

# Reengineering Customer Support

## Part 6: Designing a New Model



*Dave Brown is a management consultant, teacher, and writer. Dave teaches management training programs for Support Center University ([www.SupportCenterU.com](http://www.SupportCenterU.com)). He also consults with selected clients to establish world-class service operations and is considered an expert in the areas of process improvement, staffing models, and change management. You may reach Dave at his office in Boulder, Colorado, at 303-494-4932 or [dave.brown@SupportCenterU.com](mailto:dave.brown@SupportCenterU.com). For more information on the topics discussed in this column, contact Dave or visit [www.SupportCenterU.com](http://www.SupportCenterU.com).*

**D**esigning an efficient support center operation involves a little science and a little art, but there really isn't any magic. If you've followed the advice I've offered in my previous installments of this column series, then you are ready. We've discussed preparing to reengineer an organization, performing an in-depth operational assessment, setting up teams to carry out the project, and data collection and analysis. You will need to keep a few guidelines in mind when designing the new model.

When designing any process, it's essential to keep the design as simple and as direct as possible. Each step in the process should add value and move you closer to completion. This means that any step that could be avoided should be avoided. Any handoff or transfer that could be avoided should be avoided as well. A process designed in this manner will make the customer happy and will minimize your costs, too. So where do you begin? Generally, you should start by defining the goal, and then determine the most efficient means of accomplishing it. In customer support, the goal is to resolve the customer's issue. And the most efficient way to do this is usually to get the customer routed to the best available agent for that particular issue. (We'll revisit this later on.)

Before you design your new model, you should understand the pros

and cons of certain design approaches. Most important, you must understand how some approaches—even popular ones such as “sorting”—actually may have a negative impact on efficiency. Second, you need to look at best practices, because certain best practices can be applied to enhance your new model. Finally, before implementing your new model, you must “model” the process and predict the results.

### Things to Avoid

In general, you should avoid any design that is known to introduce inefficiencies. Also avoid any unnecessary step or steps that are likely to have a negative impact on customer satisfaction. Let's look at some examples of models that should be avoided:

- **Sorting.** In a “sorting” model, calls are routed to groups of specialists. These groups are dedicated to handling a particular type of call or problem. They don't overflow their calls to other groups or take overflow from others. But because it's nearly impossible to create a group that is exactly the right size to handle the fluctuating workload, service levels and utilization likely will be inconsistent. You may end up with calls queued up for one group while another group has idle (available) agents. This model always will result in either poor or inconsistent levels of service to the customer or poor utilization (in other words, excess cost to you). By all means, avoid this design.

## evolutions in the customer support center

---

- **Gatekeeper.** A “gatekeeper” model forces customers to go through some intermediary group before they speak with an individual who may solve their issue. And while a dispatcher for field service is a logical step in that process, the same concept is annoying to customers when applied to a telephone support operation. They view the gatekeeper as a hurdle. What’s more, it’s an added expense for you.
- **Callback.** We live in a world of instant gratification, and supporting customers in “callback” mode is no longer acceptable. Not only do customers expect someone to be available to address their issues when they call, but chances are that customers will be unavailable when an agent does call back. So begins phone tag, a frustrating process for both the agent and the customer and an expensive exercise for the contact center. If you think that you don’t have the staff to handle the calls “live” as they come in, think again. Handling calls live doesn’t require more staff, and it doesn’t increase the workload. If you can handle the work in callback mode, then you can handle it better in live mode. I’ve seen organizations reduce their workload by 50 percent by routing all calls live and eliminating all callbacks.
- **One size fits all.** Your customers have different needs and expectations. In addition, they have different levels of expertise and are likely to have different levels of urgency. If you design a model that treats each customer in the same manner and forces every customer through a system designed for the “typical” or “average” customer, you could irritate and alienate many of these constituents. Employing a process so rigid that it does not recognize and allow for unique requirements will be a problem.

This same lack of flexibility is one of the problems with the traditional “tiered” support model. In such a model, every customer starts at Level One, then gets escalated to Level Two, and so on. Yet, shouldn’t certain customers or certain types of problems be able to bypass Level One and go straight to an expert? What if certain problems are critical? What if certain customers are preferred and pay higher fees? A process that forces everyone through the same steps violates the core concepts mentioned at the outset—those of making sure that every step in the process is essential, adds value, and moves the customer closer to resolution.

### Best Practices

Certain models and processes have been proven, over time, to deliver good results. Several published industry studies have found that support centers providing world-class service have a number of common traits. Keep these best practices in mind as you design your new support model, and integrate them into your step-by-step process.

First, define the level of service that your contact-handling process is intended to deliver, and then design a model that will achieve this objective. While this may seem obvious, the reverse is often the case. Define the targets first, then question whether each step in the process is the most efficient way of moving toward the goal.

For example, if you define a goal as closing 75 percent of your incidents on the initial contact, this definition gives you a very solid framework for designing the process. In this example, it would be logical to do everything possible to ensure that the contact/incident is routed to the agent with the right skills. This, in turn, will lead you to design a process that identifies the

customer and issue so that you can route the contact appropriately. See? The goal helps to determine the process design.

But how would you know if a 75 percent first-call closure rate is a realistic target for your call center in the first place? Answer: From a best practice you have employed previously, i.e., data collection (the stage discussed in Part 5 of this column series). If, after analyzing the data you collected, you determine that once an incident reaches the agent with the right knowledge and skills, the resolve rate is greater than 75 percent, then you know that 75 percent is a reasonable target. The challenge then is to design a process that will get the customer to the right agent immediately.

Here’s another way to look at it. Don’t you eventually resolve 100 percent of your incidents (or close to it)? If you have the capability to resolve them, then why doesn’t resolution occur during the initial contact? Yes, there may be several factors involved, including researching for known issues or testing to duplicate the problem, but the most common reason why these problems don’t get resolved immediately is that the agent handling the initial contact is not the best match for the customer or issue. So the goal should be to design a process that will route each contact to the agent who is the best match.

This leads us to the next best practice, skills-based routing (SBR). Many people think of SBR as a technology that is expensive and complex to implement. However, SBR is actually a technique that essentially matches customer needs to agent skills. There are a variety of ways to implement SBR (some are fairly inexpensive), depending on the phone system involved. For

---

now, let's not worry about the *how* of such implementation (we will address implementation in the next installment of this series). Let's just worry about the *what*. What would you like to see happen? To help you visualize this, create a logical routing design—a flow chart that depicts how calls are identified and classified into certain categories and then routed to agents with certain skills. The opposite of the one-size-fits-all model is a process design that identifies the unique aspects of each call and handles it accordingly. SBR is one method of achieving this.

Equally important is a “triage” approach to handling calls. Just like the triage methodologies developed in field hospitals and still in use in hospital emergency rooms everywhere, a triage support methodology determines the urgency and complexity of each incident so that treatment decisions can be made accordingly. In the service and support world, the triage process must include defined escalation paths and thresholds that ensure customers will receive the appropriate level of attention.

On the surface, this might sound contradictory to my earlier statements about avoiding a gatekeeper model and avoiding unnecessary handoffs. However, the statement is actually completely consistent. If you use some form of SBR to get the caller to the right agent, then the agent is doing the right thing in first assessing whether he truly has the right skills to resolve the issue. If he determines that he is not the most suitable agent to handle the call, then he's doing the right thing in transferring the customer to a more appropriate agent right away. If the process is designed properly, very few transfers will be necessary. But the process should ensure that agents don't

waste their time and/or frustrate customers when someone else would be better suited to handle the issue.

There's another best practice that I refer to as taking a call center approach. I believe this is one of the most important factors when designing a model. What do I mean by “call center” approach? Think about your service/support operation as a “call/contact center,” as opposed to a “technical resource center.” What's the difference? A typical call or contact center manager believes that the key to running a smooth operation is efficient call-flow and call-handling processes. The typical technical support manager believes that support is all about solving technical problems. But I say to first approach the work from the vantage point of a call or contact center and design your processes to handle the workload efficiently. Second, ensure that the people in the contact center have the right set of technical skills to handle the calls. The best practice is always *process first*.

I've been focusing on the best way for the support center to handle “calls,” but be aware that my comments apply equally to handling e-mail or Web-initiated incidents. And when designing your support center process, one of the most important objectives is to do everything you can to get customers to resolve their own issues. While I won't delve into details here, I strongly advise that you integrate self-service options seamlessly into your processes. Customers should be given access to self-service tools, with options to transfer to a live agent or open a new case electronically (Web or e-mail) if the solution provided doesn't work out.

### **After the Design**

Now that you've designed your new

model, you need to test it before leaping into implementation. The thoroughness of your testing should be based on:

- The level of change to be implemented. (How dramatic is this change?)
- The complexity of the model. (Based on the complexity, how likely is it that an error in design occurred?)
- The level of investment. (How much is at risk if the model doesn't meet expectations?)

The more change, risk, and/or complexity involved, the more thoroughly you should test or simulate the model.

There are several ways to test the model. If an outside consultant is assisting with your reengineering, he might conduct the simulation for you as part of his services. But if you are not using a consultant (or even if you are) and you have the time and budget, a couple of tools on the market are designed specifically to simulate a call center environment. For most situations, however, I recommend using a combination of tools that you can assemble or develop yourself.

### **The Homegrown Simulation**

First, review your assumptions. Those assumptions that impact the design are primarily the workload metrics (e.g., call volumes, handle times, traffic patterns, resolve rates). Your model was designed based on these workload assumptions, so be sure that they are as accurate as possible. As I have stressed previously in this series, you always must validate the data. You should be able to reach the same conclusion using at least two different methods.

Next, use a staffing forecasting tool to predict the level of service that will be achieved based on the workload assumptions and the planned level of staffing. This can be a com-

## evolutions in the customer support center

---

plex and tedious process, but a workforce management system can simplify the analysis dramatically. If you are short on funds for new applications, you also can handle the analysis using a free demo program or online tools.

Again, a properly configured workforce management tool can run the entire scenario for you in seconds, calculating all of the different call queues, available skills, and hours of operation. But if you're doing this in a semimanual mode, you may need to run each scenario separately. You should run the model for each hour of planned operation, and you may need to run it several times for each hour (if you have multiple queues that

don't "overflow"). For instance, if you have five call queues/groups and your support center operates 12 hours per day, you'll need to run 60 scenarios in order to get a complete view of your predicted performance.

The outcome of this analysis will be a predicted level of service for each call queue for each hour of the day. You also will have the foundation of a staffing model: the number of agents to be scheduled, hour by hour, queue by queue.

Your next step is to build a pro-forma budget. Because you know the key staffing requirements for supporting the inbound workload, you can calculate the related staffing levels. Based on the data gathered in the earlier steps, first determine the workload and staffing required for follow-up or escalated calls. Then determine the staffing for non-call work, such as participation on product development teams. Next, factor in absenteeism, vacation time, and ongoing training. Also calculate the management overhead (one manager for every 10 employees, for example) and the administrative/clerical overhead. With a complete staffing plan, you should be able to build an expense budget.

These exercises should lead to a reasonably accurate view of your new support operation after implementation. Here are the key results:

- **Service level.** You will have predicted the service level (speed of answer) for each hour of operation. You actually can create a table, insert the expected call volume (hour by hour), and chart the hold times (predicted).
- **Resolve rate.** You will have predicted the percentage of calls closed on first contact. You can forecast the calls received and closed and then de-

termine the ongoing workload of follow-up calls.

- **Staffing levels and expenses.** You will have accurately determined the required headcount for the support department, and you will have created a pro-forma expense budget.

- **Customer satisfaction.** This is a little more difficult to predict, as there's more to customer satisfaction than just the quality of service (e.g., product, price). You will, however, be able to state with reasonable confidence that customers will be satisfied with the service they receive.

With this detailed information now in hand, you can perform a comparison between your current operating performance and future operating performance. This will allow you to gauge the potential benefits of implementing the planned model. And if this future state meets your expectations, you're ready to start implementing! ▼