Chapter

2

Service Center Organization

If you have the luxury of organizing or reorganizing your service center, there are a number of important factors to consider. Certainly among the most important of these is the type of service center you plan to run. It is important to consider whether you will provide the absolute best service available or the minimum service required to get by. You must determine the environment you are currently supporting and the environment you plan to support in the future. You must also consider the expected call volume and call complexity. Based on the volume and complexity of calls, you can use queuing theory to determine the number of resources and the technical skills you require to handle the anticipated volume and complexity. Another important consideration in organizing a service center is the types of support tools you will need. The tools you deploy to support the service center are likely to affect both the number of resources you require and the model you use to organize the resources.

Based on these factors, there are various approaches you can use to maximize the utilization of resources to meet your service levels and objectives. As noted in Chapter 1, "Introduction to Problem Management," these approaches are referred to as the *service delivery type*. The two service delivery types are *immediate* and *managed*. In the immediate model, customers go into a queue and wait for the next available agent, who immediately works with the customer to resolve the problem. In the managed model, the customer typically sends an email or fax, or fills out a Web-based form, which then goes into a queue. Unlike the immediate model, the managed model usually involves a controller (manual or automated) that reviews, prioritizes, and then assigns the ticket to an agent or an agent pool. Today, many service centers use both of these models.

There are, of course, various ways of implementing these models to maximize the utilization of your resources. You may implement one or more tiers of support to optimize the provision of service to customers and to optimize your utilization of resources.

2.1 Immediate Response Model

Implementing the immediate response model means that you will attempt to handle calls (or walk-ups) when they arrive. Customers will contact the service center using the contact methods you have provided for them. These methods can include the telephone, a line to stand in, email, fax, chat room, etc. Once customers have contacted the service center, you must have some mechanism for routing them to the next available agent or engineer or service representative. The routing mechanism can be as simple as one line for each agent, or it can be a sophisticated call tree that routes the customer to the most appropriate queue based on customer input; the customer then waits for the next available agent in that queue. There are other routing options in between. For example, if you do not have a call distribution system, one or more receptionists can take calls and then manually route (dispatch) them to the appropriate person or queue in the next tier.

In this model, the customer contacts the service center and is then routed to the next available agent. The pool, or pools, of agents that provide immediate response are in tier 1. Tier 1 can be one pool of agents or it can be multiple pools of agents. The simplest organizational model for a service center consists of one tier with one pool of resources. Queuing theory states that this would be the most efficient model for handling call volume while maximizing resource utilization. This model ignores complexity and variability, however, because it assumes that each resource in the pool can handle any issue that comes in. This may or may not be possible in the environment you support. In environments that support multiple products from multiple vendors on various platforms, it is not reasonable to expect that any one person can handle every type of call. Even if you could find a pool of gurus capable of providing that level of support, it would be more expensive and less efficient to use those expensive resources to handle common, recurring, simple problems.

Two examples of the single-tier, immediate response service center model are shown in Figure 2–1.

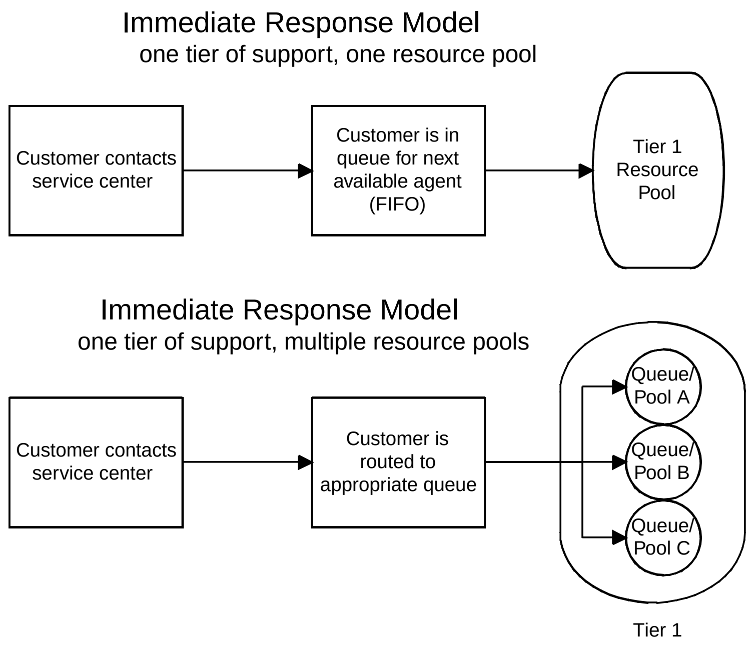
Immediate Response Model
one tier of support, one resource pool
Customer contacts service center
Customer is in queue for next available agent (FIFO)
Tier 1 Resource Pool
Immediate Response Model
one tier of support, multiple resource pools
Customer contacts service center
Customer is routed to appropriate queue
Queue/ Pool A
Queue/ Pool B
Queue/ Pool C
Tier 1

Figure 2–1 Single-tier, immediate response service center models.

In the second example, tier 1 support is subdivided into multiple pools. As mentioned above, queuing theory states that multiple pools are less efficient than a single pool. This makes logical sense, since resources in pool A and C may be underutilized, while all resources in pool B may be occupied and have many customers waiting in the queue. This does not mean that subdividing tier 1 resources into multiple pools is a bad idea. Suppose you provide support for multiple products from multiple vendors and you receive a significant number of calls from customers each day. It may be unreasonable to expect that each person can handle any call about any product, so you could divide your resources into pools that specialize in one or more products or one or more types of problems. When customers contact the service center, they are routed to the queue that is best suited to address their issue.

There are three factors you need to consider when designing the number of tiers you require and how to organize each of those tiers (one or more pools). The factors are call variability, call complexity, and call volume. Call variability refers to the different types of requests that come into the service center. The more types of requests received, the higher the variability. Request variability is important to understand because it indicates the amount of knowledge required to handle the requests. In a multiproduct, multivendor, multiplatform environment, the variability is high. If you support a highly standardized environment with few products, the variability is low. The higher the variability, the more knowledge required to provide support. The lower the variability, the less total support knowledge required.

2.1.1 Request Variability

Based on the variability of your workload, you can decide how to organize the support tier. In Figure 2–1, tier 1 is subdivided into three resource pools. This indicates that variability of requests was high enough that one resource could not be expected to efficiently handle all types of requests coming into the service center. For example, pool A may handle all requests that deal with desktop productivity software, such as Microsoft Office and Microsoft Windows. Pool B may handle all requests that deal with internal corporate applications, such as the general ledger and human resource systems. Pool C may handle all value-added services, such as ordering PCs, moves, and training requests. Keep in mind that by subdividing resource pools, you are potentially reducing efficiency in handling the call volume, because pool A may be overwhelmed with calls, while pool B and pool C have excess capacity.

One approach to deal with this inefficiency is to allow calls to overflow from one pool to another, as shown in Figure 2–2.

For this overflow approach to work, the resources in pool B must have at least some capability of dealing with requests that are typically handled by pool A. That can occur as a result of cross-training as well as having access to a shared knowledge base.

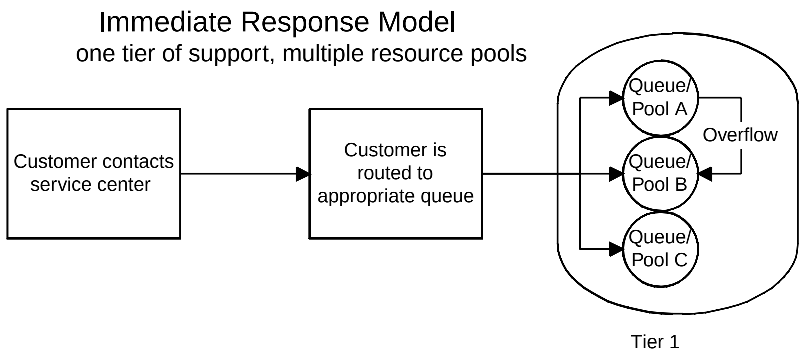
Immediate Response Model
one tier of support, multiple resource pools
Customer contacts service center
Customer is routed to appropriate queue
Queue/ Pool A
Queue/ Pool B
Queue/ Pool C
Overflow
Tier 1

Figure 2–2 The overflow approach to handling call volume.

Several points to keep in mind are

- The fewer pools within a tier, the better in terms of your ability to handle call volume efficiently.
- The fewer the pools within a tier, the broader the knowledge required by the staff.
- Cross-training pools within a tier allows for more effective call overflow.

2.1.2 Request Complexity

So far, we have considered the variability of the requests but we haven't considered the request volume or complexity. The volume and complexity of requests play an equally important role in determining how to structure the service center organization. Ideally, a service center (and its customers) would like to handle all requests immediately. However, when you consider the volume and complexity of requests, this may not be possible.

Complexity refers to the level of difficulty encountered while servicing the request. Many requests are recurrent and are thus well defined and well known by the staff. These requests are not difficult to service and therefore have a low complexity. Other requests occur infrequently and require more research. These may involve interaction between multiple products, rarely used functions and features, and so on. Complex issues usually take more time to resolve and may require more technical expertise than is available in the tier 1 resource pool.

The complexity of requests should be used to determine the number of tiers of support you require. Thus far we have only considered one tier of support, which may or may not be appropriate, depending on the complexity of the requests you receive. If a significant portion, say 90 percent, of the requests you receive are complex, then one tier of support would be appropriate. The tier would be staffed with highly skilled specialists who could handle the complex requests. They would be organized into multiple pools within the tier, based on their expertise. The staffing levels would have to take into consideration the volume of requests and the additional time required to handle complex requests. The same staff could also handle the 10 percent of calls that are less complex.

If on the other hand, 90 percent of the calls are routine, noncomplex calls, an additional tier of support should be considered. The additional tier, tier 2 in this case, would handle the 10 percent of calls that are too complex or too time consuming to be handled by the tier 1 resources. The diagram in Figure 2–3 illustrates a two-tier response model.

As illustrated in the diagram, there are two tiers of support. In this example, 90 percent of the calls are not complex and are handled by generalists in tier 1. However, when a tier 1 agent encounters a request that he or she cannot resolve, the request can be routed, or escalated, to the appropriate tier 2 group. In the example, pool A is responsible for desktop productivity software and the desktop operating system, and routes calls when necessary to pool 1. Let's assume there are 10 agents in pool A and they handle 92 percent of the requests that are routed to them. Generally, the tier 2 pools have fewer agents than tier 1 because they are handling far fewer requests—only 8 percent of the total in this case. Keep in mind that even though they are handling fewer requests, the requests they are handling are more complex and will generally take longer to resolve. An evaluation of your resolution times by tier will bear this out.

As with tier 1, the variability of expertise required will help you determine how many pools of expertise are needed in tier 2. In Figure 2–3, pool B in tier 1 can escalate calls to three different pools in tier 2. This is because of the variability of the expertise required to handle complex requests across the various corporate applications that pool B supports. Pool 2 in tier 2 may handle all complex requests for corporate financial systems, while pool 3 handles all complex requests for the human resource systems.

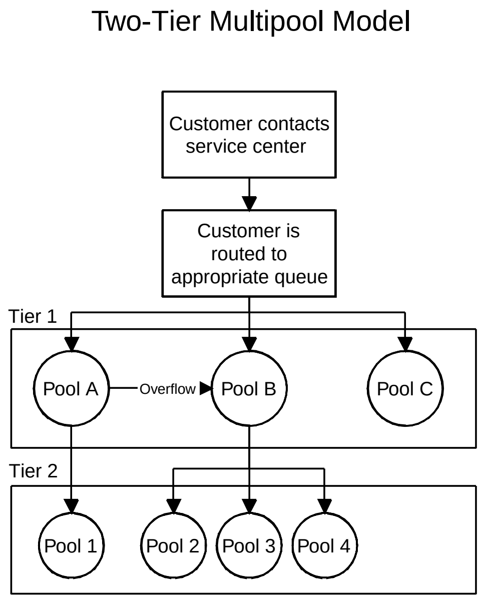

Figure 2–3 Two-tier response model for high-volume, routine calls.

Notice that pool C in tier 1 does not require a second tier of support, because they are capable of handling all requests in their area of responsibility.

The multitiered model is a good structure for efficiently utilizing technical expertise when the bulk of the requests you handle are not complex. In most tiered service centers, each successive tier has more specialized technical expertise than the preceding tier. Generally speaking, the more specialized the expertise, the more expensive the resource. This means that the generalists in tier one are on average less expensive than the tier 2 specialists. The multitier model is efficient because the less expensive resources are handling more requests, thus reducing your costs for servicing requests. You would not want a $70,000-a-year Microsoft Certified Systems Engineer (MCSE) answering questions about formatting Word documents when a $25,000-a-year generalist could provide the same service.

The multitiered model is also useful for handling overflow from tier 1. If pool A is overwhelmed with requests, their overflow can be routed to pool 1 in tier 2. This should be the exception rather than the rule, because pool 1 has their own work to do and as mentioned above, it may not be the most efficient use of tier 2 expertise. If you use this overflow approach, keep track of how often it occurs. If it occurs frequently, you should add additional resources to the tier 1 pool. It should be fairly easy to justify the costs of the additional resources if your tier 2 resources are indeed more expensive than tier 1.

There are several disadvantages to using a multitiered model. A multitiered organization is more difficult to manage and requires additional layers of management and overhead expenses. While tier 1 may have shift managers in charge of the entire tier 1 staff during the shift, each tier 2 and tier 3 pool may also have a manager, particularly if the resources in the pool are not dedicated to the service center. Processes must be put in place to handle the movement of requests between pools and between tiers. Those processes must be developed, trained, implemented, and maintained. Customer requests take longer to resolve as the request moves from tier to tier, because they are generally moved into a new waiting queue. Customers usually have to explain their request again and answer additional questions every time their request is moved to a new tier. Finally, the multitiered model can foster morale problems, because it often creates a class system, which closely matches tier level. Tier 1 resources can be viewed as lower class service center employees because they have less expertise than tier 2 resources and are generally paid less.

2.1.3 Request Volume

The same principle applies to complexity. When you use request complexity as a gauge for determining the need for a second tier of support, you must consider the volume of complex requests. If nearly all of your requests are simple and can be handled by tier 1 resources, then it probably does not make sense to add a second tier of support. As mentioned above, adding a second tier of support adds complexity and additional overhead to the service center. It also adds delays into the resolution time. In this scenario, it may be better to add several more highly trained resources to the tier 1 pool. These resources would take requests just like everyone else, but would also be available if less skilled agents needed to redirect requests that were beyond their capabilities. A second approach in this scenario is to have the more skilled agents available as mentors to everyone else in the resource pool. As mentors, they only take requests after another agent has made an attempt at resolving the problem or during overflow situations.

2.1.4 Other Reasons to Add More Tiers

So far we have evaluated request volume, variability, and complexity and their roles in implementing additional tiers of support. There are other valid reasons to consider adding additional tiers of support. One of the most common reasons is the need to dispatch technicians to the desktop. If your service center has to provide hardware support, you certainly can't do that over the phone, and you certainly do not want tier 1 agents abandoning their stations to go to the customer's desk to swap out a pair of speakers or a mouse. Typically, groups that are going to be dispatched are established as a tier 2 resource pool. You can justify the additional overhead and complexities of having a second tier even though the requests may not be variable or complex. The justification is based on the need to provide dispatch support and the need to have tier 1 agents available to manage request volume. Often, hardware support is outsourced and structured as a tier 2 resource pool being tasked by tier 1. When the support is outsourced, you need to manage the outsourced function as a separate pool so that you can closely measure the usage and performance of the resources against any service level agreements (SLAs) you have in place.

If you do not have tools that allow your agents to take remote control of customer workstations, you will inevitably have to dispatch resources to the desktop to resolve problems. As with hardware support, any dispatched group is a good candidate for a tier 2 resource pool.

Beyond dispatch, there are still other valid reasons to implement additional tiers of support. A classic reason is security and control. Suppose your customers request changes to global mailing lists or group access rights. It would not be appropriate to provide all tier 1 agents with administrator rights to make those types of changes. Where requests require system administration (SA) rights, you should consider a multi-tiered model in which the requests are routed to the appropriate pool of system administrators.

Thus far we have discussed only two tiers of support. Very often, service centers have three or more. Use the same decision-making criteria you used to add the second tier when you consider additional tiers of support. Consider volume, variability (specialization required), complexity, outsourcing, security, and control. Typically, tier 3 resources include pools such as internal developers for products developed and/or maintained in-house and external help desks for the off-the-shelf products you support. They also include groups such as your metro and wide area network carriers, ISPs, and any other groups from which your company buys products and services. Tier 3 resources are usually not full-time service center employees; they generally have other responsibilities within the company, perhaps as internal developers or architects, or they might even work for outside companies, such as software vendors.

2.2 Managed Response Model

The managed response model is very similar, at least from an organizational perspective, to the immediate response model. There are three primary differences:

1. The customer uses alternative methods to contact the service center.
2. You have a chance to manage the assignment and completion of the requests to efficiently balance the workload.
3. You have the chance to prioritize the requests and handle them in an order other than first in/first out (FIFO), as in the immediate response model.

For example, if a request arrives via email during your peak load hour from 9:00 A.M. to 10:00 A.M., you have the opportunity to delay the response to that request until, say, 10:30, when your peak load is over and more resources are available. A managed response model is shown in Figure 2–4.

Notice that the organization of the resources that handle the request in the managed response model is no different than the structure of the immediate response model. The same people can handle these requests. Just as in the immediate model, how the calls are routed to the appropriate queue depends on your tool set. Many help desk systems now have the ability to automatically log, prioritize, and route the request to a queue. If you do not have these tools, these tasks must be done manually.

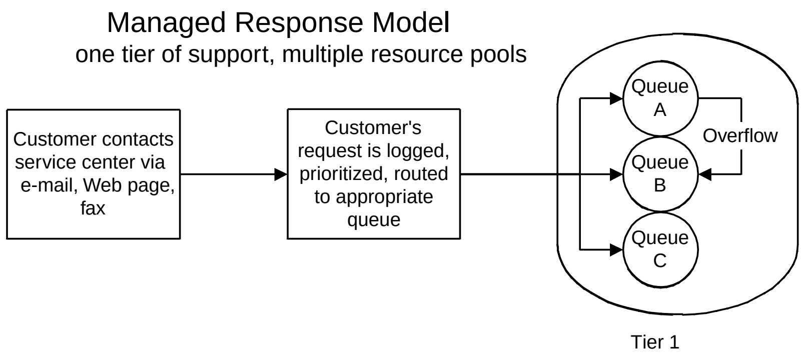
Managed Response Model
one tier of support, multiple resource pools
Customer contacts service center via e-mail, Web page, fax
Customer's request is logged, prioritized, routed to appropriate queue
Queue A
Queue B
Queue C
Overflow
Tier 1

Figure 2–4 A managed response model with one tier of support.

The primary advantage to the managed response model is that it allows you to distribute workload and smooth out the peaks and dips in request volume. It also allows you to route requests to the most appropriate person or pool. This approach does have disadvantages, though. By definition, the managed response model delays the servicing of the request, which may not be acceptable to the customer. A more significant disadvantage is that it has the potential to actually generate more calls to the service center than if the request had been initiated over the phone in the first place. This occurs because the agent who handles the request will often have to contact the customer to gather additional information. If the customer is away from his or her desk, the agent must leave a message and either wait for a return call or make a follow-up call. It may take several calls before the agent and the customer actually connect, and return calls often come at inopportune times, when the agent has turned his or her attention to other tasks. It is also more difficult to gather resolution time metrics about the request, because work may be done when the agent is not on the phone with the customer, so typical call metrics are not available.