

Calculation of Complexity Costs – An Approach for Rationalizing a Product Program

| | | |
|---|---|---|
| Christian Lindschou Hansen | Niels Henrik Mortensen | Lars Hvam |
| PhD stud., Engineering Design and Product Development DTU Mechanical Engineering | Prof., Engineering Design and Product Development DTU Mechanical Engineering | Prof. Quantitative Management DTU Management Engineering lhv@man.dtu.dk |
| | nhmo@mek.dtu.dk | |

Abstract

This paper proposes an operational method for rationalizing a product program based on the calculation of complexity costs. The method takes its starting point in the calculation of complexity costs on a product program level. This is done throughout the value chain ranging from component inventories at the factory sites, all the way to the distribution of finished goods from distribution centers to the customers. The method proposes a step-wise approach including the analysis, quantification and allocation of product program complexity costs by the means of identifying of a number of suggested Life Cycle Complexity Factors (LCCFs). The suggested method has been tested in an action based research study with promising results. The case study shows how the allocation of complexity costs on individual product variants provides previously unknown insights into the true cost structure of a product program. These findings represent an improved decision basis for the planning of reactive and proactive initiatives of rationalizing a product program.

Keywords: Complexity costs; product program; rationalization; product architecture

Introduction

Challenges

Most industrial companies offering a multitude of product variants to the market have accepted that this situation comes with a price – it is not free to handle many product variants. In fact, it has been shown in numerous examples that all stages in the life cycle are affected by the variance in a product program. To diminish the negative effect of this “necessary evil”, many efforts have been made especially during the last 15-20 years to improve the marketing, design, production and management of product programs. One unavoidable means within this area is the application of architecture-based development of product platforms. Significant contributions are found in this research field, but very little research has been centered on the actual quantification of the benefits to be achieved from the architecture-based approach. There are many reasons for this. One is that established accounting systems (e.g. in ERP-systems) focus on the direct product costs alone (e.g. standard unit cost). This is done with only sparsely including indirect costs (often equally distributed on all variants) and also without focusing on the performance of the processes delivering the product program. Therefore, the experience of the authors is that a number of promising product architecture concepts are never implemented, due to the lack of quantification in order to justify the positive effects (or diminishing of negative effects) associated with the product architecture concept throughout the product life cycle.

The method

This paper proposes an operational approach for calculating complexity costs in order to rationalize a product program. The complexity costs are calculated throughout the product life cycle and are allocated to individual product variants. This is done in order to understand the negative effects of the product variants within a product program and to obtain a better measure of the profitability of individual product variants. The approach has two important objectives:

Reactive

- Reach an understanding of the product program complexity costs
- Rationalize the product program

Proactive

- Scoping of product architecture initiatives
- Improve program (or portfolio) management

The approach has been tested in an action based research study providing not only a basis for achieving short term gains in terms of rationalizing the product program (increased EBIT margin), but also a basis for reconfiguring the supply chain to achieve a number of beneficial effects.

The paper will continue by outlining the motivation for calculating complexity costs and put the topic into an industry perspective. Subsequently, the research perspective is elaborated by compactly treating the current state-of-the-art within this composed area. Finally the 5-step approach is presented, reflected and concluded upon.

Why calculate the complexity costs of a product program?

Motivation

As mentioned in the previous section, there are two main reasons to calculate the complexity costs of a product program:

Reactive use

In order to make room for new product introductions, it is often required to rationalize the product program on a frequent basis by eliminating and/or substituting product variants. Sometimes there is a 1:1 substitution of old product families with new product families, but often the situation is much more distorted than that, making it difficult to choose which variants to phase out. And since these discontinuation activities often rely on data readily available, the product variants to eliminate are often based on revenue or unit sales alone. This is without considering the total profitability of individual product variants, and without an overview of the complexity costs associated with the product variants in the product program. In other words, there is often a lack of cost transparency across the product program resulting in portfolio decision-making made without knowledge about the profitability of single product variants, thus also without knowledge about the burden with which individual product variants impact the indirect costs.

Therefore, there is a need to understand the profitability of individual product variants and include the calculation of complexity costs on a product program level, in order to:

- Obtain a more “true” product variant performance measurement
- Obtain an overview of the product life cycle complexity costs

This knowledge can enable the rationalization of a product program, by:

- Substituting unprofitable product variants with profitable ones where possible
- Eliminating remaining unprofitable product variants

As most companies have introduced stage-gate models or equivalent to control the introduction of new products, not much attention has been paid to professionalize the discontinuation task in an equal manner. Reactive use of this approach can contribute to professionalize this task.

Proactive use

Besides being a valuable input for reactive use, the knowledge about profitability of product variants and their associated complexity costs should be used proactively to avoid the increase of complexity costs to maintain a continuous increase in profitability.

As mentioned in the introduction, architecture-based development of product families are centered on providing the right variance to the market place while at the same time diminishing the negative effects experienced internally in the company's operations. As such initiatives cannot improve all processes at the same time; there is a strong need of scoping such architecture initiatives in order to reach the desired effects. As these effects are often found in a decrease of indirect costs, there is a need of identifying, allocating and analyzing these costs to surpass the paradigm of one-sided focus on the direct variable costs and step away from even distribution of indirect costs.

Therefore, in terms of scoping architecture initiatives, there is a need for utilizing the knowledge of product variant profitability and the calculation of complexity costs of the product program, in order to:

- Identify the most profitable product variants and families (and learn from these)
- Identify the least profitable product variants and families (and *fix* these)
- Identify the largest complexity costs and their associated factors (in order to know which ones to address)

This can be exemplified by:

- Design-for-X focus: Which life phase or universal virtue has the largest potential for complexity cost savings?
- Order fulfillment strategy: Where to place the customer order decoupling point?
- Guide market pricing: Which are the optimal price and cost points across the product program?

Besides from improvement projects, the continuous tasks of program (or portfolio) management can be improved by:

- Performance measurement: Introduce product program complexity key performance indicators to take complexity costs into account (and formulate new minimum profit thresholds etc.)
- Product planning: Guide product launch and discontinuation strategies

State of the art

A literature study was undertaken to screen the research landscape for recent contributions within this field. The literature study was broadened to comprise recent studies from supply chain engineering and management, and the most relevant ones are commented here. The focus has been contributions utilizing calculation of complexity-related costs in order to rationalize product programs.

Adjacent fields of research

Activity-based costing

Cooper and Kaplan [1] among others suggested Activity-based costing as a new method to avoid the deficiencies of arbitrary allocation of overhead costs. Activity based costing allocates indirect costs first to the activities performed by shared company resources, and hereafter assigns these to individual orders, customer or even products. Thus, the method takes its starting point in the resources used, links these to activities, and *then* to cost objects.

Anderson and Kaplan [2] proposes a more accurate and efficient cost modeling principle called Time-Driven Activity Based Costing (TD-ABC) that assigns resources (e.g. all costs of a customer service department) directly to cost objects (e.g. order handling). This is done to achieve a simple cost rate measure based on time consumption.

Supply chain engineering and variant management

Lechner et al. [3] proposes the method Variety-driven Activity-based Costing (VD-ABC) to quantify the impact from adding or removing product variants in automotive logistics, based on the use of hypothetical zero-variant scenarios. This is an expansion of the TD-ABC framework allowing for the calculation of incremental complexity costs associated with variants in different logistical operations.

Mass customization

Zhang and Tseng [4] propose a modeling approach to analyze cost implication of product variety in mass customization by bridging product variety with process variety. This is done by identifying cost drivers within the product design, and the method is confined to include manufacturing costs.

Product and product program complexity dimensions and indicators

Orfi et. al [5] proposes a set of product complexity dimensions (variety, functionality index, structural index, design index, and production index), and along with these associated indicators considering the cost impact of the product complexity dimensions. Nielsen and Hvam [6] showed that product program complexity is not just harming order management, procurement and inventory costs but also has a negative effect on delivery performance and product quality.

Complexity costs definitions

It is out of scope of this short paper to review the different definitions of complexity costs.

Complexity Management

Sivadasan et al. [7] (among others) describe two types of complexity in the supply chain, *structural complexity* (increasing with the no. of elements) and *operational complexity* (increasing with uncertainty of information and element flows). Many methods within supply chain research have suggested methods to eliminate and control this complexity from a supply chain point of view. Wilson and Perumal [8] are among recent contributions offering several top-down approaches to attack interrelated product-process-organizational complexity from a

managerial view point by diving complexity costs into value adding (good complexity) and non-value adding (bad complexity).

Also, a number of product variant rationalization approaches focus solely on different “tail-cutting” methods (often named SKU-rationalization), the most interesting ones showing that there is no relation between the number of stock keeping units and market share [9].

Gap

Acknowledging the “complex” landscape of contributions within this field, one can mention that much research is centered on the definition and calculation of one might call “internal” product complexity (inside the product) and much research (especially from the supply chain area) are centered on the control of complexity from a process point of view.

Product complexity methods

The deficiency with these methods most often overlook the fact that complexity is a relative phenomenon arising between a product and a process, and thus cannot be assessed meaningfully with regards to the product and its properties itself.

Process complexity methods

The deficiency with these approached are their exclusion of the product domain in order to derive advanced numerical approaches to the calculation of process complexity costs. This has its relevance for detailed optimization tasks, but is very difficult to use as input for proactive avoidance of complexity costs.

Conclusion on gap

Considering complexity as a relational phenomenon between e.g. a product and a process (within any life cycle phase), it is in principle impossible to derive universal metrics of complexity *before* an actual investigation of the unique product/process setup had been carried out, in order to assess the actual realized costs of complexity associated with the setup. Not before understanding the *realized* costs of complexity, efficient means to reduce the future costs of complexity can be defined. For example, having 10.000 variants might not be a problem if software is configuring the variance.

It is the aim of this approach, to take a first step towards the bridging of the supply chain based attempts to quantify complexity from a process point of view, with the architecture-based approach to product program design and the elimination of negative effects associated with the handling of many product variants.

The method

Introduction

The method presented here takes its starting point in the calculation of complexity costs on a product program level entailing a focus on product variants as the complexity cost allocation objects.

Step1: Scoping of analysis

In order to determine the focus of the analysis, the scope within the product program must be decided upon. It is advisable to select a confined number of product families to include produced on a limited number of production sites and perhaps sold in a limited amount of regions worldwide. It is possible to include product families from both high-end and low-end market tiers, as long as there is adequate resemblance in the way the products are produced and handled internally in order to analyze them concurrently.

Considering the costs of complexity within the product program, it is recommendable to choose product families experiencing a gap between projected profitability and realized profitability (if known to the project group).

Step 2: ABC analysis of product profitability

The first evaluation of the gross profitability of the product variants are made by collecting the realized revenues of all variants from all sales companies with a given time period. The direct product costs are subtracted from this figure resulting in a measure of the gross contribution margin. This can be plotted for all variants on the vertical axis leaving the horizontal axis to display the net revenue per variant – as shown in Figure 1.

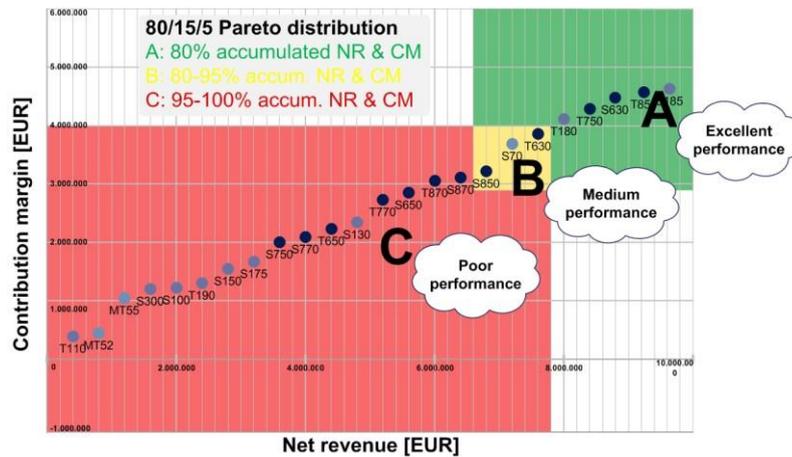


Figure 1 – ABC analysis of product profitability

A Pareto-distribution can be used to highlight the variants contributing only to the last 5% (C-variants) and 15% (B-variants) of the total contribution margin and net revenue.

Step 3: Life Cycle Complexity Factors (LCCFs)

Identification

With Step 2's early indication of the least contributing variants, the aim of Step 3 is to investigate which factors throughout the product life cycle that holds the largest complexity costs and find out whether these complexity costs distort the picture shown in Figure 1. A central aspect here is to look after LCCFs that could represent an *asymmetric cost distribution* across the product variants. In other words, look for pools of resources that are consumed differently among the product variants. An example of a LCCF includes cost of inventory (of materials, components, sub-assemblies and/or finished goods).

As LCCFs vary greatly between industries and company types, this paper will not go into details here (a future publication will expand the concept of Life Cycle Complexity Factors). However, their identification requires iterating between a top-down and a bottom-up approach:

- *Top down: Cost structure view*
Looking at the overall cost structure of the business area, where do we see the largest unallocated cost pools with a potential variant impact?
- *Bottom up: Hypotheses of cost asymmetry*
Based on the experiences of key resources, where is it likely that product variants contribute unevenly to the indirect costs?

Analysis, quantification and allocation

The basic idea here is analyze the LCCFs and find quantification objects that allows for approximations of the indirect costs in order to allocate them directly to product variants where applicable. By dividing all costs with the net revenue recorded on each variant – all costs are comparable as percentages. Often, it is necessary to settle with incomplete data extracts, and be creative in applying unconventional quantification objects to bring forward reliable approximations (inspiration can be found in [2] and [3]).

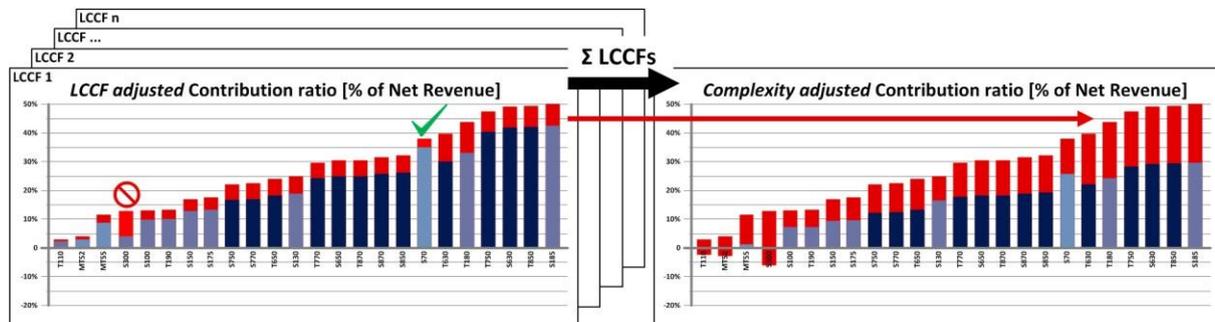


Figure 2 – Adjusting contribution ratios for LCCF induced complexity costs

As seen in Figure 2 the costs allocated from the analysis and quantification of LCCFs can be accumulated to give an overview of the complexity adjusted contribution ratios [%] (as well as the complexity adjusted contribution margins [EUR]). The right side of Figure 2 is the best possible estimation of the true profitability of the product variants.

Step 4: Short-term fixing

Based on the insights from step 2-3, it is now possible to calculate different scenarios of “fixing” the product program by the means of the reactive measures mentioned earlier. Most often, a thorough analysis reveals several low-hanging gains, and several percentage points of increased EBIT are usually the results of this. It is important not to assume 0% substitutability of discontinued variants (rare cases only). Assume a decent percentage in order to estimate the true incremental revenue loss, which is always a lot less than the actual revenue recorded on the variants.

Step 5: Complexity reduction program

As Step 4 is about cleaning up the product program reactively, Step 5 is about implementing the findings from Step 2-4 proactively in a complexity reduction program. A central aspect here is to identify the *drivers* causing the LCCFs to create complexity costs. We name these the LCCDs (Life Cycle Complexity Drivers) and the aim is to work with these drivers to decrease their negative impact on complexity. An example might be country specific customization of product variants that could be solved differently. The cost transparency achieved in Step 3 can serve to justify the cost of changing the country specific feature or solution.

The complexity reduction program contains the initiatives of diminishing the negative effects of the complexity cost factors (e.g. by postponing the customer order decoupling point) and initiatives of actively working with the complexity cost driver to eliminate them or delimit their negative effect on the factors.

Experiences from application

Case

The approach was applied in an action research-based study in globally leading manufacturer of mechanical consumer products. The approach was followed from Step 1 to Step 5. A number of factors supported the success of the approach leading to a large potential EBIT increase. Firstly, the company has a long history of mergers and acquisitions and is operating through a global supply chain and a global sales organization creating long “distances” and much room for cost distortion from revenue generation to cost allocation. Secondly, recent SKU-rationalization initiatives were solely based on revenue leaving a lot of improvement potential behind. Thirdly, many initiatives concerning the complexity drivers were put on hold due to the lack of quantification to support a business case narrowly focusing on standard unit cost. The application of the approach improved the cost transparency, refined the SKU-rationalization and provided a basis for continuing and scoping the initiatives of eliminating the negative effects of the complexity cost drivers.

Reflection and further work

The incorporation of hypothetical single-variant scenarios could be relevant in order to gain input for the estimation of “variant cost sensitivities”. The performance of all operations depend on the number of variants, and estimating this variant cost sensitivity across functions or major fixed cost pools can be valuable input. Also, further operationalization of the identification and allocation of complexity costs is relevant. Even though complexity costs do not derive from products individually, refined allocation methods for assigning these costs on component/product/product family level can help to identify the cost asymmetry of interest.

Conclusion

This paper proposes an operational method for rationalizing the product program based on the calculation of complexity costs. This is done by obtaining a “true” measure of the product cost through the identification of the largest and most asymmetric Life Cycle Complexity Factors (LCCFs) to provide a comprehensive input for quantitatively assessing the benefits of streamlining the product program. The nature and size of the LCCFs are a strong and non-negligible input for prioritizing initiatives of a complexity reduction program. The method was tested in an action-research based case study with promising results.

References

- [1] Cooper, R., Kaplan, R.S., 1988. Measure Costs Right - Make the Right Decisions. *Harvard Business Review*. 66, 96-103
- [2] Anderson, S.R., Kaplan, R.S., 2007. *Time-Driven Activity-Based Costing: A Simpler and More Powerful Path to Higher Profits*. Harvard Business School Press Books
- [3] Lechner, A., Klingebiel, K., Wagenitz, A., 2011. Evaluation of product variant-driven complexity costs and performance impacts in the automotive logistics with variety-driven activity-based costing. *IMECS - Int. MultiConference Eng. Comput. Scientists*. 2, 1088-1096
- [4] Zhang, M., Tseng, M.M., 2007. A Product and Process Modeling Based Approach to Study Cost Implications of Product Variety in Mass Customization. *IEEE Trans. Eng. Manage.* 54, 130-144
- [5] Orfi, N., Terpenney, J., Sahin-Sariisik, A., 2011. Harnessing Product Complexity: Step 1-Establishing Product Complexity Dimensions and Indicators. *Eng. Econ*. 56
- [6] Nielsen, J.B., Hvam, L., 2011. Product Complexity Impact on Quality and Delivery Performance. *World Conference on Mass Customization, Personalization, and Co-Creation*
- [7] Sivadasan, S., Efstathiou, J., Frizelle, G., Shirazi, R., Calinescu, A., 2002. An information-theoretic methodology for measuring the operational complexity of supplier-customer systems. *ijopm*. 22, 80-102
- [8] Wilson, S.A., Perumal, A., 2009. *Waging War on Complexity Costs*, 1st ed. McGraw-Hill, Europe
- [9] Mahler, D., Bahulkar, A., 2009. Smart complexity. *Strategy & Leadership*. 37, 5-11