MASTERGRAPHICS / White Paper

7 Myths of Configurators

Customer-facing configurators are powerful tools for high-variety manufacturers. Implemented properly, configurators can deliver significant ROI and process improvements. According to a recent white paper[1] from Lighthouse Transformations, documented returns from configurator implementation in a variety of industries include the following.

- 95% reduction in costs to complete an order
- Order cycle time reductions from 17-33 hours to less than 1 hour
- 20% to 50% reduction in engineering support for presales tasks
- Days Sales Outstanding (DSO), (a measure of how long it takes a customer to pay an invoice), decreased from an average of 60 to 29 days

Regrettably, not every company that implements a configurator sees these benefits. In some cases the problem lies in the execution of the project at the technology and process level. In too many other instances, however, the problem lies in the expectations of what the configurator will deliver. If the expectations of benefits are overestimated or unrealistic, the configurator will disappoint or be viewed as a failure, regardless of its actual performance. In addition, the configurator may not address the actual needs of the company in terms of configuration management, as the objectives it was designed to meet were based on erroneous assumptions.

Given the all too frequent mismatch of expectations with reality, it is not unusual to see a company make several attempts at configurator deployment before achieving some degree of success. The goal of this white paper is to help companies avoid expensive failures and inefficient investment of precious resources by identifying seven of the most common myths of configurators.

Types of Configurators

I recognize three types of configurators, each with its own objectives, user group, and infrastructure requirements.

1. *Customer-facing:* This configurator is used by a company's customers or sales force to design or configure the product. In many cases this configurator is deployed on the web. At a minimum the customer-facing configurator captures design intent, but may not (and it most cases, should not) capture detailed specification and engineering requirements.
2. *Engineering:* This is really not a type of configurator, but is included in this list due to misconceptions. It is based on knowledge-based engineering (KBE) principles, using engineering rules and algorithms. For project engineering, it is used to create an exact specification, BOM, and manufacturing documentation for a configuration. For product engineering, it is used for New Product Development (NPD) and product change orders.
3. *ERP:* Many ERP systems include a configuration module that is used to configure the product for manufacturing purposes. In some cases this may include a customer-facing interface, but this may also be just an internal manufacturing-driven application.

Not recognizing the differences between these types of configurators and the respective objectives underlying each

continued...

is one of the major points of confusion about configurators that factor into several of the myths.

This White Paper is focused on the customer-facing configurator.

One – A Configurator Is a Strategy

The statement "Acme Widgets will become the leader in its industry by deploying a configurator" is not what one usually sees as a statement of strategy, yet many companies act as if it were. It is not uncommon to hear an executive say, "Once we deploy the configurator (usually with a catchy trade name), our sales will go up, we won't see any more order errors, and our customers will see us as THE market leader."

The deployment of a configurator is a tactic, not a strategy. It will not solve or prevent problems with marketing assumptions, sales policies, or with the product itself. In addition, as more and more companies in all types of industries deploy configurators, customers are coming to expect the availability of configurator. Companies that do not have one will be considered more difficult to work with and not as competitive.

The customer-facing configurator is a mission-critical tactic for most high-variety manufacturers to successfully execute sales, marketing, and production strategy. It is also a necessity just to remain competitive in many markets. A company, however, cannot base its future growth and profitability (and survival) on the deployment of a configurator.

Two – The Primary Justification of a Configurator is Increased Sales

A configurator can facilitate the process of selling, but will, by itself, rarely increase sales or revenue. The company that deploys a configurator is still dependent on the quality of its product, the desire of the market to acquire that product, its ability to attract customers, and the ability of its sales force (or online sales process) to close the sale.

This lack of impact on sales by a configurator is especially true for complex products that require some level of "professional" or technical involvement in the configuration process. In these cases, part of the customer buying motivation is the perception that the company is the expert in this field and that the customer expects to leverage that expertise in the configuration process. An impersonal interaction with a configurator, no matter how "smart," does not take the place of the one-on-one discussion with the company's expert.

The dependence on increased sales as the justification for a configurator can lead to a death spiral of requirements for the configurator. The usual response to not seeing increased sales after deployment of a configurator is thinking that more options need to be presented to the customer, more configuration features need to be available, a more aesthetically-pleasing or attention-grabbing interface needs to be created, and so on. As this cycle proceeds, the configurator is either never fully deployed or collapses under unnecessary overhead, or customers are overwhelmed by the number of choices (see Myth Four below).

Justification for a configurator is best defined by a combination of sales and order management process improvements, sales and pricing policy enforcement, market penetration, product management, and lean manufacturing objectives. This combination of objectives is unique to every company and must be in line with the company's strategic objectives.

Three – Who the Sales Force Is Doesn't Matter

Many companies make the mistake of not considering who their sales force is when developing the requirements for the configurator. That is, they don't consider the geographical dispersion of the sales force, their level of computer literacy, or their level of product technical competence. A company may also fail to consider its sales policies when determining what functions and features the configurator should have as well as its level of complexity.

The issue of computer literacy is relatively straightforward in terms of what the impact is on a sales force being

continued...

able to use a configurator. (And, yes, there are still many issues with computer literacy in the workforce.) If a sales representative is not comfortable using the computer, he or she will not use the configurator, no matter how easy or attractive its interface is.

Technical literacy with the company's products is just as critical. Just providing a configurator will not, by itself, make every sales person a technical expert in configuration. There is no guarantee that the configurator will be used properly or that the configurations designed will be appropriate if it requires a level of technical expertise beyond that of the average sales person. This is not to say that you need to change the expectation for your sales people, rather, if you require the involvement of a sales engineer or other technical personnel at this time to ensure proper product specification, a configurator will not change that requirement.

The number of sales people and their geographical dispersion play into what features, functions and level of complexity should be in the configurator. As the number of sales people and the geographical territory covered increases, the complications of recruitment, support and deployment increase. First, the pool of top-level sales people who have engineering-level technical expertise is limited. Second, current technology makes it much easier to provide technical support, but for a sales force numbering in the hundreds, technical Help Desk support is still a time-consuming necessity. Finally, if the assistance required is functional, that is, for product configuration issues, the burden usually falls on limited engineering resources.

Sales philosophy also plays a part in the appropriateness of a configurator for a sales force. Sales philosophy can range from wanting the sales force to focus exclusively on the act of selling and building relationships to a technical sale where the sales person himself is responsible for technical design and support. Depending on the type of sales philosophy you require, you need to match the configurator to the needs of the selling process. An overly technical, complex configurator will frustrate the pure sales person; a too simple one will frustrate the technical seller.

All of these arguments apply to a configurator the customer uses himself. As with any application, you need to consider the user first and foremost in determining functions and features.

Four – Customers Want to Choose From Every Possible Option the Company Offers

One of the most common mistakes companies make when implementing a configurator is to put every possible option from the price book (or even when it's not in the price book) in the configurator. The belief is that customers will be most satisfied when they can choose exactly what they want from all possible choices. There are three main reasons as to why this is not a good strategy.

1. Humans tend to become confused when presented with too many choices. Jay Forrester of MIT, in his work on systems dynamics, found that 5 to 7 choices are the maximum that people can comfortably manage. More recent marketing and interactive research places the range at 3 to 5[2]. With confusion come irrational choices, bad choices, and "buyer's remorse;" and unhappiness with you as the product seller[3].
2. Your customers, as a rule, are not expert designers of your product. They want and need assistance from you, whether through interaction with a sales person or through a guided sales system, in order to make good and appropriate design choices. In addition, not all of the part options are appropriate for selection without a higher-level of intervention, whether for design, cost, delivery, profitability issues, or other reason.
3. Limitation of options is an important tool for market segmentation. If you do not have the capability of limiting option choices in the configurator based on market segment, you lose that critical marketing capability.

Configurators can be very important tools in terms of helping the sales force understand what is possible with the product range and providing bottom line competitive advantage. As with customers, however, sales people can also become confused when presented with too many choices. This is another reason for needing to really understand who your sales force is in order to correctly define configurator requirements and determine system architecture.

continued...

Five – Engineering or Configuring – Who Cares?

There is a significant difference between the concepts of configure-to-order (CTO) and engineer-to-order (ETO). There is a difference not just in the product characteristics, but in the sales process and tools required. As stated in the Types of Configurators section, the type of tool that engineering uses for an ETO product is usually a true engineering application combined with knowledgebase capabilities (knowledge-based engineering, KBE). One of the common mistakes, however, is deploying an engineering-level tool as a customer-facing configurator.

Just as too many options can confuse and frustrate customers and sales people, the inclusion of engineering requirements and functions in a configurator can also do so. It is very difficult, if not impossible, to capture and codify all the engineering rules required for specification of a complex product. That will lead, therefore, to the need for some level of human interaction to complete the configuration or specification process. If this is an expectation of the configuration, it forces the sales force to become engineering experts, a role which may not be a good fit.

Engineering requirements may also slow down or interfere with the configuration process, given the complexity of technical questions. If your sales process is such that the sales force focuses on developing a relationship with the customer, and the responsibility for project engineering is in-house, then the customer-facing configurator should focus on capturing design intent, not detailed engineering specifications. (The need to include some engineering algorithms or rules is definitely acknowledged, the point is, however, to keep them to a minimum.)

A typical concern in this scenario is, "but we need engineering for accurate pricing!" This underscores the need to fully review business processes with respect to strategic and tactical objectives and value-chain linkages before making configurator decisions. Detailed engineering specifications may be required for accurate pricing, but exactly when is that accurate pricing needed and who is the best at delivering it? In other words, the configurator needs to support and align with the best possible business process needed to close sales and deliver product.

Six – If It's Technically Possible, Build It Into the Configurator

There is a real temptation to try to make the configurator as technically (visually, business rules, offering breadth) robust as possible to be able to show the customer how sophisticated the company is. This rationale rarely delivers the expected returns while opening up other problem areas.

- As discussed in Myth Five above, there is a distinct difference between CTO and ETO. One of the dangers of trying to do too much in the configurator is that it morphs from a CTO application into an ETO application, resulting in a potentially serious misalignment between the application and your objectives and users.
- Visualization is usually a key feature of a configurator; it is very important, however, to not let it become the central focus of the application. Visualizations can be a sale facilitator but are rarely the primary reason for the sale.
- If a part is sold only a couple of times a year, does it warrant the level of development effort required to include it in the configurator? For some parts, the answer is yes. For others, however, the answer is no. This is especially true for parts with complex visualization, specification algorithms, or business rules. In addition, it can become more difficult to remove a part from your offering once that effort is put into incorporating it into the configurator, even if part rationalization analysis calls for its obsolescence.

Seven – A Configurator is an End-to-End Business System

Just as a configurator can be made too complex by incorporating too many design and engineering functions in it, it can also be burdened with too many business functions (and expectations). A configurator is a critical component of a complete enterprise business system and must, therefore, have its processes and data model aligned with the other components. Its two primary functions are capturing design intent and order information and then feeding that data into the other business system components which are better suited to support the order management, manufacturing, and financial processes.

There are three primary categories of problems associated with mistaking a configurator for an end-to-end business system:

- Trying to recreate capabilities that already exist in business system applications

continued...

- Generating a bill-of-materials
- Limiting scope and complexities

Recreating the wheel

There are many ERP, MRP, accounting, solid modeling, and PLM (product lifecycle management) applications that transform the outputs from a configurator into the financial, order, and manufacturing data needed to successfully deliver and invoice for an order. Given this plethora of choices, why should you try to recreate functionality that already exists and is well-tested in these applications? Data exchange standards are such that moving data from one application to another is an accepted reality of enterprise infrastructure.

Bills of Material

A common error in thinking with respect to configurators is the expectation that it will generate a bill-of-material (BOM). The first question that must be asked is which bill-of-material should it generate? The sales bill? The engineering bill? Or the manufacturing bill? Each of these is different in content and structure and is used during different parts of the order management and manufacturing process.

Second, the decision-making process required for configuration process does not necessarily depend on direct manipulation of the assemblies and parts that make up a product. If a manufacturing or engineering bill is incorporated in the configurator, the level of complexity increases dramatically and what should be a CTO (configure-to-order) application becomes an ETO (engineer-to-order) application. All of the concerns noted in Myth #6 arise then, as well as issues of scope and complexity, as noted below.

The third issue with respect to BOM expectations is product modeling best practices. In Lighthouse Transformation's White Paper, "The Business of Complexity Configuration: Understanding Product Modeling"[4], the following is stated as Product Modeling Principal No. 2: "Sales configuration will be limited to those product and service offerings that are required to be changed to specify the customer needs." It goes on to say, "This implies that part numbers and bills-of-materials should not be generated in the configurator unless it is absolutely required (emphasis added). If this practice is followed, downstream applications like BOM generators, shop floor systems, and purchasing application, typically do not affect the customer specification and can be executed without changing the sales configuration data."

Scope and Complexity

The scope and complexity of a configurator that tries to be an end-to-end business system is significant. The move from a CTO to an ETO application, by itself, is significant, with development scope usually expanding to 2 to 3 years and seven-figure budgets required. Adding other business system capabilities similarly increases scope and complexity. This reality reinforces the need to fully understand the objectives of the configurator and the processes it needs to support in order to avoid unnecessary development investment and application complexity.

Conclusion

Whether you purchase a configurator application (and there are many excellent choices in the marketplace) or choose to develop your own, there are really only three key points to remember for a successful implementation.

1. Develop realistic business objectives for the configurator and make sure that it meets them.
2. Know your sales force and customers.
3. Keep it simple! You can always add more functionality and complexity, but it is very difficult to remove it once deployed.

There are no absolute truths, however, when it comes to configurators. Each implementation must be defined by its objectives. Your deployment success will be determined by your own careful consideration of your product, your business, and your staff.

continued...

THE AUTHOR

Stephanie N. Green, Target Account Manager, Manufacturing, with MasterGraphics, has twenty-plus years of experience with configurator projects. She is a recognized expert in high-variety manufacturing (HVM) strategy and tactics, with a special emphasis on managing value chain complexity. Ms. Green, founder of the High-Variety Manufacturing Professional Community, http://highvarietymfg.ning.com, is a frequent participant in industry forums, and is regularly asked to collaborate on professional publications.

ABOUT MASTERGRAPHICS

MasterGraphics helps businesses focus on innovation by enhancing their design and post-design processes to reap bottom-line benefits via increased productivity, cost effectiveness, superior quality control and enhanced operational efficiency. We help you innovate by taking your design data further: We make it work.

mastergraphics.com
mastergraphics@mastergraphics.com
(800) 873.7238

[1] "Mitigating Risk in a Quote to Order Project", Lighthouse Transformations, www.lighthousetransformations.com, White Paper

[2] Synchronization, Mark Warren

[3] A recent book, Nudge, by Richard H. Thaler and Cass R. Sunstein, is a very interesting treatise on choice architecture and the irrationality of human choice behavior.

[4] "The Business of Complex Configuration: Understanding Product Modeling", Lighthouse Transformations, www.lighthousetransformations.com