The causes of product configuration project failure

Haug, Anders; Shafiee, Sara; Hvam, Lars

Published in:
Computers in Industry

Link to article, DOI:
10.1016/j.compind.2019.03.002

Publication date:
2019

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Haug, A., Shafiee, S., & Hvam, L. (2019). The causes of product configuration project failure. *Computers in Industry*, *108*, 121-131. https://doi.org/10.1016/j.compind.2019.03.002

Final draft post-refereeing. Published in Computers in Industry 108 (2019) 121–131.

The causes of product configuration project failure

Anders Haug[a], Sara Shafiee[b], Lars Hvam[c]

a Department of Entrepreneurship and Relationship Management, University of Southern Denmark, Universitetsparken 1, 6000 Kolding, Denmark
b Department of Mechanical Engineering, Technical University of Denmark, Denmark
c Department of Management Engineering, Technical University of Denmark, Denmark

Abstract

Several studies reporting benefits from the use of product configurators in engineering-oriented companies can be found in the literature. On the other hand, some literature also claims that companies abandon configurator projects in many cases. Such claims are, however, supported only by references to the experience of the authors and do not involve much information about the projects. This represents a significant gap in the literature, as insights from failed configurator projects could help prevent other companies from making similar mistakes and promote the development of more appropriate methods and tools. To address this gap in the literature, this paper outlines an overall framework for understanding failure in configuration projects. Using this framework, eight failed configurator projects are investigated. These cases demonstrate that poor decision-making in one phase can have escalating negative consequences in the subsequent phases until the configurator project eventually fails. Furthermore, based on the insights obtained from the eight projects studied, this paper defines a set of guidelines for avoiding failure in configurator projects.

Keywords: Product configuration; Product configurator; Mass customisation; Configuration project failure; Engineer to order (ETO)

1. Introduction

Product configurators are a subclass of expert systems, which represent one of the most successful applications of artificial intelligence principles in recent decades (Sabin &Weigel, 1998; Blecker et al., 2004; Belle at al., 2018). Product configurators include a knowledge base with information about product features, product structure, production processes, costs and prices (Forza and Salvador, 2007), allowing them to simulate work normally carried out by product experts, such as sales staff and engineers. Automating and mass-customising knowledge work through the use of configurators can achieve a number of benefits, including: reduced time for generating quotes, fewer errors in product specifications, less resource use for product specification, more exact quotes, higher similarity of sold products, better supplier communication and more (Ardissono et al., 2003; Forza and Salvador, 2002a, 2002b, 2007; Gronalt et al., 2007; Mortensen et al., 2010; Haug et al., 2011; Trentin et al., 2012; Haug, 2013; Hvam et al., 2013).

Although the literature includes several studies that show the possible benefits of configuration projects, the reality is a bit more nuanced. In fact, it has been claimed that many projects fail to achieve the planned benefits or are even abandoned before being implemented in the organisation (Møldrup and Møller, 2004; Ladeby and Oddsson, 2011; Haug et al., 2012). The problem in understanding the causes of such failures is that companies are often reluctant to talk about them, as opposed to their successes (Møldrup and Møller, 2004). This is also reflected in the product configuration literature, which does not include detailed studies of failed projects. However, studies of failed configurator projects could be highly valuable, as they could offer insights that may help other companies avoid making similar mistakes, and they could also provide future configuration research with a stronger basis for the development of methods and tools.

In the general IT literature, several authors have addressed the topic of project failure (Whitney and Daniels, 2013; Dwivedi et al., 2015). However, this literature is too general to understand the situation of configuration projects in detail. More specifically, although many of the general challenges of IT projects also apply to configurator projects, there are also certain challenges particular to product configurators. These challenges largely concern the complexity of such systems, which involve: a diverse set of process elements (e.g. machines, operations); a high variety of component parts and assemblies; and a large number of constraints and rules (Ardissono et al., 2003; Salvador and Forza, 2004; Zhang and Rodrigues, 2010). Thus, development of configurators concerns not only data and information, but also knowledge about how to specify products according to customer demands. Such knowledge (e.g. about what is a suitable product design in relation to a particular set of customer demands) is often rather subjective, which can make it difficult to elicit or get domain experts to agree on (Forza et al., 2006; Haug, 2012). To add to this complexity, configurator projects typically involve a range of different stakeholders with varying interests (Hvam et al., 2008; Haug, 2010), and their costs and benefits can be difficult to estimate before a project (Friedrich et al., 2014; Shafiee, 2107).

To address the gap in the literature on failures of product configuration projects, this paper posits the following question:

What are the reasons for failure of product configurator projects?

This paper focuses on product configurator projects involving automation of engineering work, i.e., product developments choices affecting product costs, producibility, quality, performance, etc. This typically concerns configurators applied by ETO (engineer-to-order) companies as part of the quotation process (Mortensen et al., 2010; Haug et al., 2011; Haug, 2013; Hvam et al., 2013). Here it should be noted that for ETO companies, product standardization typically only involves part of the product assortment, and often configurators are only used for preliminary design (as opposed to detailed design) in order to make quotations (Haug et al., 2009). Furthermore, it should be noted that although the paper focuses on ETO companies, its results may also be relevant in the context of make-to-order and assembly-to-order companies. This kind of investigation is, however, outside the scope of this paper.

The remainder of the paper is structured as follows. First, relevant literature is summarised. Based on this, the paper proposes a framework for organising failures of configuration projects. Then, the paper studies eight failed configurator projects and develops a set of guidelines for avoiding failure in configurator projects based on these.

2. Literature review

In line with the research question of this paper, the literature review focuses on identifying reasons for why configuration projects fail. To begin with, at a project management level, two archetypical types of projects may be considered when evaluating the causes of project failure (Meredith and Mantel, 2002), namely Type 1 and Type 2 projects. Type 1 projects are defined as being well-understood routine projects that have clearly defined scopes and involve relatively little uncertainty but may involve high technical complexity. Thus, failure in such projects would typically be a result of a lack of expertise in handling unexpected deviations from the plan. Type 2 projects are characterised by having many unknowns and an unclear scope. Therefore, besides complexity, a major cause of failure in such projects concerns planning-related problems because of an unclear scope (Meredith and Mantel, 2002). These two failure-promoting issues have been considered in the configuration literature, i.e. the handling of product complexity (e.g.

Sabin and Weigel, 1998; Haug and Hvam, 2007) and configurator project scoping (e.g. Hvam et al., 2008; Shafiee et al., 2014).

Configurator projects in ETO-contexts fall into the category of IT projects of high complexity. For complex IT projects, Murray (2000, p. 441) defined the following attributing factors of project failure: (1) unrealistic project scope with regard to resource availability and experience; (2) excessive use of new technology (for the company or industry) deemed critical for success; (3) business issues related to the project being new for the company or not well understood; (4) software vendors promoting application packages that exceed the companies' ability to manage effectively; and (5) the company's improper management of project scope by allowing continuous expansion.

Barki et al. (2001) proposed a similar classification on causes of failure in IT projects, which includes: newness of the technology, application size, application complexity, lack of experience and organisational environment. Based on such risks, Ewusi-Mensah (2003, p. 43) defined a set of factors in software project abandonment: (1) unrealistic project goals and objectives; (2) poor project team composition, project management and control problems; (3) inadequate technical expertise, problematic technology base/infrastructure; (4) lack of executive or support/commitment; (5) changing requirements; and (6) cost overruns and schedule delays.

The topic of project failure has also been addressed in the product configuration literature, although no detailed studies of failed projects are found in academic journals. In this context, a research project on product configuration in relation to economy, technology and organisation was carried out from 2003 to 2004 (Edwards et al., 2005). Besides successful cases, this research project also tried to include failed projects in order to get a wider scale of analysis. However, in spite of their "persistent efforts", companies refused or kept dragging out their acceptance, implying that they needed to exclude such cases (Møldrup and Møller, 2004).

As opposed to systematic studies of failed configurator projects, some authors have referred to their experience without providing much information about these projects. One example is Haug et al. (2012) who, based on the experience of the authors and without providing information about these cases, described a set of causes of configuration project failure, including that (1) the task of developing the configurator turns out to be much more difficult and time-consuming than anticipated, (2) projects become more costly than anticipated, (3) projects fail "to produce prototypes that indicate a probability of success", (4) configurators are not accepted by the organisation and (5) the persons who are going to use the configurator do not agree on the information that has been implemented. Another example is Ladeby and Oddsson (2011) who, based on their experience, claimed that configurator projects are "prone to delays or failures" and argued that configurator projects may fail in three types of developments tasks: (1) configurator development (i.e. software-related), (2) organisational development (i.e. change management) and (3) user development (i.e. training).

Although not addressing configurator project failure specifically, several studies have identified challenges of such projects. These include Barker and O'Conner (1989), who described the configurator project at Digital Equipment Corporation. In this study, Barker and O'Conner identified three dimensions in which configurator projects may face challenges: (1) strategic/business (connecting the software development with the business), (2) technical (software-related and development-related) and (3) human resource/organisational (role clarity, actor evolvement and task execution). Another example is Haag (1998), who gave a general outlook on the problems of adding a sales configurator to the SAP's R/3 business software suite. Based on SAP customer experiences, Haag mentioned three main shortcomings of the previous configurator modules, which were addressed by the third generation SAP configurator: (1) configurator functionality, (2) modelling ease, and (3) integration. Further issues mentioned include:

support for engineer-to-order scenarios; support for upgrades; integration of advanced features and business processes; and product data reuse.

Tiihonen et al. (1996) carried out a survey of 10 Finnish companies to study their product configuration problems. Here, they identified the following five “problems areas”, each divided into a set of factors: (1) economic importance of product configuration, (2) product configuration task, (3) product configuration process, (4) long-term management of product knowledge and configurations, and (5) interfaces to other systems and processes.

Ariano and Dagnino (1996) presented a case study of a furniture manufacturing company applying a product configurator for the creation of bills of materials. The case study identified five main challenges for implementation: (1) too few persons understanding the software developed, (2) lack of management support, (3) a need to change functions within the organisation, (4) a need for highly specialised knowledge to further develop the system and (5) difficulties in communicating the benefits produced by the system in order to ensure continuous support.

Tiihonen et al. (1998) further elaborated on the studies of 10 Finnish companies by Tiihonen et al. (1996) to define the following configurator project challenges: (1) understanding (often re-engineering) the sales-delivery process, (2) designing products for configurability, (3) long-term management of configuration models, (4) ensuring a match between company needs and configurator functionality, and (5) considering the suitability of processes for being supported by configurators.

Aldanondo et al. (2000) used their experience from 10 cases (which they did not describe) to conclude that it can be challenging to master the two kinds of expertise that are needed to develop a configurator, namely industrial expertise and configurator development expertise. The problem of not having employees who possess both types of expertise is that it produces a need for knowledge transfers between experts from different domains. According to the authors, it is typically too time-consuming to train people to become experts in both areas. Other issues mentioned by the authors include challenges of representing the structure of the configurator models and identifying the right questions to ask the customers during the configuration process.

Forza and Salvador (2002a) described the case of a small voltage transformer manufacturing company. In this case, many problems arose because of the difficulty in collecting, storing and retrieving precise and correct information on the characteristics of the many product variants within the configurator scope. These problems implied that the sales department frequently needed to consult the technical staff to supply customers with information about order feasibility, which was not always possible, implying delays and even errors. The need for the technical staff to be involved in sales support activities also absorbed many engineering hours for routine configuration activities, as opposed to product development tasks.

Forza and Salvador (2002b) described a case study of the implementation of a product configurator in a small manufacturing enterprise (mould bases for plastics moulding and punching bases for metal sheet punching). The project faced several major challenges, including that product modelling proved to be challenging because of high product variety. This implied that many resources were needed to create these product models and that these became highly complex, which made them difficult to implement into the configurator software. Furthermore, after launching the system, it was difficult to document the product model, which was needed to avoid maintenance becoming tied to the person who implemented the product knowledge in the configurator.

Forza et al. (2006) described the case of a company producing electric motors and alternators. In this study, they found that for product families of high complexity and variety, defining the solutions spaces would be extremely challenging and most likely not economically feasible. Thus, these product families were not implemented into configurators.

Hvam et al. (2006) described the case of a large manufacturer of cement plants. In this case, there were challenges in implementing the product configurator into the organisation and gaining acceptance from the users and domain experts. On this basis, they argued for the need to train and educate the users of the configurator, as well as to make a plan for how to introduce the system to the users.

Zhang and Helo (2016) conducted a survey of 61 companies to investigate the implications of product configurators. The challenge mentioned by most respondents was ensuring continuous product evolution, i.e. preventing the product solution space from becoming fixed because of challenges in developing the knowledge base of the configurator. Other frequently mentioned challenges included a lack of IT system designers and developers, unclear customer requirements and employee concerns about job loss.

Myrodia et al. (2017) described the case of a building elements manufacturer and installation service provider. In this case, three major challenges were identified: (1) adequate testing of the configurator before launch, (2) difficulties in supporting the entire product portfolio and (3) employee resistance to work routine changes.

Based on a literature review, Kristjansdottir et al. (2018) categorised the main challenges faced by manufacturing companies in relation to the use of product configurators: (1) IT-related, (2) product modelling, (3) organisational, (4) resource constraints, (5) product-related and (6) knowledge acquisition. The authors investigated the individual importance of these categories of challenges through a survey of 22 companies, showing organisational issues to be the most frequently encountered challenge (68% of the companies) and product-related issues to be the least frequently encountered challenge (23% of the companies).

Finally, it should be noted that, if the focus is moved away from the complex configurators in ETO companies and towards the technically much simpler online configurators to be operated by regular consumers, more studies have been carried out (see www.configurator-database.com). However, such configurator projects face much different challenges from the ones encountered in ETO companies (i.e. more marketing issues as opposed to technical complexity issues), which means their findings are not transferable to the topic of this paper.

3. A framework for understanding failure in configuration projects

To structure the causes of failures in product configurator projects, such projects may be understood as going through different phases. Based on descriptions of configuration projects in the literature (Forza and Salvador, 2007; Hvam et al., 2008; Haug et al., 2012; Hvam et al., 2013; Shafiee et al., 2018), a configuration project may be perceived as going through five distinct phases, divided by the actor types with main responsibility for these phases. This is illustrated in Figure 1, where it should be noted that the five phases or the five roles are not strictly divided, as phases may be carried out simultaneously or progress in an iterative manner. Furthermore, the persons involved in the projects may have multiple roles; for example, the ones acting as knowledge engineers may also be the ones developing the configurator (Haug et al., 2012; Shafiee et al., 2018). These five phases are subsequently further described.

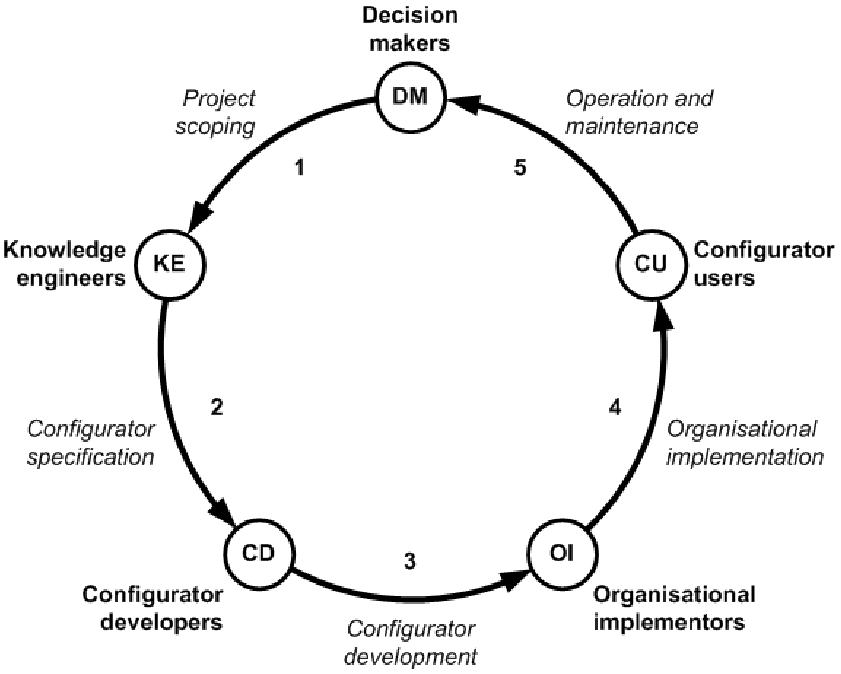

Fig. 1. Configurator project cycle

As illustrated in Figure 1, the first phase, project scoping, concerns relevant decision makers defining the scope of the project. The initial scope defined in this phase may later be altered during the course of the project as experience is gained (Hvam et al., 2008; Shafiee et al., 2014).

The second phase, configurator specification, concerns the acquisition of product-related knowledge and system requirements, as well as transforming these into a design of the future system. More specifically, the knowledge that will constitute the knowledge base of the configurator is acquired from relevant domain experts, and system requirements are retrieved from future users (Hvam et al., 2008; Haug et al., 2009; Shafiee et al., 2018; Friedrich et al., 2014). This information is sometimes formalised into a configuration design before being implemented into the configurator software (Hvam et al., 2008; Haug et al., 2009; Felfernig et al., 2014; Myllärniemi et al., 2014).

The second phase includes both knowledge acquisition and configurator design. In this context, it could be argued that these two tasks should be perceived as separate phases, given their different focuses. However, in practice the two phases can be extremely hard to separate, since configurator design decisions often occur while acquiring knowledge and requirements, and sometimes the descriptions made while acquiring knowledge are directly passed on to the configurator developers, for which reason such descriptions are both knowledge acquisition documentation and configurator design specifications.

In the third phase, configurator development, the configurator specification is converted into one or more configurators. Here, knowledge engineers may carry out the actual modelling in a configurator software shell or support more dedicated software experts in creating the configurator (Hvam et al., 2008; Haug et al., 2009).

The fourth phase, organisational implementation, concerns relevant parties being informed about the configurator, in part to ensure that the upcoming users are properly motivated to use it (Hvam et al., 2008). Obviously, this work may occur before completion of the previous phase but is initiated as soon as the decision about developing a configurator is made. The phase also includes future users being taught how to use the configurator (Hvam et al., 2008).

In the fifth phase, operation and maintenance, the configurator is in operation, which involves continuously updating information and further configurator development (Hvam et al., 2008; Haug, 2010;

Shafiee et al., 2017). Evaluations of the effect of the configurator may also be carried out. Such evaluations form the basis for further decisions, which may involve extending the configurator scope, developing additional integrations with other IT systems, creating additional configurators or terminating projects. As project evaluations during the operation and maintenance phase can give rise to re-scoping of existing configurators or the development of additional configurators, the entire process may be illustrated as a cycle.

Configurator projects may (at least in principle) be abandoned in each of the five phases described in Figure 1. Specifically, in each phase problems may be encountered that render the project unfeasible or unprofitable, and thus it is terminated. However, problems leading to project termination may have also been "inherited" from previous phases. In other words, poor decision-making in one phase may negatively affect subsequent phases. These hypothesised termination points and indirect contributors to project failure are explored in the empirical studies described below.

4. Research method

Based on the framework outlined by this paper, the empirical studies focused on understanding the reasons that configurator projects fail. Given that the literature did not provide much insight into this topic, an explorative approach was applied. Such an approach is generally considered appropriate when constructs for a phenomenon are yet to be clearly identified and delineated (Eisenhardt and Graebner, 2007; Yin, 2009). More specifically, case studies of eight failed configurator projects were carried out (Eisenhardt and Graebner, 2007; Yin, 2009). Hence, the unit of analysis is a product configurator project.

The first step of the research was to identify projects that had failed. This was done through discussions among the authors, who were familiar with many such projects through their network with consultancies, industrial companies, software vendors and other researchers. Here some delimitation was made to enable comparisons, namely that (1) the companies were ETO (engineer-to-order) companies, (2) the configurators were aimed at internal users and (3) the companies applied standard configurator software shells. The first two delimitations were applied to ensure that the configurator projects were of adequate complexity, as opposed to the comparatively simple online sales configurators. The third delimitation was applied because, according to studies in the literature (e.g. Forza and Salvador, 2002a, 2002b; Hvam et al., 2008, 2013; Haug et al., 2011, 2012; Shafiee, 2017), the use of standard software appears to be the most common approach in ETO companies, and the development of configurators while not using standard software could be expected to involve other types of challenges. Finally, the focus was limited to Danish companies because the authors' networks mainly included such.

The second step concerned getting access to information about failed projects. This task turned out to be challenging, as it was hard to make knowledge engineers talk about such failures. Thus, the final case selection was mainly driven by possibilities rather than opportunities, i.e. revelatory case sampling (Yin, 2009). However, the study aimed to include at least one project being terminated at each of the four last phases of the model in Figure 1. Eight projects ended up being included in the study.

In the third step, data from the selected projects was collected in the form of project materials and semi-structured interviews with knowledge engineers who had been involved in the failed product configuration projects. Retrieved documents included product models, process diagrams, time plans, project descriptions, screenshots and similar. The interviews were conducted based on an interviews guide, focusing on the clarifying the course of events in the projects, in particular, signs of problems and failures during the projects. The interviews were given on condition of anonymity.

In the final step, data from the selected projects were analysed. Documents were analysed through reviews and discussions in the researcher group. Interviews were analysed and compared to the document data through discussions of transcripts of interviews and notes taken during these. Here the focus was on identifying decisions and events during the course of a project that could have contributed to the later termination of the project. Such decisions and events were subsequently organised according to the framework described in Figure 1.

Information about the eight projects studied is shown in Table 1. Here it should be noted that projects 4 to 6 were conducted in the same company but with different project managers and users. As seen, the table includes information about if the configurators have a sales and/or engineering focus; where "sales focus" refers configurators only producing preliminary designs as a basis for calculating quotation prices, while "engineering focus" refers to configurators capable of producing detailed bills of materials. In all the cases, only part of the products sold were supported (or supposed to be supported) by configurators. Furthermore, in case 3 to 8 standard configurator shells were used, and in case 2 intended to be used. Finally, in all the cases, the configurators included, or were estimated to would have included, more than 1000 attributes and rules/constraints. Thus, all the projects involved relatively complex configurator knowledge bases.

Table 1. Overall project characteristics

Case	**Company industry**	**Company size ***	**Configurator focus ****	**Specification lead time *****	**Failure point**
1	Flow control equipment	8,000	S + E	5-20 days	1) Project scoping
2	Building contractor	15,000	E	10-30 days	2) Configurator specification
3	Building modules	100	S	5-20 days	3) Configurator development
4	Chemical process systems	3,000	S	20-90 days	3) Configurator development
5	Chemical process systems	3,000	E	5-30 days	4) Organisational implementation
6	Chemical process systems	3,000	S	5-30 days	4) Organisational implementation
7	Electronic products	100	S + E	3-5 days	5) Operation and maintenance
8	Electronic components	2,000	E	2-5 days	5) Operation and maintenance

*** Approximately**

**** Sales (S); Engineering (E)**

***** Lead time for sales and/or engineering specification processes before configurator use**

5. Results

This section briefly presents the findings of each project, then discusses them and converts them into a set of guidelines for configurator projects.

5.1 Case 1

The first case involves a flow-control equipment manufacturer. The company had heard about other companies applying product configurators with great success, which made them consider pursuing this path as well. Thus, the company contacted a consultancy bureau with expertise in the matter. However, after two meetings and a workshop arranged by the consultancy bureau, the company pulled out of the project. The reasons given were that the project would require too great an investment in relation to the uncertainties associated with the potential benefits. Thus, in relation to the model illustrated in Figure 1, the project was abandoned in the scoping phase.

5.2 Case 2

The second case involves a large building contractor. The configurator project focused on developing a configurator that could provide support for engineers by supporting engineering work in relation to a particular building element. This involved both standard and customisable components (e.g. dimensions). To collect knowledge from domain experts, a consultancy bureau was hired. As the company was not entirely sure about implementing extracted knowledge into a configurator, decisions related to software development were not made before initiating the knowledge acquisition phase. The configurator project was terminated in the configurator specification phase, as the product seemed poorly suited for configuration (too little product standardisation).

Issues related to each of the phases are described in Table 3.

Table 3. Case 2

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The relevance of and motivation for the use of configurators seem to have been overestimated • Lack of clarity about project scope entailed the creation of product models that were later discarded
Configurator specification	• Analyses of how a configurator could support the product showed that the product was not well suited for it, and thus the idea was abandoned.
--- Terminated ---	

Although the configurator project was abandoned, the project was (at least officially) not considered to have been a failure by the company. More specifically, the models of the building parts in focus proved to be highly useful as a basis for further product development in the direction of more standardised solutions. Additionally, as no investments were made in configurator software or development, the costs of efforts used on the project (meetings and presentations) were considered to be relatively insignificant.

5.3 Case 3

The third case involves a medium-sized company producing and mounting customised building parts, consisting of both standard and custom-made components. This configurator project focused on developing a configurator that could provide support for sales staff in sales processes by automating engineering work and calculating prices. Furthermore, the configurator should be able to generate a 3D model of the defined product based on customer demands.

A consultancy bureau was hired to collect knowledge from domain experts; as for software development, a software company was hired to develop the configurator using a proprietary software shell they had developed. The project was terminated in the configurator development phase after about three years of development. At this point in time, cost budgets had been raised several times without the software company being able to provide satisfactory prototypes.

Issues related to each of the phases are described in Table 4.

Table 4. Case 3

Contributing causes to project failure:	**Explanation**

(Initial) scoping phase	• The decision to include 3D models turned out to be overly ambitious • The choice of configurator software and company requirements turned out to be a poor fit
Configurator specification	• More details than initially acquired were needed by software developers • The efforts needed to create the configurator in the specific software were underestimated by the knowledge engineers • The scope was unrealistic and should have been changed in this phase
Configurator development	• The software developers kept exceeding repeatedly raised budgets until the project was terminated
--- Terminated ---	

5.4 Case 4

Case 4 involves a sales configurator in a large ETO company specialising in the production of chemical processing systems and plants. The project focused on the sales process for a particular type of catalyst. The goal of the project was for the configurator to produce sales prices and preliminary bills of materials for 70% of the sales requests.

Issues related to each of the phases are described in Table 5.

Table 5. Case 4

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The project plan did not consider how knowledge should be collected and documented • The project scope was not clearly defined and constantly changed during the project
Configurator specification	• The knowledge acquired was not documented in an adequately structured form • Some information needed could not be acquired because of a lack of both knowledge engineer and domain expert resources • There was some resistance from domain experts to share knowledge • The number of man-hours needed was underestimated • The user interface design was not discussed with relevant stakeholders
Configurator development	• Relevant stakeholders disagreed about what the project scope was, which made it hard to know what to implement • The results of the configurator tests did not live up to the decision-makers’ demands
--- Terminated ---	

In Case 4, the tests did not fail technically but in relation to the perception of the quality of the configurator by relevant business managers. However, three years after the termination of the project, a new configuration project for the product family in focus was initiated. This project faced some of the same problems, but eventually a configurator was put into operation in the company.

5.5 Case 5

Case 5 involves the development of a technical configurator at the same company as in Case 4. This project focused on creating a technical configurator that could generate detailed bills of materials, as well as other engineering specifications. The project was initiated because of the success of a sales configurator for the product in focus. However, as compared to the sales configurator, the complexity turned out to be significantly higher.

Issues related to each of the phases are described in Table 6.

Table 6. Case 5

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The expectations for the configurator may have been unrealistic, as the success of the sales configurator was not directly translatable to a technical configurator project
Configurator specification	• The knowledge to be acquired was rather complex, and the management of this process was somewhat unstructured • The scope was not clearly defined, making it unclear what to specify
Configurator development	• The main stakeholders of the project (e.g. the business project manager) changed during the course of the project, causing confusion among developers about direction • The project managers had a different vision for the project than the developers, and they failed to communicate this • Inadequate resources were assigned to testing the system
Organisational implementation	• The department meant to use the system did not accept the system and the project was terminated
--- Terminated ---	

One year after having abandoned the project, a project with a similar focus was launched. However, instead of building a separate technical configurator, it was decided to extend the sales configurator to include both. To ensure the success of this project, another project manager was assigned to the project, and the budgets were significantly raised. This project, however, led to a configurator being put into operation.

5.6 Case 6

Case 6 was a project in the same company as in cases 4 and 5. The project focused on supporting both sales and engineering processes for a particular type of chemical processing plant and involved configurator integration with the engineering simulation tool (simulating chemical processes for a given configuration of a product). The goal of the project was for the configurator to produce sales price and a preliminary bill of materials for 70% of the sales requests and with a level of detail covering 60% of the engineering work. The project was terminated after three years of development in the organisational implementation phase.

Issues related to each of the phases are described in Table 7.

Table 7. Case 6

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The scope of the project was too large • Too many stakeholders were involved • A proper architecture for integration with other systems was not defined • The scope involved knowledge of a complexity difficult for the knowledge engineers to understand • Knowledge access aspects were not adequately considered, implying that all relevant knowledge was not acquired
Configurator specification	• Procedures and policies to validate acquired knowledge were missing

	• The software development task was underestimated (overly ambitious design) • The project scope was changed several times in this phase
Configurator development	• A lack of access to relevant information in the company made certain requirements impossible to fulfil • The tests of the system required many experts in various company IT systems, several of whom were not available during the project • The scope was significantly reduced because of data accessibility and test issues
Organisational implementation	• After having tested the system, much resistance from users and department managers was encountered because of the system's complexity and limitations • Eventually, the department that the configurator was aimed at refused to use the system
--- Terminated ---	

The major lessons of this project were to ensure organisational support before launching a project and to beware of overly extensive projects with regard to the number of stakeholders. At the time of the study, the company had not given up the ambition to produce a configurator with a similar focus in the future but was aware of the need to handle such a project differently, most particularly to ensure stakeholder support before initiation.

5.7 Case 7

Case 7 involves a medium-sized company that produces a particular type of industrial electronic products. The configurator project aimed to build a configurator that could produce a quote for business customers and a preliminary bill of materials for customised products fulfilling the most common customer demands. The configurator was to be used by both sales staff and business customers with a certain level of product expertise. The configurator took about one year to develop. The configurator was taken out of operation about 4-5 months after having been launched.

Issues related to each of the phases are described in Table 8.

Table 8. Case 7

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The expectations for the level of customer use were unrealistic • Sales staff (future users) were not significantly involved in defining the scope to ensure their support • The potential benefits of the projects were not clearly defined
Configurator specification	• Parts of the configurator specifications were too detailed for them to be profitable to develop • The sales staff and customers (users) were not significantly involved in the design of the configurator
Configurator development	• User tests were almost not conducted
Organisational implementation	• Few efforts were made in "selling" the idea to the upcoming users • Only very limited training was given to sales staff
Operation and maintenance	• The sales staff did not believe that the configurator made their job easier and some had fears of being replaced by the configurator; thus, use was limited • Customers only used the web-based configurator to a very limited extent and preferred to communicate with the sales department just as before the configurator • Because of lack of use of the configurator, decision-makers decided to abandon the

	configurator
--- Terminated ---	

In Case 7, the interviewed knowledge engineer argued that the development team might have been a bit blinded by the potential of the technology, while failing to recognise that the future users had very different worldviews and interests.

5.8 Case 8

Case 8 involves a large company that produces a particular type of electronic component for different industrial uses. The configurator project was aimed at building a configurator that could produce quotes for the business customers and bills of materials for the production department. The configurator took about one year to develop. Only one person carried out the knowledge acquisition, as well as the configurator design and development. The configurator was dropped about two years after having been launched, shortly after the person in charge of the project left the company.

Issues related to each of the phases are described in Table 9.

Table 9. Case 8

Contributing causes to project failure:	**Explanation**
(Initial) scoping phase	• The scope had a rather short-term perspective, one-sidedly focusing on the development of the configurator, as opposed to operation and maintenance aspects
Configurator specification	• The knowledge engineer did not create documentation in a form understandable to others • The configurator design did not adequately consider further development and maintenance issues
Configurator development	• The models in the configurator did not have a clear structure, making them difficult to develop further • The configurator knowledge base was not documented in any form, making it hard to maintain
Organisational implementation	• While the configurator was well received by the users, the configurator was not integrated into the IT department, but remained a “one-man project”
Operation and maintenance	• After two years, the person in charge of the configurator development left the company, and attempts to update it by others failed. Being unable to update the configurator, it was taken out of operation shortly afterwards
--- Terminated ---	

The company later initiated a new and successful configurator project in which they carefully ensured that the configurators could be maintained, even in cases of employees leaving. This included documentation of implemented knowledge, as well as multiple developers working on the configurator.

6. Discussion of findings

As the eight cases showed, configurator projects may be abandoned for a number of reasons, often as a result of decisions made in previous phases. This is illustrated in Figure 2, where T1 to T5 (“T” for "termination") illustrate the five possible points of project termination and the 10 dotted arrow lines illustrate indirect contributors to project failure.

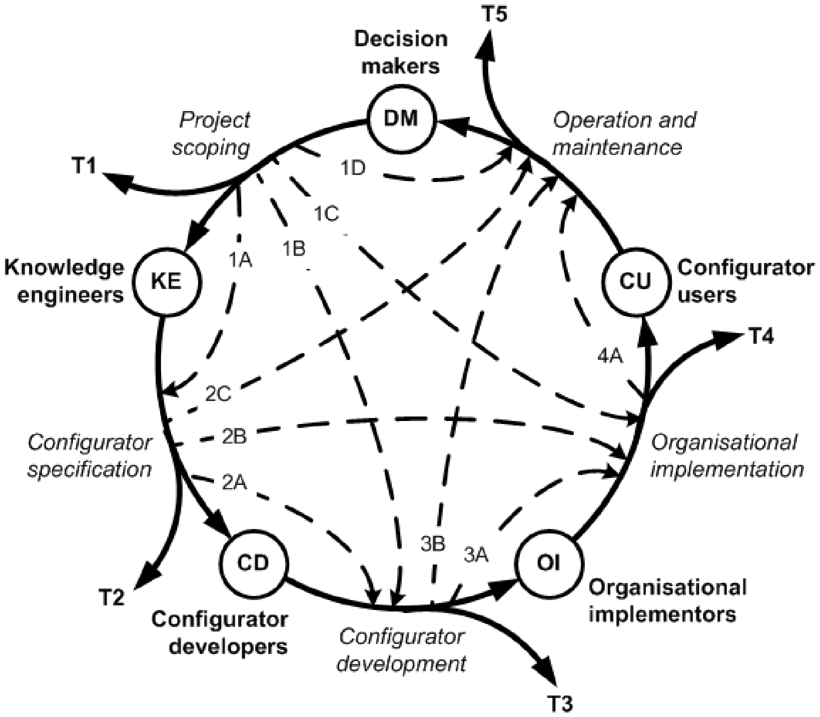

Fig. 2. Project termination points and their causes

To demonstrate the empirical basis for the five termination points and 10 types of indirect causes of configurator project failure, select examples from the cases studied are shown in Table 10 and 11, respectively.

Table 10. Causes of project termination

Relation	Causes identified by the case studies
T1	The possible project scopes do not seem adequately profitable (Case 1)
T2	The products are poorly suited for configuration (Case 2)
T3	The configurator development costs surpass budgets (Case 3) The configurator development does not live up to decision maker demands (Case 4)
T4	The configurator does not live up to stakeholder demands (Case 5 and 6)
T5	The configurator is not adequately used because of a lack of user acceptance (Case 7) The configurator cannot be maintained and further developed due to lack of structure/documentation (Case 8)

Table 11. Indirect causes of project failure

Relation	Example from case studies
1A	The project scope involves inaccessible product knowledge (Case 4)
1B	The project scope involves an unsuitable software shell (Case 3)
1C	The project scope is not in line with organisational wants and needs (Case 6)
1D	The project scope lacks adequate maintenance procedures (Case 8)
2A	The configurator design does not match the configurator software (Case 3)
2B	The configurator design does not match organisational wants and needs (Case 7)
2C	The configurator design makes maintenance difficult (Case 8)

3A	The configurator ends up not matching organisational wants and needs (Case 6)
3B	The configurator ends up being hard to maintain (Case 8)
4A	Adequate maintenance procedures are not defined (Case 8)

As seen in Table 11, all the 10 indirect causes of configurator project failure could be identified in the case studies. Based on the development in the cases, it appears that poor decision-making in one phase can have escalating negative consequences in the subsequent phases, until the configurator project eventually fails. Thus, it may be argued that there is a need for a holistic, rather than procedural, approach to configurator projects — i.e. that each decision made is considered in relation to all subsequent phases, rather than merely solving a present problem. One possible way to address this could be to anchor projects with a few key persons throughout the project life cycle.

The 10 indirect causes of project failure can be understood through the Type 1 and Type 2 project typology by Meredith and Mantel (2002). With regard to the first four (1A-D), these concern problems related to project scope and are, therefore, Type 2 problems (i.e., many unknowns and an unclear scope). The last four indirect causes (2C, 3A, 3B and 4A) relate to Type 1 problems (i.e., a lack of expertise in handling unexpected deviations from the plan); where cause 2C, 3B and 4A concern an inability to carry out configurator changes and cause 3A concerns an inability to fulfil configurator specifications. The last two causes (2A and 2B) may relate to both Type 1 and Type 2 problems. More specifically, a configurator design that does not match the configurator software or organisational requirements could both be explained by unclear scopes and a lack of expertise. Based on these observations, it seems that problems emerging early in configuration projects are more related to Type 2 problems, whereas problems emerging later in a project are more related to Type 1 problems.

7. Guidelines

The causes of configurator project failure identified in the eight cases may be converted into a set of guidelines for avoiding failure in configurator projects. These guidelines are shown in Table 12.

Table 12. Guidelines for avoiding failure in configurator projects

Phase	**Lessons learned from the case studies**
1) Project scoping	1) Carefully investigate if the products and processes are suitable for configuration (Case 2)
	2) Be careful about buying or making firm decisions about configurator software at this stage, since it later may turn out to be unsuitable (Case 3)
	3) Ensure that relevant product knowledge is assessable (Case 4 and 6)
	4) Ensure that relevant stakeholders agree on the project scope (Case 4)
	5) Be careful about assuming that earlier configurator successes can be translated into successes with new projects (Case 5)
	6) Ensure that the scope is realistic (Case 6)
	7) Be careful about involving too many stakeholders (Case 6)
	8) Ensure that the system can be integrated with other IT systems (Case 5)
	9) Ensure the support of relevant users (Case 6, 7)
	10) Involve relevant users in defining the project focus and scope (Case 6, 7)
	11) Be clear about the focus and benefits of the project (Case 7)
	12) Make a plan that goes beyond development but also involves operation and maintenance (Case 8)
2) Configurator specification	13) Ensure agreement on the scope before initiating knowledge acquisition (Case 6)
	14) Investigate whether products need to be redesigned to become more configurable (Case 2)

	15) Ensure that enough detail is extracted to build the configurator (Case 3)
	16) Apply a structured plan to acquire, communicate and maintain the knowledge (Case 4, 5, 6)
	17) Ensure that acquired knowledge is well documented (Case 4, 8)
	18) Ensure that knowledge engineers have or have access to the needed product expertise (Case 6)
	19) Define procedures for validating acquired product knowledge (Case 6)
	20) Ensure that domain experts are motivated for sharing knowledge (Case 7)
	21) Ensure agreement on the scope before initiating configurator design (Case 5, 6)
	22) Involve future users in configurator design (Case 4, 7)
	23) Wait until configurator requirements are fully known before entering into contracts with software suppliers (Case 3)
	24) Consider if the design can be realised within budget constraints (Case 6)
	25) Consider the necessary interfaces to other systems (Case 6)
3) Configurator development	26) Ensure agreement on the scope before initiating configurator development (Case 4)
	27) Do not initiate configurator development phase before ensuring the development plan is realistic with regard to time and resources (Case 3, 4, 5, 6)
	28) Do not initiate configurator development phase before ensuring the configurator specification is realistic (Case 5, 6)
	29) Do not raise budgets before carefully examining the realism of promised deliveries (Case 2)
	30) Make agreements with stakeholders about testing and product knowledge validation procedures (Case 5, 6)
	31) Ensure that adequate resources are assigned and used for technical tests (Case 5)
	32) Ensure that adequate resources are assigned and used for user tests (Case 7)
4) Organisational implementation	33) Ensure adequate resources for training users (Case 7)
	34) Ensure adequate efforts to get users to support the configurator project (Case 5, 6, 7)
	35) Ensure that managers will support the process in the future (Case 5, 7)
	36) Be aware of the organisational changes, especially for long-lasting projects (Case 5)
	37) Ensure that configurator maintenance and further development can be carried out by multiple employees (Case 7)
5) Operation and maintenance	38) Ensure that changes made are properly documented in the knowledge base, as it may otherwise be extremely difficult to update (Case 8)
	39) When necessary, continue efforts in making users support the project (Case 7)
	40) Ensure that user change demands (changes and additions) can be fulfilled (Case 8)

As seen in Table 12, some of the guidelines are relatively general for IT projects, while other are more specific for product configurator projects. In this context, it should be noted that none of the guidelines alone are absolutely configurator project specific, as there exists a plethora of IT systems, of which some would fit at least one. On the other hand, in combination the guidelines are configurator project specific. In this context, a main aspect that sets product configurator projects (as well as many other knowledge-based systems) apart from general IT projects is the acquisition, implementation and maintenance of complex domain expert knowledge. What is special for product configurators, as compared to other knowledge-based systems, is that their knowledge base concerns engineered products. Thus, the guidelines that in particular are related to product configurators are items 1-2, 14-20, 28, 30, and 37-38.

Finally, as stated in Section 3, the different phases of configurator projects may be carried out in an iterative manner, which is an aspect of configurator projects that may be promoted by the application of the guidelines. More specifically, iterations may be a consequence of consulting the guidelines to detect if issues related to previous phases have emerged, and in such cases, to go back to the previous phases and address these issues.

8. Conclusions

This paper addressed the gap in the literature on why product configurator projects fail. More specifically, the paper developed a five-phase framework for organising configurator project failures, on which basis eight failed configurator projects were studied. From these studies, the identified direct and contributing causes to configurator project failure were transformed into a set of guidelines for avoiding failures in configurator projects. These guidelines may be used in future projects to critically examine the choices made in the project.

The projects studied demonstrated that poor decision-making in one phase may produce escalating negative consequences in the subsequent phases, until the configurator project eventually fails. The implication of this is that there is a need to employ a more holistic, rather than procedural, approach to configurator projects. More specifically, each decision can have consequences in all subsequent phases; therefore, a decision should not be viewed in relation to the current or next phase, but in relation to the project as a whole. This may be challenging, as different actors are typically involved throughout the project. However, anchoring the project with a few key persons throughout its life cycle would obviously be one way to address this.

The studies demonstrated that the proposed framework was a useful means for organising configurator project failures, in the sense that decisions and events could be placed under the five phases and their effects on subsequent phases could be clearly defined. In other words, configurator project failure can be understood in terms of the phase in which the project is terminated, and the decisions made in previous phases contributing to this.

Given that formal studies of failed configurator projects could not be found in the literature, the contributions of this paper address a significant gap in the configurator literature. As compared to existing claims about reasons for configurator project failure, this paper provides more extensive and detailed explanations, supported by systematic empirical studies (as opposed to authors' experiences). It should, however, be noted that since only eight cases were studied, more causes could exist. To identify such possible causes, future research should engage in more in-depth case studies, based on the overview provided by this paper. Also, future research should engage in investigations of the relative importance of such failure causes. For practitioners, the case descriptions and guidelines may help steer them away from potential failures. For future research, the paper provides insights into a type of configurator project that had yet to be explored — insights that may be useful, for example, in relation to further development of methods and techniques to support configurator projects.

References

Aldanondo, M., Rougé, S., and Véron, M. (2000). Expert configurator for concurrent engineering: Caméléon software and model. *Journal of Intelligent Manufacturing*, 11(2), 127–134.

Ardissono, L., Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Petrone, G., Schäfer, R., and Zanker, M. (2003). A framework for the development of personalized, distributed web-based configuration systems. *AI Magazine*, 24(3), 93–108.

Ariano, M., and Dagnino, A. (1996). An intelligent order entry and dynamic bill of materials system for manufacturing customized furniture. *Computers & Electrical Engineering*, 22(1), 45–60.

Barker, V.E., O'Connor, D.E., Bachant, J., and Soloway, E. (1989). Expert systems for configuration at Digital: XCON and beyond. *Communications of the ACM*, 32(3), 298–318.

Barki, H., Rivard, S., and Talbot, J. (2001). An integrative contingency model of software project risk management. *Journal of Management Information Systems*, 17(4), 37–69.

Belle, A.B., Lethbridge, T.C., Garzón, M., and Adesina, O.O. (2018). Design and implementation of distributed expert systems: On a control strategy to manage the execution flow of rule activation. *Expert Systems with Applications*, 96(1), 129-148.

Blecker, T., Abdelkafi, N., Kreutler, G., and Friedrich, G. (2004). Product configuration systems: State of the art, conceptualization and extensions. In A.B. Hamadou, F. Gargouri, and M. Jmaiel (Eds.), *Proceedings of the eight Maghrebian conference on software engineering (MCSEAI 2004)*. Sousse, Tunis: Centre de publication Universitaire, 25–36.

Dwivedi, Y.K., Wastell, D., Laumer, S. et al. (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1), 143-157.

Edwards, K., Hvam, L., Pedersen, J.L., Møldrup, M. and Møller, N. (2005). *Udvikling og implementering af konfigureringssystemer: Økonomi, teknologi og organisation (Final report from PETO research project)*. Copenhagen, Denmark: Technical University of Denmark.

Eisenhardt, K.M., and Graebner, M.E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50(1), 25-32.

Ewusi-Mensah, K. (2003). *Software development failures anatomy of abandoned projects*. Cambridge, MA: MIT Press.

Felfernig, A., Hotz, L., Bagley, C., and Tiihonen, J. (2014). *Knowledge-based configuration: From research to business cases*. Burlington, MA: Morgan Kaufmann Publishers.

Forza, C., and Salvador, F. (2002a). Managing for variety in the order acquisition and fulfilment process: The contribution of product configuration systems. *International Journal of Production Economics*, 76(1), 87–98.

Forza, C., and Salvador, F. (2002b). Product configuration and inter-firm coordination: An innovative solution from a small manufacturing enterprise. *Computers in Industry*, 49(1), 37–46.

Forza, C., and Salvador, F. (2007). *Product information management for mass customization: Connecting customer, front-office and back-office for fast and efficient customization*. New York, NY: Palgrave Macmillan.

Forza, C., Trentin, A., and Salvador, F. (2006). Supporting product configuration and form postponement by grouping components into kits: The case of MarelliMotori. *International Journal of Mass Customisation*, 1(4), 427–444.

Friedrich, G., Jannach, D., Stumptner, M., and Zanker, M. (2014). Knowledge engineering for configuration systems. In A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen (Eds.), *Knowledge-based configuration from research to business case*s. Waltham, MA: Morgan Kaufman, 139–155.

Gronalt, M., Posset, M., and Benna, T. (2007). Standardized configuration in the domain of hinterland container terminals. In G. Friedrich, L. Hvam, and F. Salvador (Eds.), *Series on business informatics and application systems innovative processes and products for mass customization: Vol. 3*. Berlin: GITO-Verlag, 105–120.

Haag, A. (1998). Sales configuration in business process. *IEEE Intelligent Systems and Their Applications*, 13(4), 78–85.

Haug, A. (2010). A software system to support the development and maintenance of complex product configurators. *International Journal of Advanced Manufacturing Technology*, 49 (1–4), 393–406.

Haug, A. (2012). The illusion of tacit knowledge as the great problem in the development of product configurators. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(1), 25-37.

Haug, A. (2013). Improving the design phase through interorganisational product knowledge models. *International Journal of Production Research*, 51(2), 626-639.

Haug, A., and Hvam, L. (2007). The modelling techniques of a documentation system that supports the development and maintenance of product configuration systems. *International Journal of Mass Customization*, 2(1/2), 1–18.

Haug, A., Hvam, L., and Mortensen, N.H. (2011). The impact of product configurators on lead times in engineering-oriented companies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 25(2), 197–206.

Haug, A., Hvam, L., and Mortensen, N.H. (2012). Definition and evaluation of product configurator development strategies. *Computers in Industry*, 63(5), 471-481.

Haug, A., Ladeby, K., and Edwards, K. (2009). From engineer-to-order to mass customization. Management Research News, 32(7), 633-644.

Hvam, L. (2006). Mass customisation of process plants. *International Journal of Mass Customisation*, 1(4), 445–462.

Hvam, L., Mortensen, N.H., and Riis, J. (2008). *Product customization*. Berlin/Heidelberg: Springer.

Hvam, L., Haug, A., Mortensen, N.H., and Thuesen, C. (2013). Observed benefits from product configuration systems. *International Journal of Industrial Engineering: Theory, Applications and Practice,* 20(5-6), 329-338.

Kristjansdottir, K., Shafiee, S., Hvam, L., Boven, M., and Myrodia, A. (2016). Quantification of benefits and cost from applying a product configuration system. In C. Forza et al. (Eds.), *Proceedings of 8th international conference on mass customization and personalization community of Europe (MCP-CE 2016).* http://www.mcp-ce.org/home/mcp-ce-2016/proceedings

Ladeby, K.R., and Oddsson, G. V. (2011). Reference frame for product configuration. In F. Piller, H. Chesbrough, and M. Tseng (Eds.), *2011 World Conference on Mass Customization, Personalization, and Co-Creation.* http://findit.dtu.dk/en/catalog/2389421561

Meredith, J.R., and Mantel, S.J. (2002). *Project management: A managerial approach*. Hoboken, NJ: John Wiley & Sons.

Møldrup, M., and Møller, N. (2004). Development and implementation of product configuration systems: A change management perspective. In K. Edwards, L. Hvam, M. Møldrup, N. Møller, L. Pedersen, and J. Riis (Eds.), *Proceedings of international conference on economic, technical and organizational aspects of product configuration systems (PETO)* (1-18). Lyngby, Denmark: Technical University of Denmark.

Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P., Lindschou, C. and Frobenius, S. (2010). Making product customization profitable. *International Journal of Industrial Engineering: Theory, Applications and Practice,* 17(1), 25-35.

Murray, J.P. (2000). Reducing IT project complexity: Information strategy. *The Executive's Journal*, 16(3), 30-38.

Myllärniemi, V., Tiihonen, J., Raatikainen, M., and Felfernig, A. (2014). Using answer set programming for feature model representation and configuration. In A. Felfernig, C. Forza, and A. Haag (Eds.), *Proceedings of the 16th international configuration workshop*. Graz, Austria: Graz University of Technology.

Myrodia, A., Kristjansdottir, K., and Hvam, L. (2017). Impact of product configuration systems on product profitability and costing accuracy. *Computers in Industry*, 88(1), 12–18.

Sabin, D., and Weigel, R. (1998). Product configuration frameworks: A survey. *IEEE Intelligent Systems and their Applications*, 13(4), 42–49.

Salvador, F., and Forza, C. (2004). Configuring products to address the customisation-responsiveness squeeze: A survey of management issues and opportunities. *International Journal of Production Economics*, 91(3), 273-291.

Shafiee, S., Hvam, L., and Bonev. M. (2014). Scoping a product configuration project for engineer-to-order companies. *International Journal of Industrial Engineering and Management*, 5(4), 207-220.

Shafiee, S., Kristjansdottir, K., Hvam, L., and Forza, C. (2018). How to scope configuration projects and manage the knowledge they require. *Journal of Knowledge Management*, 22(5), 982-1014.

Shafiee, S., Kristjansdottir, K., and Hvam, L. (2017). Automatic identification of similarities across products to improve the configuration process in ETO companies. *International Journal of Industrial Engineering and Management,* 8(3), 167–176.

Tiihonen, J., Soininen, T., Männistö, T., and Sulonen, R. (1996). State of the practice in product configuration: A survey of 10 cases in the Finnish industry. In T. Tomiyama, M. Mäntylä, and S. Finger (Eds.), *Knowledge intensive CAD: Volume 1*. Boston, MA: Springer, 95-114.

Tiihonen, J., Soininen, T., Männistö, T., and Sulonen, R. (1998). Configurable products: Lessons learned from the Finnish industry. In *Proceedings of 2nd international conference on engineering design and automation*. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.705.7178

Trentin, A., Perin, E., and Forza, C. (2012). Product configurator impact on product quality. *International Journal of Production Economics*, 135(2), 850–859.

Whitney, K.M., and Daniels, C.B. (2013). The Root Cause of Failure in Complex IT Projects: Complexity Itself. *Procedia Computer Science*, 20(1), 325-330.

Yin, R.K. (2009). *Case study research: Design and methods.* Los Angeles, LA: Sage Publications.

Zhang, L.L., and Helo, P.T. (2016). Empirical investigation on implications of configurator applications for mass customization. In V. Modrak (Ed.), *Mass customized manufacturing theoretical concepts and practical approaches*. Boca Raton, FL: CRC Press, 217–234.