

How to Measure Customer Satisfaction

Murali Chemuturi

Introduction

Project-based organizations place a lot of emphasis on customer satisfaction, and rightly so, as customer satisfaction is the key for improving these companies' internal processes. A *customer satisfaction rating* (CSR) is often obtained through a questionnaire—the *customer satisfaction survey* (CSS). This method, however, suffers from the drawback of customers likely being emotionally influenced while filling out these questionnaires.

Naomi Karten, an expert on the subject of customer satisfaction (www.nkarten.com), states in her seminar *Tales of Whoa and the Psychology of Customer Satisfaction*: “People tend to rate service higher when delivered by people they like than by people they don't like.” Karten also goes on to describe what one can do to be “likable.” More often than not, Karten contends, the CSS rating received from the customer represents *perceived* feedback rather than impartial feedback.

This is not to say that companies do not get any value from customer-filled CSR forms. But they must recognize that responses can be emotionally based, and that the customer is not one person, but an organization—meaning multiple people. While so, only one person represents the organization and fills out the survey. Would this person consult all concerned before filling it out? Ideally, he or she should, but often, he or she will not.

This gives rise to the need for a way to compute a CSR based on internal data—data that is free from bias and that gives a realistic metric on customer satisfaction.

Why Should We Measure Customer Satisfaction with Internal Data?

Consider the following three scenarios:

1. The customer is pragmatic and not swayed by influences like the recency factor and the one-incident factor, prejudices of any kind, poor judgment, or personal stake. This customer keeps meticulous records of the project execution and is expert at data analysis. While it may be rare to have such a customer, his rating is likely a true reflection of the vendor's performance.
2. The customer is an average person. His rating is influenced by some of the factors mentioned in the first scenario. Let us assume that he rates the vendor's performance as poor. If this low rating (which is biased) were accepted, the personnel involved in the project execution would also receive low ratings in the organization as a result. They might, in turn, receive lower hikes (salary increases) and bonuses, if any at all. This would de-motivate these workers, as it is possible that they in fact did a fairly good job and merit a better rating.

3. The customer is an average person. His rating is influenced by some of the factors mentioned in the first scenario. Let us assume that he rates the vendor's performance as high. As a result, the personnel involved in the project execution might receive better hikes and bonuses. Such a situation would further de-motivate the personnel from the second scenario.

Scenarios two and three give rise to the phenomenon known as “rewarding the under-performers and punishing the better performers”—a disastrous situation for any organization. An impact even more disastrous is that the organization does not have a realistic picture of how satisfied its customers really are. In such a situation, any efforts to improve customer satisfaction would be taken in the wrong direction.

Aspects Critical to Customer Satisfaction

I have been using the following method to compute a customer satisfaction metric, based on internal data, in all the organizations to which I have provided consulting services. I developed this system through reverse-engineering of the vendor-rating metric that manufacturers use to rate their suppliers. The method is based on the five following parameters I believe are critical to customer satisfaction, which are *tangible* aspects that can be measured objectively.

1. Quality

Quality comes first. The dictum “customers forget the delays but not the quality” aptly states the value of quality. Furthermore, customers forget everything else if—and only if—the quality delivered is superb.

2. On-time delivery

Nothing irritates a customer more than not receiving a delivery on the promised date. When a delivery is late, plans at the customer's end have to be redrawn, resource allocation has to be shifted, and all subsequent actions have to be rescheduled, causing the customer a lot of inconvenience.

3. Money

This refers to money the customer is paying. It is not uncommon for escalation clauses to be built in to contracts. When the vendor chooses to apply an escalation clause and to bill more money, it greatly inconveniences the customer. The customer must obtain sanctions and approvals for the extra payout, as well as answer quite a few questions in the process. In short, price escalations irritate customers.

4. Issue factor

Most projects have issue resolution mechanisms (methods to solve problems). Some vendors, in their eagerness to always interpret the specs accurately (and in their fear that they might in fact misinterpret specs), raise more issues. When valid issues are raised, the customer is usually more than happy to resolve them. But when the issues raised are trivial, the customer becomes annoyed.

5. Accommodation and cooperation

Few projects are ever completed without changes having been requested by the customer. When the customer requests a change, the vendor should accommodate and cooperate with the customer, and implement the change without postponing the delivery and without increasing the price.

Quality Rating

No project is ever perfect, and most times, defects may not be detectable immediately upon delivery. If defects are detected during the warranty period, the customer is happy. However, what is important is whether the defects fall into an acceptable range. Usually, the customer's expectation is "zero defects," but all professionals on quality know that "zero defects" is rarely achieved.

Sometimes, customers specify what is an acceptable *defect density* (number of defects per number of opportunities for error); other times, the defect density is implicit. Customers select vendors based on their certifications or market reputation. But reputation alone does not lend itself for measurement. Using six-sigma philosophy, we can measure and specify the expected defects based on the "sigma level" of the vendor organization.

If an organization is at 6-sigma level, then the expected defects from that organization total three defects for every million opportunities. If the organization is at 5-sigma level, the expected defects total three defects for every 100,000 opportunities. At 4-sigma level, three defects for every 10,000 opportunities. At 3-sigma level, three defects for every 1,000 opportunities.

The expected number of defects delivered should be contrasted against the actual number of defects delivered.

Defects begin to be counted during the acceptance testing stage because they can be discovered by the customer just as they can be in pilot-runs, during live or production runs, throughout the warranty period, and afterward.

Normally, defects are classified as one of three categories: critical, major, and minor. I use only the critical and major defects, since minor defects can sometimes merely be a difference in perception—the customer may perceive as a defect what the vendor may not consider a defect.

The defect density is computed as defects per unit size, or conversely, as units of product per one defect. The size is usually measured as *lines of code* (LOC), *function points* (FPs), or any other size measure used in the organization. What is important is to select one size and use it for all measurements.

Here is the formula to compute a *quality rating* (QR) for customer satisfaction:

$$\mathbf{QR = (actual\ defect\ density - accepted\ defect\ density) \div accepted\ defect\ density}$$

If the actual defect density is less than the accepted defect density, then this metric will be negative, meaning customer expectations have been exceeded. If the actual defect density is the same as the accepted defect density, then this metric will be zero—customer expectations have been fully met. If the actual defect density is more than the accepted defect density, then this metric will be positive, and it means customer expectations have not been fully met.

Delivery Schedule Rating

Nothing is more frustrating than not receiving a delivery on an agreed-upon day. This frustration may be eased if somebody calls to tell you that the delivery is going to be delayed, but the frustration is there just the same. The funny part is, even if a delay is the result of a change that the customer requested, late delivery still frustrates the customer. It is as if the customer is thinking, “Can’t they accommodate this teeny-weeny change without postponing the delivery date? Vendors always take any opportunity to delay delivery!”

Oftentimes, vendors prefer to compromise on quality than to delay delivery. The philosophy is this: it will take some time for the customer to unearth the defect, but it takes no time for the customer to come down heavily if delivery is not on time. Excuses like “Sorry for the defect; here is the corrected version” or “In our fervent efforts to deliver on time, this defect crept in” can be quite convincing.

Customers might forget delayed deliveries, but they seldom forget poor quality. When asked for references, they normally highlight the quality a vendor provides over on-time delivery. That is the reason I place this aspect as second in importance when determining customer satisfaction.

To compute this metric, we contrast accepted delivery with actual delivery. But which date should you use as the accepted delivery date? To compute the highest rating possible, take the latest accepted delivery date. To derive a true customer satisfaction rating, then take the date that is on the purchase order. Some organizations use both—one for internal purposes and one for the external purposes.

The formula for computing a *delivery rating* (DR) for customer satisfaction is as follows:

DR = (actual days taken for delivery – accepted days for delivery) ÷ accepted days for delivery

To determine actual days taken for delivery, use the number of calendar days between the date of the purchase order and the date on which delivery was actually effected. To determine the accepted days for delivery, use the number of calendar days between the date of the purchase order and the date of delivery specified on the purchase order.

If actual delivery was made before the accepted delivery date, then this metric will be negative, meaning customer expectations have been exceeded. If actual delivery was made on the accepted delivery date, then this metric will be zero—customer expectations have been fully met. If actual

delivery was made later than the accepted delivery date, then this metric will be positive, and it means customer expectations have not been fully met.

Price Rating

Obviously, no vendor can bill the customer for an amount that was not agreed to by the customer—that is if the vendor expects his invoice to be respected in full and without issue. Why is this an important factor? Because sometimes contracts are drawn up using an hourly rate with a maximum amount, allowing some variance on either side. In such cases, the final billed amount can either be lower or higher than the specified amount.

When a price escalation clause is implemented or an additional payment is requested against a change, some negotiating usually occurs before the customer accepts the escalation; the amount accepted might not be the same as requested by the vendor. The fact that extra money is being requested and the resultant negotiations can certainly frustrate the customer.

Whenever the customer has to pay more than the purchase order value, the customer is dissatisfied. Needless to say, the customer is certainly pleased when the vendor charges less money than the amount specified on the purchase order.

To compute the *price rating* (PR), use the price agreed to (before taxes) on the original purchase order and the final amount billed. Here is the formula for computing customer satisfaction in this area:

PR = (actual amount billed – amount on the purchase order) ÷ amount on the purchase order

If the actual amount billed was less than the purchase order amount, then this metric will be negative, meaning customer expectations have been exceeded. If the actual amount billed was equal to the purchase order price, then this metric will be zero—customer expectations have been fully met. If the actual amount billed was more than the purchase order price, then this metric will be positive, and it means customer expectations have not been fully met.

Issue Rating

Issues crop up during project execution mainly because of unclear specifications or a lack of understanding the specs. Issues may also occur because of a conflict or an error in the requirements.

When the vendor raises an issue whose origin is attributable to the customer, the customer's satisfaction is not usually affected. However, the customer's satisfaction does become affected if the issues raised are due to the vendor's improper understanding of the requirements. Customers expect any shortfall in exhaustive requirements specifications to be bridged by the vendor. Failure to meet these expectations cause dissatisfaction in customers.

To compute an *issue rating* (IR), use the *issue density* (ID). While we can easily compute actual ID, there is no accepted measure for an acceptable ID. We also use software size for computing ID. While issues can directly relate to requirements, we cannot use the number of requirements, as the method for defining requirements can vary the number significantly.

Thus, the ID is computed as follows:

$$\text{ID} = \text{number of issues raised} \div \text{software size}$$

Software size can be any software size measure, such as LOC or FP. Since there is no universally acceptable ID, an organizational standard should be defined and continuously improved.

The formula for computing IR for customer satisfaction is as follows:

$$\text{IR} = (\text{actual ID} - \text{standard ID}) \div \text{standard ID}$$

If the actual ID was less than the standard ID, then this metric will be negative, meaning customer expectations have been exceeded. If the actual ID was the same than the standard ID, then this metric will be zero—customer expectations have been fully met. If the actual ID was more than the standard ID, then this metric will be positive, and it means customer expectations have not been fully met.

Cooperation Rating

Most projects would not be complete without a few change requests from the customer—software maintenance projects run on these. But since change requests are commonly implemented before delivery, how then do they give rise to customer dissatisfaction?

Change requests cause additional work for the vendor, and their impact is felt in two ways: revised delivery schedule and higher cost. In some cases, the vendor absorbs both, and in others, the vendor absorbs the impact on price only and passes the impact on delivery schedule on to the customer. Still in other cases, the vendor absorbs impact on delivery schedule and passes on the impact on price to the customer. In the remaining cases, the changes are rejected.

Of course, the customer is happy when change requests are accepted without impacting the price or the delivery schedule. But since this does not always happen, that is why we compute a *cooperation rating* (CR), the formula of which is the following:

$$\text{CR} = (\text{no. of change requests received} - \text{no. of change requests implemented without affecting delivery date or price}) \div \text{no. of change requests received}$$

If the number of change requests received were the same as the number of change requests implemented without affecting either delivery schedule or price, then this metric will be zero, meaning customer expectations have been fully met. If the number of change requests received

were greater than the number of change requests implemented without affecting either delivery schedule or price, then this metric will be positive, and it means customer expectations have not been fully met.

There is no way to exceed customer expectations in this rating.

Composite Customer Satisfaction Rating

Having computed all five ratings critical to achieving customer satisfaction, we are ready to compute the *composite customer satisfaction rating* (CCSR).

Obviously, all five ratings do not carry the same importance in achieving customer satisfaction. These ratings can also vary from organization to organization, and from customer to customer. Some customers may perceive quality as being the most important aspect of a product or a service, while some may perceive delivery as the most important aspect. Still for others, the highest of importance might be placed on price. Given these differences in customers' perceptions and preferences, it is necessary to assign weights to each of the five ratings in order to arrive at a reasonably accurate CCSR.

The sum of all the weights must equal 1.00 in order to calculate a meaningful CCSR. Table 1 shows an example of how weights can be distributed.

Serial Number	Rating	Weight
1	Quality Rating (QR)	w1 = 0.30
2	Delivery Rating (DR)	w2 = 0.30
3	Price Rating (PR)	w3 = 0.30
4	Issue Rating (IR)	w4 = 0.05
5	Cooperation Rating (CR)	w5 = 0.05
	Total weight	1.00

Table 1. Example of apportioned weights among the five ratings.

The formula to compute CCSR is this:

$$\text{CCSR} = 5 - (\text{QR} \cdot w1 + \text{DR} \cdot w2 + \text{PR} \cdot w3 + \text{IR} \cdot w4 + \text{CR} \cdot w5)$$

This formula gives the CCSR on a 5-point scale. It is possible for the CCSR to be greater than 5 in some cases. When this happens, it means that customer expectations have been exceeded.

Use of CCSR

While I do not advocate doing away with CSSs altogether (ultimately, what the customer perceives is also important), consider these facts: Only one person in a customer organization fills out CSSs, despite the fact that many people in the organization may use the product. This one person's expectations can be managed, making it possible to calculate an accurate rating. But the other users (some of whom could be decision makers) can certainly still unearth the defects in the product.

This is to say that perception-based ratings alone cannot be relied upon. Contrasting CSS ratings with CCSR allows organizations to improve their processes.

Suppose that the internal CCSR is in agreement with a CSS rating. This means that the customer's perception is in sync with reality, and that customer expectations are being managed as they should be. The organization's strengths are equal in service and expectation management, giving a realistic picture to management. In this case, the organization needs to take corrective action based on the rating should it be poor.

Suppose that the internal CCSR is way below the CSS rating. This means that the customer's perception of an organization's service is better than the service is in reality. This is not of any benefit to the organization, because if it continues to praise itself based on the customer's *perception* that its level of service is high, then the organization will head toward decay. Resources will continue to place emphasis on expectation management rather than on service, thus never improving services. In this case, resources need to be trained in order to improve service.

Now suppose that the internal CCSR is way above the CSS rating. In such a case, the customer's perception of an organization's service is poorer than the service is in reality. This shows that the organization is concentrating on service without any concern for expectation management. Interpersonal relations and communication with the customer are being neglected. Here, resources need to be trained in expectation management.

There is scope in the CCSR method for organization-based adaptation. Some of the five ratings may be dropped or substituted, or new ones may be added to suit the specific organization.

About the Author

Murali Chemuturi is a Fellow of the Indian Institution of Industrial Engineering and a senior member of the Computer Society of India. He is a veteran in the software development industry and is presently leading Chemuturi Consultants, which provides consultancy in software process quality and training. Your feedback is greatly appreciated—please e-mail the author at murali@chemuturi.com.