

Book Everything

By Paul Magel, Senior Vice President

BlueCherry Application Solutions at Computer Generated Solutions, Inc.



Paul Magel

Sometimes it's important to recognize that change is good.

In the course of implementing BlueCherry 120+ times over the past few years, we have occasionally been faced with similar questions from our customers who ask how they can leave certain of their business practices unchanged with the new system when, in truth, it doesn't make sense to do so. A case in point is the issue of not booking orders once the entire inventory/WIP quantities have been called for. I submit that this is not the ideal way to run the business.

First, some details. We have seen a number of companies that refuse to accept sales orders once their available goods have been sold out. In itself, this is a completely understandable notion—if I have 1200 units of dress style 2000-EGG in production and orders for 1198 have come in by today, I could reasonably take the position that I will sell two more and then that's it—whoever calls after that simply won't get any 2000-EGG. This is what's generally known as a first-come, first-served premise, and, again, it is totally reasonable and understandable.

Companies that follow this business philosophy have traditionally set their manual or older computer systems up in such a way that the order entry clerk cannot even add a "sold-up" item to a sales order. While it is easy to see the basic thinking behind this logic, it is also important to get a deeper understanding of why this approach was taken "in the old days."

Basically, the reason this stringent approach has historically been used has to do with the limitations of the systems (people and computers) involved. That is, in every case we have seen, the logistics of the customer service department were such that people needed to maintain records manually to track things like availability. Since people are fallible, the concern was not that a popular style would be oversold per se, but that nobody would be aware of that situation until time came to actually ship the goods.

Thus, the real reason behind the idea of simply denying the user the ability to enter a sales order for style 2000-EGG once it was sold out was the fear of angering the customer, who typically would not be told the dresses are not coming the very week they were supposed to arrive thanks to the lack of proper information.

Which brings me to the explanation we give to those customers so they understand why not booking potential sales orders is actually a sub-optimal business practice.

First, you need to recognize the benefit of capturing these potential sales, even if you are never going to fill them. This additional data, when mixed and sorted with the data of the sales orders you can fill, becomes real information that leads to valuable knowledge. Specifically, if you don't have a record of

how much you could have sold then you will not really have a good idea of what to plan for next season.

Using my example of 2000-EGG above, say it was far more popular than anticipated, and while 1200 were produced in total customer orders came in for a total of 4000 units. By not booking the additional 2800 pieces worth, one loses information that could prove extremely valuable in the future. Specifically, estimating production requirements for the next season or year of how many similar dresses should be produced will be based on the 1200 booked, not the 4000 demanded this time around.

The point is there is real value to the information associated with the orders that could not be filled, and for this reason it is always desirable to capture all demand for a given product.

Once we explain the data repercussions of booking all sales into BlueCherry, the question we are asked is "how do I maintain my first-come first served philosophy if I keep taking orders?" And the answer to that is simply "by setting up the allocation rules and sorts appropriately."

That is, the user sets the sort sequence used by the allocation process; it is another parameter in BlueCherry. By choosing which factors should be included and in what order they are sorted, the system becomes tailored to reviewing the sales orders in the progression required by the company. By putting Priority (Order) Date above the customer Priority Code or the order Cancel Date, you are programming BlueCherry to give orders that arrived sooner the top positions.

The next question is invariably "how do I know which orders cannot be filled?" Since BlueCherry assigns specific codes to each order line during allocation based on its findings, reports can be run (automatically) to display orders that will not be filled (or be late, etc.). This means that Customer Service can contact those customers who will not be receiving 2000-EGG hours after the orders arrived, rather than not being able to know until the week the order is due to ship.

Meanwhile, since the allocation applies to all orders in the system, those that arrived via EDI are also included in the mix. In a world where customer service is not allowed to hand-enter orders when a style is oversold, the issue of dealing with EDI orders becomes another manual process completely. By simply booking everything and getting daily updates as to who will or won't be shipped, one can handle all demand in the same manner, notify customers virtually immediately if they can't get what they want, and still capture all of the data that next year can become quite useful for planning purposes.

It is generally agreed that knowledge is power; by leaving sales demand out of the system one forgoes some real power.

Paul Magel is Senior Vice President for BlueCherry Application Solutions at Computer Generated Solutions, Inc. (CGS). For more information, visit www.cgsinc.com.