

# Automatic Configuration of Modularized Products

Joel Sauza-Bedolla<sup>1</sup>, Stefano Amato<sup>2</sup>, Alfredo Fantetti<sup>2</sup>, Andrea Radaelli<sup>2</sup>, Alex Saja<sup>2</sup>,  
Gianluca D'Antonio<sup>1</sup>✉, and Paolo Chiabert<sup>1</sup>

<sup>1</sup> Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Turin, Italy  
{joel.sauza,gianluca.dantonio,paolo.chiabert}@polito.it

<sup>2</sup> Alta Scuola Politecnica, corso Duca degli Abruzzi 24, 10129 Turin, Italy  
{stefano.amato,alfredo.fantetti,andrea.radaelli,  
alex.saja}@asp-poli.it

**Abstract.** In business to business manufacturing, a major competitive advantage comes from the personalization of the product for the customer. In order to customize a product, companies go through a long process of customer interviews and specialized product development processes: this results in a time-consuming design phase and in a highly variable production process. In this paper, a method to improve the efficiency of product development and manufacturing, keeping a high degree of customization, is presented. A standardization effort is performed to identify a set of interchangeable components and to define a set of functional constraints. The consequences of such standardization are a dramatic reduction of the time expected to design and produce an item, as well as in lowered degree of variability of both the manufacturing process and the warehouses content. The presented methodology has been applied to a manufacturer of ink dispensing systems.

**Keywords:** Mass customization · Modularization · Product configurator  
Standardization

## 1 Introduction

Manufacturing companies are facing the well-known antithesis between high product variety and fast delivery time. Highly-customized products must be designed and produced into an increasingly competitive environment, and must satisfy the multifaceted needs of their customers; this leads to an intense effort for continuous and fast re-designs [1]. In the last years, different authors [2, 3] pointed out that a product configurator is an effective tool to support the response to this contrast. Customer requirements are quickly individuated, while his choices are guided through an automatic process that ends with a finite number of standardized products. Hence, the re-design effort for the company is severely reduced, whereas, from the customer perspective, service perception and satisfaction are improved [4]. An automatic product configuration also supports some central phases of Product Life cycle Management (PLM), in terms of possibility to automatically generate Bill of Materials (BOM), and to integrate internal functions of the company [5]. However, although these tools can be extremely effective on both

the internal and external performances, their implementation usually needs non-trivial efforts and money investment, and thus become prohibitive for small-sized companies. The present paper aims to develop a scalable methodology for the implementation of a product configurator, mainly devoted to Small-Medium Enterprises (SMEs) designing industrial machinery and scalable products whose structure can be decomposed in parametrical modules. The methodology is validated through a case-study: a product configurator has been implemented into a small company that assembles machines for mixing inks.

The rest of the paper is organized as follows: the state of the art analysis is discussed in Sect. 2. The methodology for the implementation of a product configurator developed in this work is presented in Sect. 3. The case-study and the validation of the model are presented in Sect. 4. Conclusive remarks and hints for future developments are presented in Sect. 5.

## 2 State of the Art

Product configuration has been an area of active research in the last years. Several ways to implement a product configurator have been proposed; depending on the chosen approach, different cost, development time and effort, configurator quality can be achieved. Felferning et al. [6] showed a method based on modeling the product using the Unified Modeling Language (UML) that can then be interpreted automatically by a configuration engine. Haugh et al. [7] compared seven different strategies to develop product configurators, each with its advantages and drawbacks for handling projects according to complexity, duration, and risks. Yang et al. [8] presented an approach for encoding configuration models into the Dynamic Constraint Satisfaction Problems (DCSP). Gembariski and Lachmayer [9] introduced a process model for defining multi-variant products. Wang et al. [10] described a method for modularizing existing products improving design efficiency. Although these works deploy different approaches, the following basic steps can be identified:

- Step 1. *Preliminary analysis*. This step consists in interviewing product experts and consulting company documentation to retrieve information about the knowledge and reasoning process underlying product development as well as on the projects formerly dealt.
- Step 2. *Knowledge representation*. This step consists in structuring the acquired information in a form that a computer system can utilize to solve a task. According to the chosen representation the literature classifies product configurators in the following categories [11, 12]:
  - Rule-based: product knowledge is expressed as a set of rules or implications. The system can draw conclusions using the logical process of deduction.
  - Model-based: the product is represented through decomposable entities and interactions between their elements.
  - Case-based: the knowledge necessary for reasoning is a set of records of configurations sold to former customers. The system attempts to solve the

current configuration problem by finding a similar, previously solved problem and adapting it to the new requirements.

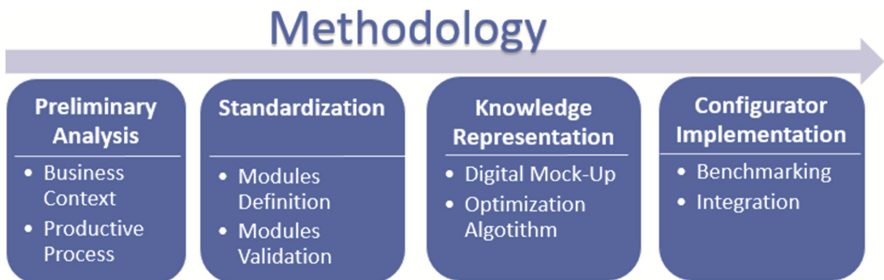
Step 3. *Configurator implementation*. The last step consists in implementing a software able to take in input the customer requirements, analyze them, and provide all the product information and specification necessary to validate the design and start the manufacturing phase.

Nevertheless, many investigations about the implementation of product configurators were focused on specific case-studies, and lacked generality. In particular, such researches mainly focused on the development of company-customized product configurators [13], sometimes with obsolete techniques [14]. Custom-built software can offer a direct and more effective improvement of firms' performance, but this implementation technique certainly requires an expensive Information Technology (IT) consultancy support [2]. This investment often discourages SMEs that aim to implement a product configurator.

Thus, an analysis on the support that new technologies and tools can provide in developing novel, cheaper solutions can be valuable. In particular, the present work aims to extend the state of the art by presenting a standard methodology applicable, even through low cost tools, to SMEs that aim to implement a product configurator to improve their performances by preserving a high product variety, and ensuring compliance with delivery time schedules.

### 3 Methodology

The three steps summarized in the previous section play a crucial role in integrating a product configurator within a company. However, the methodologies proposed in literature do not take into account the standardization tasks: this step plays a key role, especially in SMEs offering products with high customization or flexibility levels. Therefore, a four-steps methodology (shown in Fig. 1) is proposed here: the standardization tasks are placed between the Preliminary analysis and the Knowledge representation. The description of each step and the corresponding sub-steps is provided in the following.



**Fig. 1.** Schematic of the methodology to achieve a product configurator presented in this work.

### 3.1 Preliminary Analysis

The analysis of the company's requirements for the configurator is performed with a two-sided approach: an analysis of the external factors (business analysis) and internal aspects (productive processes).

**Business Context Analysis.** The preliminary analysis begins with a deep analysis the core business of the company. A significant timespan must be identified to study company processes: internal tasks must be decomposed, in order to understand the reasons of possible issues, time wastes, and delivery delay. Further, the communication channels between the company and its customers are studied: an overview of customers' requirements and constraints provides a basic idea about the general architecture of the configurator and its interface with the users.

**Productive Process Analysis.** A deep analysis of the products delivered in the observation timespan is necessary to define the features most frequently requested by the customers and the solutions most frequently provided by the company. This statistical analysis provides valuable information and suggestions for the subsequent standardization step. Moreover, this approach allows to directly evaluate the design efforts performed in the development of each product variant, outlining the amount of resources that could be saved by implementing a standard product configurator.

### 3.2 Standardization

The statistical analysis previously performed provides data useful to define a number of frequently adopted elements. In fact, one of the main goals of the product configurator is to reduce product variety to a limited set of variants. Therefore, the recurring elements must be standardized in order to avoid the proliferation of such variants: they are decomposed into elementary functional blocks and sets of standard modules are defined. Two phases are necessary:

**Modules definition.** The results of the preliminary analysis highlight the most frequently used components and the impact that each variant to the standard product has in terms of: (i) additional design efforts required; (ii) change in the production processes; (iii) change in the number and type of components; (iv) additional costs related to the variant. Each functional group with a significant recurring rate should be defined as a standard module.

**Modules validation.** To validate the set of standard modules, the projects performed in the observation timespan (or even in a longer time interval) must be checked again: the requirements of the customers have to be reconsidered, to check whether the standard modules enable to satisfy such necessities. In case some requests cannot be properly solved, the impact of such issue has to be evaluated and, if necessary, the set of standard modules must be enlarged.

### 3.3 Knowledge Representation

The objective of this stage is to create a library of reusable parts to virtually represent the product and to display a mockup of the product to the client. Moreover, in order to allow the easy evaluation of product variants, an optimization algorithm must be employed.

**Digital Mock-Up.** The standard modules have to be implemented into a library of parts and components designed with a modeling software, possibly in 3D. Nowadays, many CAD software allow connecting detailed parametric technical drawings of each part into a variable assembly. These platforms can store all the components, modules and assemblies, permitting to modify with basic instructions their main geometrical and appearance features. This possibility can be used to apply optional changes to a standard product, to obtain fast 3D product representations.

**Optimization algorithm.** One of the main issues in the final product definition is to find the best combination of modules according to the defined requirements. Hence, the definition of an optimization algorithm is necessary. A constrained optimization problem must be solved, where the modules are decision variables and the user's requirements represent the problem constraints. A quantity  $f$  to be optimized must be chosen. Possible choices for the objective quantity may include space occupation or economical costs. Therefore, for a given set of input requirements, the algorithm should find the best combination of modules which optimizes the objective quantity.

### 3.4 Configurator Implementation

The last step of the methodology consists of: the definition of the requirements of the system, the choice of the configurator software, and the implementation of the formalized knowledge. Selecting a configurator software already available on the market allows to exploit the advantages of software reuse [15]: lower production and maintenance costs, shorter implementation time, and increased software quality. Instead, creating a completely new configurator software would require an unaffordable financial effort for small and medium enterprises.

**Benchmarking.** In order to choose the software that best fits the needs of the company, all the existing configurator platforms must be considered. A feature matrix to compare the available alternatives must be deployed: each row is one of the requirements provided by the Preliminary analysis, which can be weighted according to a priority scale. Then the alternative which satisfies the most "Must have" requirements is selected, if the price is considered acceptable by the company; in case of ties, the alternative having the most "Nice to have" requirements is chosen. An example of such feature matrix is presented in the case study discussion.

**Integration.** In order to provide effective results, the configuration system must be well integrated into the company business processes as well as with the other IT systems deployed. First, the configurator must support the automated generation of sales quotes

and other specification documents, such as blueprints, BOM, and detailed product specifications. Further, the configurator can be required to communicate with other information systems, such as: the ERP, for order fulfillment, or PDM systems, for archiving custom product variants.

## 4 Case Study

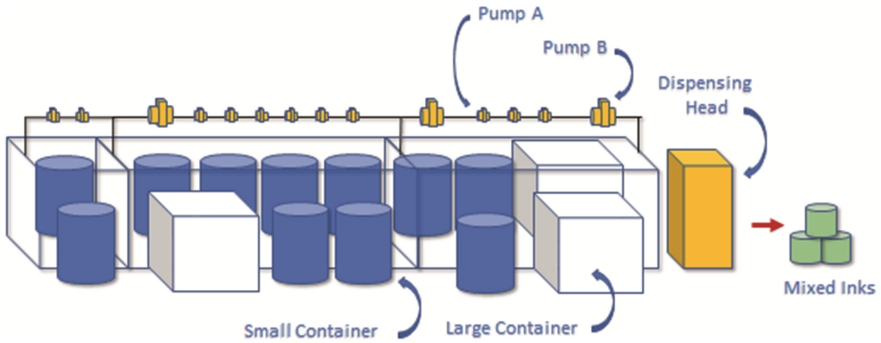
The methodology presented in Sect. 3 has been applied to a small manufacturing company in the area of Turin (Italy) that produces integrated dispensing systems for ink, paint and chemical dispensing and mixing. Due to an increasing product demand, the company decided to improve the design and production efficiency for a family of products. Thus, the implementation of a product configurator has been considered as a main objective by the company board and the presented methodology devoted to SMEs has been applied. In the following, data concerning machines specification will be anonymized to preserve industrial secret.

### 4.1 Preliminary Analysis

**Business Context Analysis.** The complete business analysis of the company is described in [16]. In the following, key aspects are presented. The company customers are both large and small businesses, which require automated dispensing systems for a huge array of applications. In particular, a subset of the company products has been considered, with prices ranging between 30-45 k€ and need long times for delivery (approximately 10 weeks), since the majority of them are actually tailor made in an engineering-to-order (ETO) approach.

**Productive Process Analysis.** When a customer requests a quotation, a company engineer is chosen as project chief and is in charge of designing a machine fulfilling the requested requirements. Since the company employs several engineers with different expertise, the lack of a standard design methodology leads to a huge product variability: the same set of requirements provided to different engineers can result in final products with different configurations. This approach leads to design and manufacturing inefficiencies. The years 2014-15 have been selected as observation timespan: in this period, 18 machines of the selected family have been designed and produced.

The requirements for each order and the solutions provided by the company have been carefully analyzed. An example of typical structure for a machine is shown in Fig. 2. Product analysis led to the following results. First, the dispensing head – which releases the final ink mixture into a small bucket – was found to be usually placed at one extremity of the machine. Second, two kinds of raw material containers, with different volume, are mostly used: they will be labeled Large Container (LC) and Small Container (SC). Third, one pump per each container is placed to carry the raw material towards the dispensing head; two types of pumps (A and B) are generally used for the two types of containers respectively.



**Fig. 2.** Representation of a machine based on the non-modular design approach.

The main design constraint was the maximum encumbrance of the machine imposed by the customer. Each designer dealt with this issue by proposing steel structures of different sizes. Furthermore, project chiefs stated that three kinds of requirements need particular attention: (i) Topology: space occupation and containers accessibility; (ii) Layout: position occupied by the different containers (LC should be close to the dispensing head) and pumps capacity; (iii) Maximum allowable structure length: profile section resistance, maximum loads.

## 4.2 Standardization

**Modules definition.** 15 modules were defined. Among the different constraints, topology was the most limiting in the design process, as the machine size is often in contrast with the space management of the facilities where it is to be installed. Each module consists in a structural part, a set of pumps, hydraulic and electrical connections and room for the containers. The modules differ for:

- type of containers: the modules can host (i) only LC; (ii) only SC; (iii) both the two types of containers;
- number of containers: three different standard lengths, based on the size of the steel profiles have been used (labeled L1, L2, L3);
- accessibility: modules can host containers (i) on both the sides or (ii) on a single side, for example to support installations close to wall.

**Modules validation.** The 18 analyzed projects have been redesigned through the set of standard modules, with the following results:

- 7 projects were totally accomplished, with a space occupation close to the original project, with a max difference of 1% (~ 10 cm).
- 7 projects were accomplished with an overlength smaller than 10%; the maximum surplus was equal to 70 cm on a 8.2 m machine;
- 1 project was accomplished with overlength greater than 10%;

- 3 projects were considered to be not solvable with standard modularization, because of the particular conformation of the available space, such as too small rooms, which did not comply with modules size.

Therefore, 15 projects out of 18 could have been created using the standard modules, leading to a dramatic simplification in the technical office job, the inventory organization and saving a lot of time to be spent in more challenging designs. The representation of a standardized machine is shown in Fig. 3; it can be compared to the non-standardized design shown in Fig. 2.

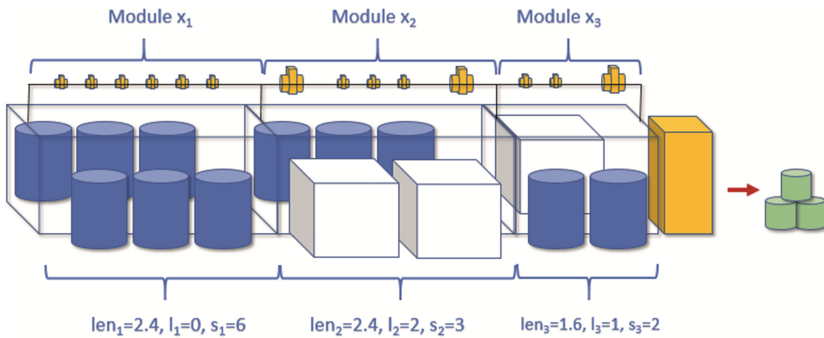


Fig. 3. Representation of a machine based on the modular design approach.

### 4.3 Knowledge Representation

**Digital Mock-Up.** The standard modules defined in the previous step have been modelled with a 3D CAD software already available in the company; a parametric design approach was adopted. The parametric models have been used, at a higher level, to propagate data between different layers of the assembly (interpart links) and to create associative copies of geometry between parts (constraints links).

**Optimization algorithm.** The Preliminary analysis showed that the variable to be optimized was the surface occupied by the machine. Hence, the chosen objective function  $f$  to be minimized has been the total length of the configuration. The following attributes were defined for each module: (i) length, denoted by  $len_i$ ; (ii) width, denoted by  $wid_i$ ; (iii) number of large containers, denoted by  $l_i$ ; (iv) number of small containers, denoted by  $s_i$ . The subscript  $i = 1, \dots, n$  denotes the identification for each module; in the present case study,  $n = 15$ .

Variables for the customer requirements were also defined:  $LCC$  and  $SCC$  denote, respectively, the number of the requested large and small containers. The size of the room – supposed to be rectangular – is stored in the variable  $R = (R_1, R_2)$ . The variable  $c$  is used to model the length of the dispensing head;  $\delta$  is used to determine the orientation of the machine in the space. The following optimization problem has been obtained:



$$\min \sum_{i=1}^n len_i x_i \quad \text{subject to} \left\{ \begin{array}{l} \sum_{i=1}^n l_i x_i \geq LCC \\ \sum_{i=1}^n s_i x_i \geq SCC \\ WID = \max_{j.s.t. x_j > 0} wid_j \\ len_i x_i + c \leq (1 - \delta)R_1 + \delta R_2 \\ WID \leq \delta R_1 + (1 - \delta)R_2 \\ x_i \in N \quad \forall i \in \{1, \dots, n\} \\ \delta \in \{0, 1\}, \quad WID \geq 0 \end{array} \right.$$

As this represents an LP problem, exact solution methods can be used, such as the Branch & Bound. The optimizer has been developed using the C# programming language to accommodate the needs of the company; to handle the LP problem, the open source library COIN-OR was used. Finally, an executable program, automatically run by the configurator, reads the inputs from the graphical user interface, loads the specific attributes of the modules, solves the optimization problem and yields as output a list of modules (optimal configuration) back to the configurator.

#### 4.4 Benchmarking and Development

The research of existing software led to an initial set of approximately 20 configurator systems. A first analysis enabled to reduce this selection to four alternatives: A = Auto-desk Configurator 360, B = Tacton CPQ, C = KBMax, D = MyCustomizer.

**Table 1.** Requirements matrix of the developed case-study.

		Alternatives			
Requirements	Priority	A	B	C	D
Bill of material generation	Must have	✓	✓	✓	✓
Web access	Must have	✓	✓	✗	✓
Multilingual support	Nice to have	✓	✓	✓	✓
3D visualizations	Nice to have	✓	✓	✓	✗
Engineering drawings generation	Must have	✓	✓	✓	✗
Integration with the company CAD	Must have	✓	✓	✓	✗
Extensible trough API	Must have	✓	✓	✓	✓
Mobile platforms support	Nice to have	✓	✓	✗	✓
Free trial	Nice to have	✓	✗	✗	✗

The full features matrix is shown in Table 1. The Alternative A best fits with such requirements. Data collected from the user through the web application are sent to the optimizer, which computes the best machine configuration. In turn, the optimizer provides the Alternative A with the machine configuration to generate the 3D visualizations, the specifications documents and the blueprints, which are embedded and shown to the user. Furthermore, the cost of the software licenses amounts to about 4500€ per year and they are considered acceptable by the company board. At the moment of writing

this paper, the integration between the configurator and the company ERP system was not yet developed.

## 5 Conclusions

Product configurators are an effective tool to balance the needs of product customization and manufacturing process standardization. However, a high effort is often needed to implement a configurator within a company, resulting in a low spread of such tool in SMEs. In this paper, a methodology to effectively realize, through low cost means, such tool is presented and validated through a case-study.

However, beside the mere implementation issues, further aspects must be considered. For example, internal issues may arise: employees could perceive this tool as a competitor in the workforce, a serious threat to their job. To tackle such issues, a multi-faceted approach is necessary: the management must point out that the configurator does not represent a substitute of human workforce, but represents a support to deal with repetitive tasks.

Furthermore, the definition and modelling of the standard modules is a time consuming activity which has to be considered by the company board. Resources also need to be allocated for the creation of an efficient optimization algorithm, either by hiring an external consultant or creating the algorithm with the company internal resources.

The presented approach results particularly effective with Engineering-To-Order (ETO) companies, whose core business consists in modular products, or whenever a parametrical modular decomposition is effectively possible. In fact, the results of this method are strictly bounded to the simplicity of the product, as an excessive product variety could introduce considerable difficulties in the implementation of an automatic configurator.

Research in this field could drive to the creation of an effective, affordable product configurator that generates these benefits for a larger number of companies, and could be a further step towards the popular concept of the Industry 4.0, pursuing the objective of a completely automated interaction between customer requirements and manufacturing sector.

**Acknowledgments.** This work has been supported by Alta Scuola Politecnica (ASP), an educational programme funded by Politecnico di Torino and Politecnico di Milano (Italy) devoted to promising students in Engineering and Architecture, based on ad-hoc courses and the development of multidisciplinary projects. <http://www.asp-poli.it>. The authors also thank Inkmaker srl, the industrial partner that supported this project.

## References

1. ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schonsleben, P., Tseng, M., Bernard, A.: Product variety management. *CIRP Ann. Manuf. Technol.* **62**(2), 629–652 (2012)
2. Zhang, L.L., Helo, P.T., Kumar, A., You, X.: Implications of product configurator applications: an empirical study. In: *International Conference on Industrial Engineering and Engineering Management* (2015)
3. Salvador, F., Chandrasekaran, A., Sohail, T.: Product configuration, ambidexterity and firm performance in the context of industrial equipment manufacturing. *J. Oper. Manag.* **32**, 138–153 (2014)
4. Trentin, A., Perin, E., Forza, C.: Increasing the consumer-perceived benefits of a mass-customization experience through sales-configurator capabilities. *Comput. Ind.* **65**, 693–705 (2014)
5. Terzi, S., Bouras, A., Dutta, D., Garetti, M., Kiritsis, D.: Product lifecycle management - from its history to its new role. *Int. J. Prod. Lifecycle Manag.* **4**(4), 360–389 (2010)
6. Felfernig, A., Friedrich, G., Jannach, D.: Conceptual modeling for configuration of mass-customizable products. *Artif. Intell. Eng.* **15**(2), 165–176 (2001)
7. Haug, A., Hvam, L., Mortensen, N.H.: Definition and evaluation of product configurator development strategies. *Comput. Ind.* **63**(5), 471–481 (2012)
8. Yang, D., Ming, D., Chang, X.-K.: A dynamic constraint satisfaction approach for configuring structural products under mass customization. *Eng. Appl. Artif. Intell.* **25**(8), 1723–1737 (2012)
9. Gembariski, P.C., Lachmayer, R.: Forward variance planning and modeling of multi-variant products. *Procedia CIRP* **21**, 81–86 (2014)
10. Wang, P., Liu, Y., Nee, A.: Modular design of machine tools to facilitate design for disassembly and remanufacturing. *Procedia CIRP* **15**, 443–448 (2014)
11. Blecker, T., Abdelkafi, N., Kreutler, G., Friedrich, G.: Product configuration systems: state of the art, conceptualization and extensions. In: *Eight Maghreb Conference on Software Engineering and Artificial Intelligence*, Sousse, Tunisia (2004)
12. Sabin, D., Weigel, R.: Product configuration frameworks - a survey. *IEEE Intell. Syst. Appl.* **13**(4), 42–49 (1998)
13. Kristanto, Y., Lelo, P., Jiao, R.J.: A system level product configurator for engineer-to-order supply chains. *Comput. Ind.* **72**, 82–91 (2015)
14. Forza, C.: Product configuration and inter-firm coordination: an innovative solution from a small manufacturing enterprise. *Comput. Ind.* **49**(1), 37–46 (2002)
15. Sommerville, I.: *Software Engineering*, 9th edn. Pearson, London (2011)
16. D'Antonio, G., Mottola, S., Prencipe, G., Rosa-Brussin, A., Sauza-Bedolla, J., Chiabert, P.: Deployment of product configurators: analysis of impacts within and outside the user company. In: *IFIP 14th International Conference on Product Lifecycle Management*, Seville (2017)