Interfaces

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Intel Calculates the Right Service Level for Its Products

Dennis Arnow, http://orcid.org/0000-0001-6340-1192Sean P. Willems

informs
http://pubsonline.informs.org/journal/inte/

INTERFACES
Articles in Advance, pp. 1–4
ISSN 0092-2102 (print), ISSN 1526-551X (online)

Intel Calculates the Right Service Level for Its Products

Dennis Arnow,[a] Sean P. Willems[b]

[a] Independent Consultant, San Carlos, California 94070; [b] Haslam College of Business, University of Tennessee, Knoxville, Tennessee 37996
Contact: dennisarnow@gmail.com (DA); swillems@utk.edu, http://orcid.org/0000-0001-6340-1192 (SPW)

Received: April 25, 2016
Revised: January 25, 2017
Accepted: February 20, 2017
Published Online in Articles in Advance: May 18, 2017

https://doi.org/10.1287/inte.2017.0893

Abstract. All too often, companies do not rigorously calculate service levels. Instead, they arbitrarily set service levels by employing a top-down mandate. They employ this arbitrary approach because they have difficulty in quantifying the economics of a specific service level; the primary difficulty these companies encounter is quantifying the cost of not satisfying demand. Intel's Customer Fulfillment and Logistics Group has developed a data-driven approach to calculate customer service levels. The major breakthrough in this work is a simple three-step process that diverse functions across the supply chain can employ to agree on the costs associated with a given service level.

Keywords: inventory • service • supply chain application

Introduction

Founded in 1968, Intel introduced the world's first microprocessor in 1971. One way of describing Intel's evolution as a company is to generalize the 1980s as the era in which design was the constraint, the 1990s as the era in which manufacturing was the constraint, and the 2000s as the decade in which supply chain performance is the constraint. As we focused on ways to improve supply chain performance, we found that Intel's existing methods for choosing service levels were inadequate. While Intel employees were guided by strong business intuition and a desire to do what was best for the business, the company could not rigorously quantify the trade-off between the costs and benefits of a given service level. The old process Intel followed selected a service level in an ad hoc fashion. Our breakthrough, which allowed us to achieve consensus on product-specific service-level targets, established a data-driven process that quantified the cost associated with a given service level.

Calculating Implementable Service-Level Targets

Calculating a product's service level is a three-step process.

Step 1: Calculating Inventory-Driven Costs

The costs associated with providing a specific service level are well understood. These inventory-driven costs involve traditional holding costs tied to the cost of capital plus component devaluation, price protection, and product return and obsolescence costs that are associated with holding product in advance of realized demand. Callioni et al. (2005) provide a detailed discussion of inventory-driven costs.

Step 2: Calculating Stockout Costs

The costs associated with not making a sale are less clear. When a customer asks Intel for a specific product that is not available, any of these three outcomes are possible: lost sales, deferred sales, or sales substitution.

A *lost sale* is the most obvious outcome of not providing supply in response to a customer order. However, the lost-sales outcome does not always result in losing the entire margin from the sale. For example, if Intel does not have the product in stock, the customer might go to an electronics distributor for the product. Purchasing from an alternative channel, such as a distributor, has at most a marginal revenue impact to Intel because Intel records revenue from selling the product to the distributor. However, in some of Intel's market segments, customers have alternative suppliers providing form-fit-function drop-in replacements, and a lost sale at Intel can indeed result in a pure lost sale for customers ordering those products; Intel calls this as a walk-away scenario.

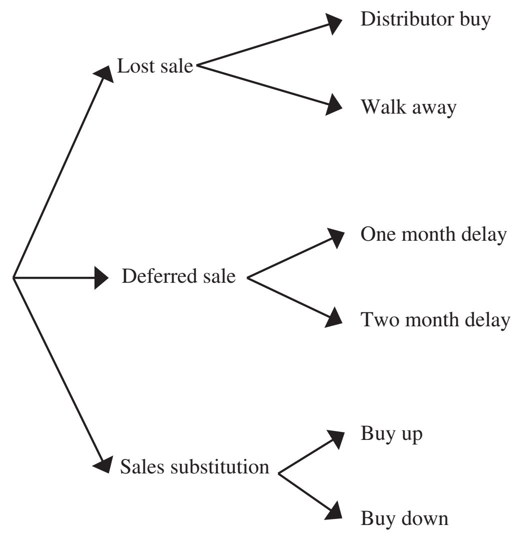
A *deferred sale* occurs if the customer will wait for the product to become available. For certain products, especially those for which a form-fit-function replacement is not available, or perhaps because the true delivery schedule is negotiable, the customer will not be happy to learn that Intel is out of stock, but is willing to wait for the product to become available. In this case, revenue for the current purchase is delayed. This delay creates a time-value-of-money cost to deferred sales. In most cases, the sale is deferred for 30 or 60 days.

Sales substitution occurs when the customer and Intel mutually agree to substitute the initially requested product with one that is similar in nearly all characteristics except price and performance. Sometimes the customer agrees to buy an alternate product that has a higher price than the one initially requested; we consider this a buy-up scenario. Sometimes the customer agrees to buy an alternate product that has a lower price; we consider this a buy-down scenario. In comparison to the initial product, a buy-up sale generally results in higher net margin, while a buy-down sale generally results in lower net margin from the sale.

A product's expected stockout cost is more difficult to calculate than its expected inventory-driven cost. Inventory-driven costs are straightforward functions of the inventory level and the supply chain's operating parameters, such as demand variability, supply variability, and lead time. For all the vagaries introduced by variability and time, each inventory-driven cost is self-contained and predictable in its behavior. In contrast, for any given stockout outcome, numerous scenarios, which can be quite complex and have far-reaching implications, are possible. For example, a customer who cannot buy a low-margin Intel motherboard chip set might decide to cancel a high-margin microprocessor. Perhaps the customer would be angry enough to never order again from Intel in the future. While a product's inventory-driven costs almost surely must be less than the product's gross margin, which is simply a rational requirement for Intel to offer the product for sale, the stockout cost for some outcomes might greatly exceed the gross margin for that product. This type of situation can paralyze a business team trying to calculate stockout costs.

To move this from an intractable and unquantified argument to a tractable and quantifiable solution, we employed a deceptively simple two-step process to calculate stockout cost. First, for each of the three stockout outcomes, we quantified the cost of the outcome's associated scenarios. Second, we assigned a probability to each outcome and then to each scenario associated with an outcome. Once we knew the costs and

Figure 1. Stockout Cost Is Calculated for a Product by First Determining the Relevant Outcomes That Can Occur During a Stockout and Then Determining the Scenarios Associated with Each Outcome



Notes. The walk-away and buy-down scenarios are often the most expensive. A deferred sale is not costly, and a buy-up sale can often increase Intel's margin from that sale.

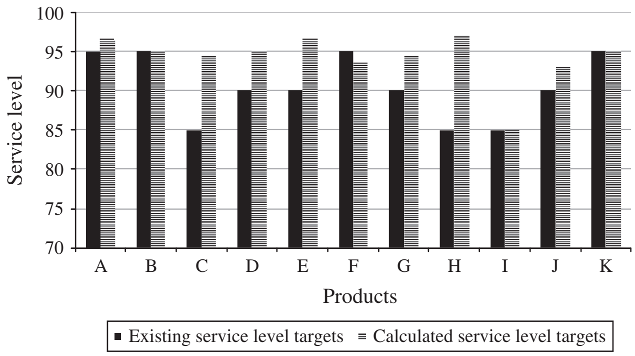
associated probabilities, calculating the expected per-unit stockout cost became a simple exercise. Oral et al. (1972) provide a useful decision-tree framework to calculate expected stockout cost in exactly this manner. Figure 1 includes a decision tree that represents the most common outcomes and scenarios encountered at Intel.

Step 3: Calculating the Correct Service Level

The first two steps reduce inventory-driven costs and stockout costs into a common unit of measure: dollars per unit. The calculated service level, which we present in the appendix, strikes the right trade-off between inventory-driven cost and stockout cost.

In practice, the calculated service-level target for most Intel products increased. Although a few products required a lower service-level target, most service-level targets increased by a few percentage points, but some increased almost 20 percentage points (i.e., from a service-level target of 80 percent to one that exceeded 99 percent). On average, the increase in one Intel business unit was an absolute increase of five percentage points. Figure 2 shows a before-and-after snapshot of

Figure 2. The Chart Shows Existing and Calculated Service-Level Targets for 11 Products



Note. For 7 of the 11 products, the calculated service level is higher than the existing service level.

a set of Intel products, rank ordered by volume, that employed this approach.

Prior to this work, Intel employed a rule-of-thumb service-level target based on product attributes. Products with a 95 percent service-level target were sold to multiple top-tier customers, while products with a 90 percent service-level target supplied a single top-tier customer. Products with significant sales in distribution (i.e., where the channel typically had inventory that could satisfy end-customer needs) maintained an 85 percent service level.

Learnings

We found adopting the following five guiding principles to calculate stockout cost to be useful.

First, *consider only near-term costs*. After a few minutes of discussion, even the most numbers-oriented planners and analysts found themselves contemplating long-term implications of not satisfying demand. We cut this conversation off and focused on the immediate costs associated with not having the item in stock. Although our approach underestimates the true cost of a stockout, we employed the concept that near-term costs are more quantifiable and "real" than long-term costs. Furthermore, in practice, these near-term costs are significant enough to allow us to determine the right real-world answer.

Second, *consider only first-order effects*. Not only would people contemplate cost ramifications far in the future, but their thoughts would commonly wander into situations that were far from obvious or far from likely to be realized. For example, a customer who wants to buy a microprocessor that is not in stock might develop such antipathy for Intel that he or she would not buy a smartphone because it also uses an Intel microprocessor. While these situations might be true, we kept our focus on the most measurable and quantifiable scenarios that were directly attributable to the product being analyzed.

Third, *consider only dominant modes*. We limited the team to developing no more than five scenarios per outcome. Thus, the team could focus on the most likely cost implications of a given stockout outcome.

Fourth, *disassociate costs from service level*. The per-unit stockout cost likely varies as the service level decreases. For example, if service levels are generally quite high, there could be few lost sales because customers will generally wait or purchase the product from a distributor. However, if service levels are generally quite low, the customers will routinely find an alternative non-Intel source of supply. These two alternatives have different cost implications to Intel. Our approach of asking people for a reasonable estimate of the scenario's cost worked well in helping us to develop an effective lower bound on the cost.

Fifth, *survey broadly to determine the likelihood of each scenario*. To estimate the likelihood of each scenario, we surveyed customers and customer-facing account representatives, and we solicited upper and lower bounds from experienced managers. Although these surveys are susceptible to bias, we found a surprising uniformity in the answers across the organization and functional areas. This gave the team confidence that the ultimate answer would at least be traceable to generally accepted intuition. Once we coupled this with sensitivity analysis, the team could let senior management know how important (or not) the accuracy was. In cases in which we found a wide range, which influenced the final solution, we implemented a target that used the most conservative value and then revisited the product several months later to determine if the scenario estimates required revision in the face of actual performance.

Conclusion

This paper describes a data-driven approach Intel has adopted to determine the right service level for products across its business units. This model was first applied at Intel in 2006 and has remained in continuous use for the past 10 years.

The approach begins by calculating inventory-driven costs and stockout costs. The first cost reflects the inventory cost associated with providing a specific service level; these costs have traditionally been modeled by the operations community and are relatively easy to develop. The second cost quantifies the stockout cost associated with a given service level. Due to the emotion tied to service and the fear of not providing sufficient stock on hand, these costs have traditionally not been clearly quantified; however, once a framework is established, quantifying these costs is straightforward.

Appendix

As we note in the body of the paper, inventory-driven costs are the easier costs to define. Wieland et al. (2012) documents the inventory practices at one of Intel's business units, while Manary et al. (2009) documents how Intel adjusts the standard single-stage inventory equation to accommodate forecast bias.

This section details the stockout calculation for a single product. We have removed any product-specific notation for ease of presentation. We let O denote the set of outcomes for the product and S_i the set of feasible scenarios for the ith outcome; x_{ij} denotes the cost, and p_{ij} denotes the probability, of outcome i's scenario j. Equation (A.1) formulates the expected stockout cost:

$$\text{Stockout cost} = \sum_{i\in O}\sum_{j\in S_j} p_{ij}x_{ij}. \tag{A.1}$$

Under Type II service, a heuristic to determine the service level, as Hadley and Whitin (1963, Sections 4-2) discuss, employs an iterative solution that solves Equation (A.2) and an equation for the order quantity.

$$\text{Service level} = 1 - \left(\frac{\text{inventory-driven cost}}{\text{stockout cost}}\right) \cdot \left(\frac{1}{\text{number of cycles per year}}\right). \tag{A.2}$$

At Intel, we treat the production quantity, and hence the number of cycles per year, independently of service level, because of the cyclic schedule maintained to maximize wafer fab utilization. Therefore, Equation (A.2) is solved only once to determine the calculated service level.

References

Callioni G, de Montgros X, Slagmulder R, Van Wassenhove LN, Wright L (2005) Inventory-driven costs. *Harvard Bus. Rev.* 83(3): 135–141.

Hadley G, Whitin TM (1963) *Analysis of Inventory Systems* (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Manary MP, Willems SP, Shihata AF (2009) Correcting heterogeneous and biased forecast error at Intel for supply chain optimization. *Interfaces* 39(5):415–427.

Oral M, Salvador MS, Reisman A, Dean BV (1972) On the evaluation of shortage cost for inventory control of finished goods. *Management Sci.* 18(6):B-344–B-351.

Wieland B, Mastrantonio P, Willems SP, Kempf KG (2012) Optimizing inventory levels within Intel's channel supply demand operations. *Interfaces* 42(6):517–527.

Verification Letter

Alison Friday, Customer Fulfillment, Planning and Logistics Group (CPLG), Intel Corporation, Hillsboro, OR 97124, writes:

"I am writing to verify that the paper by Dennis Arnow and Sean Willems, titled 'Intel Selects the Optimal Service Level for its Products,' documents work that was successfully applied at Intel. Prior to this work, Intel's Customer Fulfillment and Logistics Group struggled to determine an optimal service level. The methodology developed in the paper was able to align the business on the costs and benefits of proving a specific service level. In particular, we saw that products fell into two classes: products that should have a lower service level and products that should have very high service level. The paper talks through the attributes of each product type, and the sensitivity analysis employed to help the team become comfortable with these results. By 2006, this was applied across 150 products in the business, and is still in use today."

Dennis Arnow has held supply chain leadership roles in a wide range of industries: manufacturing, semiconductor, consumer electronics, and startups. He was with Intel for over 10 years. After Intel, Dennis was VP Supply Chain at Logitech and VP Global Supply Chain at Atmel where he applied analytical methods to supply chain problems. He has been consulting since 2015. He earned his BS, MS, and MBA from MIT.

Sean P. Willems is the Haslam Chair in Supply Chain Analytics at the University of Tennessee's Haslam College of Business. His research focuses on supply chain design and optimization problems. Sean received his BSE in decision sciences at The Wharton School, University of Pennsylvania, and his master's in operations research and PhD in operations management from MIT.