of this research is an implementation approach for holistic PLM with higher involvement of all stakeholders and lower investment needs for full business process analysis. The approach is structured in five major phases as pictured in figure 1. After defining the scope and process state, non-functional requirements are defined parallel to a three level process map. The operations in the lowest detail level of the process map serve as functional requirements. For each operation a set of possible PLM implementation alternatives is proposed. The alternative options are scored against the non-functional requirements and consolidated into a functional and technical specification of the entire PLM implementation. Solution sets are managed as knowledge in a knowledge base for later reuse.

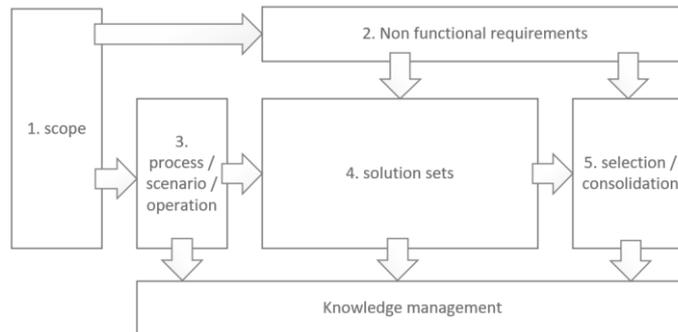


Figure 1. Schematic overview of steps in proposed set based PLM implementation method

In the following paragraphs, each phase is explained in more details.

### 3.1 Scope, State and maturity level

At the start of the method scope, state and maturity level must be made explicit for all stakeholders in the implementation process.

**Scope.** The stakeholder must define which part of the organization or processes will be affected and involved in the PLM implementation. Interactions between this part and other parts of the organization need to be identified and described in terms of input and output of materials and information.

**State.** The stakeholders must be able to answer whether the implementation of PLM affects the current state of the organization, future state or is a mean to transform from current to future state. This is important for process modeling in the third phase where process models need to be descriptive (describe what people are doing) or prescriptive (define what people should do) [13]. Bokinge and Malmqvist [6] found that the two main strategies for PLM implementation are: adopt the commercial software or adopt the business processes. This confirms the need to have a common awareness of state.

**Maturity level.** Maturity level assessment is a way to determine if an organization is ready for a specific level of PLM implementation. To do this, the frameworks of Batenburg et al. [1] is selected to use for work units and processes.

**VSM.** A suitable method to assess scope and state may be value stream mapping, where stakeholders are offered a method to reach a collective understanding of current and potential future state as well as the scope boundaries. A value stream can be used to define a scope definition for a specific product or product family [14].

### 3.2 Requirements analysis

Non-functional requirements are closely related to the business case of the implementation, since they deal with business aspects like legal constraints, economic constraints, safety, efficiency, capacity and performance.

Requirements analysis and requirements engineering (RE) are used in software development. RE distinguishes between functional and non-functional requirements, where non-functional requirements are requirements that cannot be defined as functional, data or process [15].

A framework of common requirements for companies in an industry segment needs to be defined for this paper's research. When starting RE with this framework, stakeholders can focus on the validation and negotiation phase and have better documented requirements sets at the end.

### 3.3 Process modeling

The implementation methodology in this research aims to offer a framework of process modeling conventions that describes the tasks that the organization has to perform in a way that is usable for a PLM system design, can be understood by all stakeholders and can be reused in a modular way for future implementations in the same industry segment.

**Three detail levels.** Main concept of this process modeling framework is the definition for three detail levels:

- Process level. This is used to identify the main activity in the high level process chain of the organization.
- Scenario level. These are the user stories or what people have to do in their work.
- Operation level. These are specific tasks that a person or machine performs to do the work.

Processes and operations are sequential in most cases. Scenarios do not necessarily have to be sequential. This is one of the key elements in this approach. A process can be seen as a black box with input and output. Inside the black box certain “things” have to be done, which are represented by scenarios. Scenarios are composed of operations. In figure 2 an example is given to explain the principle.

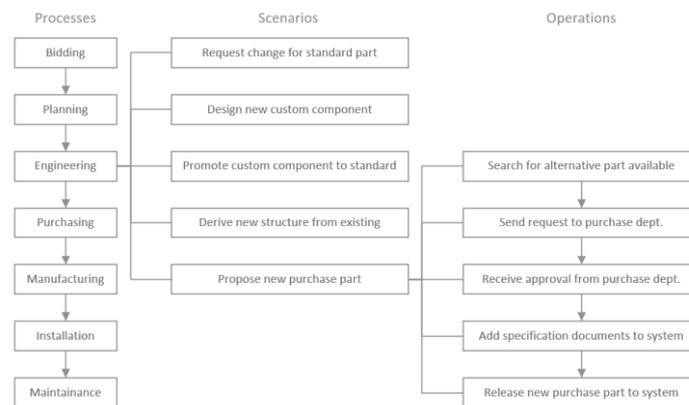


Figure 2. Example of 3-tier process mapping for engineer-to-order company.

**Reuse and duplicity.** Operations can appear in multiple scenarios and scenarios can be used in multiple processes. Also many operations are common in general or common for a specific industry segment.

**Modeling language.** For this paper three methods have been reviewed to capture and model processes in the three different levels.

**UML 2.0.** Unified Modeling Language [16,17] is a widely accepted framework of notations, used to develop software. It is very suitable for object oriented analysis and design and therefore can be used to model the interaction with data-objects. Since a PLM software implementation often uses already developed software, only a few diagram types are regularly used: mainly use-case diagrams, activity diagrams, use-case narrative and class diagrams.

**BPMN 2.0.** Business Process Model and Notation [18,19] has many similarities to the UML Activity Diagram (AD). BPMN has more meaningful symbols for business processes and is automatically connected to Business Process Execution Language. Research has been done on difference in readability between UML AD and BPMN, but this was not significant [20,21].

**VSM.** Value Stream Mapping [14] is a technique that originates in lean manufacturing. For this paper, VSM is useful because it focuses on the operation instead of the roles. It does not answer the question what a person does, but what is needed to be done in order to get a result. This leaves more room for alternative options for a given requirement.

After a review of different process modeling languages with a target group of PLM consultants, BPMN 2.0 has been named more suitable to model scenarios and operations. In figure 3, the previous example scenario is modeled in this language.

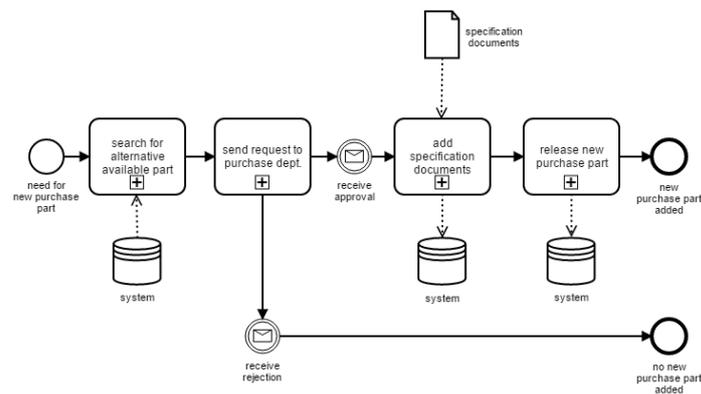


Figure 3. A scenario modeled in BPMN 2.0.

To standardize the method of process modeling, the way BPMN is used needs to be narrowed down in purpose specific conventions for this implementation strategy. The detail levels must be right, not too detailed and not too course. Following conventions are proposed in this context:

- Scenario in the organization = process in BPMN
- Operation within a scenario = sub-process BPMN
- Definition when to use triggers, conditions, forks and states
- Start and end conditions must be compatible with connected processes

### 3.4 Solution sets

The method to relate a finite number of possible solutions to an operation is inspired by SBCE [8]. The core principle is that more ways can be identified to perform an operation in a PLM environment. In the previous example, the scenario

“propose new purchase part” includes an operation “send request to purchase department”. A finite list of possible solutions could look like this:

1. Fill in a paper form and deliver it at purchase dept.
2. Write an email and send it to purchase dept.
3. Fill in a standardized digital form in a word processor and store it in a predefined location on the network.
4. Fill in a form in an ERP system
5. Create a new item in a PLM software and start a “propose” workflow for approval.

The set of a single operation (functional requirement) and the list of possible solutions (function fulfillment options) is a module in the implementation strategy.

### **3.5 Rationale based solution selection and consolidation**

When the solution sets are defined, different solutions can be related to the non-functional requirements. This can be done in a scoring matrix. Not each NFR requires a similar decision. Some NFRs are hard requirements (yes/no) and others are performance NFR that will result in a scalar value. Raudberget [22,23] has developed a number of these selection matrices for product development purpose that can be tailored towards the needs for PLM implementation.

After determining which solutions are preferred or candidate for the PLM implementation, the process of consolidation of the feasible solutions takes place.

### **3.6 Knowledge management**

In line with the principles of SBCE (proven technology) and product configuration (no development in the specification process), the solution sets should be re-used as much as possible. Therefore, a structure of knowledge management is needed.

Kennedy et al. [10] describe knowledge elicitation as a second value stream intersecting the primary value stream of the solution sets. Check sheets are used to capture design knowledge as the process is executed. With this method, capturing knowledge can be as accessible as possible for those involved in the implementation process.

## **4 Conclusion and future research**

The literature review, performed for this research, did not reveal any similar approach to PLM implementation as proposed in this paper. So far, it can only be suspected that this method has a higher stakeholder involvement with less implementation time compared to other methods, since it has not been put to the test. Initial feedback from domain experts who have experience in PLM implementation has been positive.

Until now, individual elements of the proposed framework have been tested ad-hoc in running projects. The process modeling phase has been tried most thoroughly

in a larger project as a second attempt after a “traditional” technical specification. The difference in involvement by the main stakeholder was significant and it laid a fundament under a successful implementation of software afterwards.

The next step to verify the proposed methodology is to perform case studies. A first case study is in progress at a Dutch SME company that develops waste recycling equipment. Phase 1 (scope) of the proposed method has been tested completely and confirms the benefits as expected. Stakeholders are more involved and there is a better common understanding of the project. Phase 2 and 3 are progressing in parallel, but it is too early for conclusions.

In parallel, a large number of finished PLM implementations are going to be analyzed in the three layered structure of processes, scenarios and operations. This will result in a better understanding of the commonality of scenarios and operations in a specific industry. The result will be used as input for a catalog of reusable solution sets.

To quantify the potential benefits of this proposed implementation methodology, an extensive survey is planned among PLM customers to unveil their initial motivations and expectation before they implemented the systems and what was realized at the end. Also, NFRs will be retrieved from this survey to build the framework of common requirements.

## References

1. Batenburg R, Helms R, Versendaal J (2006) *PLM roadmap: stepwise PLM implementation based on the concepts of maturity and alignment*. International Journal of Product Lifecycle Management 1(4):333–351
2. Silventoinen A, Pels H, Kärkkäinen, Lapela H. (2011) *Towards future PLM maturity assessment dimensions*. Paper presented at PLM11 – 8<sup>th</sup> international conference on Product Lifecycle Management
3. Silventoinen A, Papinniemi J, & Lampela H (2009) *A roadmap for product lifecycle management implementation in SMEs*. Paper presented at the The XX ISPIM Conference
4. Schuh G, Rozenfeld H, Assmus D, Zancul E (2008) *Process oriented framework to support PLM implementation*. Computers in Industry 59 (2008) 210–218
5. Navarro R, Cloonan J, Buboio R, Tiwari A. (2011) *Improving efficiency in Product Lifecycle Management implementation projects by applying lean principles*. Paper presented at PLM11 – 8<sup>th</sup> international conference on Product Lifecycle Management.
6. Bokinge M, Malmqvist J. (2012) *PLM implementation guidelines - relevance and application in practice: a discussion of findings from a retrospective case study*. International journal Product Lifecycle Management, Vol 6. No. 1
7. Eureka W, Ryan N. (1988) *The customer driven company, managerial perspectives on QFD* ASI Press, Dearborn, Michigan.
8. Ward A, Sobek II, D. (2014) *Lean Product and Process Development, Second Edition*. Lean Enterprise Institute, Cambridge, MA, USA
9. Sobek D, Ward A. Liker J (1999) *Toyota's Principles of Set-Based Concurrent Engineering*. Sloan Management Review, Winter 1999, 40(2), pp67-83.

10. Kennedy M, Harmon K, Minnock E. (2008) *Ready, Set, Dominate: Implement Toyota's set-based learning for developing products and nobody can catch you*. The Oaklea Press, Richmond, Virginia
11. Khan M, Al-Ahaab A, Doultsinou A, Shehab E, Ewers P, Sulowski R. (2011) *Set-based concurrent engineering process within the leanPPD environment*. Paper presented at the 18th ISPE international conference on concurrent engineering, Massachusetts, 2011.
12. Hvam L, Mortensen N, Riis J. (2008) *Product configuration*. Springer-Verlag, Berlin.
13. Münch J, Armbrust O, Kowalczyk M, Soto M. (2012) *Software process definition and management*. Springer-Verlag, Berlin.
14. Martin K, Osterling M (2014) *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill, New York.
15. Mahalakshmi K, Prabhakar R (2013) *Performance Evaluation of Non Functional Requirements*. Journal of Computer Science and Technology.
16. OMG. (2015) *Unified Modeling Language Specification, version 2.5*. Object Management Group (OMG).
17. Fowler M, Scott K (2000) *UML Distilled, second edition, a brief guide to the standard object modeling language*. Addison Wesley Longman, Reading, Massachusetts.
18. OMG. (2011) *Business Process Model and Notation (BPMN) Version 2.0*. Object Management Group (OMG)
19. Freund J, Rücker B (2014) *Real-Life BPMN: Using BPMN 2.0 to Analyze, Improve, and Automate Processes in Your Company*. Camunda, Berlin.
20. Geambasu C. (2012) *BPMN vs. UML Activity Diagram for business process modeling*. Accounting and Management Information Systems Vol. 11, No. 4, pp. 637–651.
21. Peixoto D, Batista V, Atayde A, Borges E, Resende R Pádua C. (2008) *A comparison of BPMN and UML 2.0 Activity Diagrams*. Proceedings of VII Simpósio Brasileiro de Qualidade de Software
22. Raudberget D. (2010) *Practical applications of set-based concurrent engineering in industry*. Journal of Mechanical Engineering 56(2010)11, 685-695
23. Raudberget D. (2011) *Enabling set-based concurrent engineering in traditional product development*. Paper presented at the International Conference on Engineering Design, 2011, Copenhagen.