Chapter 1

# An Introduction to Designing VMware Environments

COPYRIGHTED MATERIAL

Designing VMware vSphere environments can be a complex topic, one that means many different things to many different people. In this chapter, we'll provide an introduction to designing VMware vSphere implementations. This introduction will give a preview of some of the more detailed discussions that take place in later chapters and will provide a framework for how the other chapters fit into the overall process.

This chapter will cover the following topics:

- The importance of functional requirements in VMware vSphere design
- The what, who, and how questions involved in VMware vSphere design and why they're important
- An overview of the VMware vSphere design process

## What Is Design?

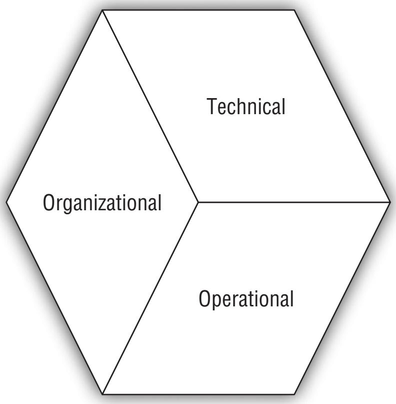
When we talk about "designing your VMware vSphere environment," what exactly does that mean? In the context of VMware vSphere, what is design? What does design entail? These are excellent questions — questions that we intend to answer in this chapter and the coming chapters throughout this book.

In our definition, *design* is the process of determining the way in which the different elements that make up a VMware vSphere environment should be assembled and configured to create a virtual infrastructure that is strong yet flexible. Design also includes the process of determining how this virtual infrastructure will integrate with existing infrastructure as well as how the virtual infrastructure will be operated after the implementation is complete.

That's a reasonable definition; but for someone who is new to VMware vSphere design, does this really describe what design is? Does it help you understand the nature of design, or what makes up a design?

In looking at a VMware vSphere design, we can say that it has three key facets: the technical or structural facet, the organizational facet, and the operational facet. Figure 1.1 shows how these three facets are all part of the larger entity that we refer to as *design*.

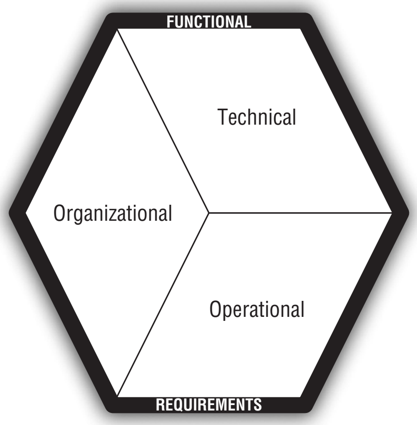
**FIGURE 1.1**
The different parts of VMware vSphere design are merely facets of a larger entity.



These three facets serve to organize the design in a way that is logical to us, grouping together information, decisions, criteria, constraints, and standards. We'll explore these facets in more detail later in this chapter in the section titled "The Facets of vSphere Design."

When defined or described this way, VMware vSphere design seems simple. But as you'll see in this book — or perhaps as you've already seen, depending on your experience — it can be complex. Even in the most complex of designs, however, a single unifying element brings the different facets together. What is this single unifying element, as illustrated in Figure 1.2? It's the functional requirements of the design.

**FIGURE 1.2**
The functional requirements unify the different facets of the design.



Functional requirements are incredibly important. In fact, we can't stress enough the key role that functional requirements play in VMware vSphere design (or any IT design task, for that matter). Functional requirements are important because they answer the question "What *things* should this design *do*?"

It's important to remember that companies implement VMware vSphere for a reason, not just for the sake of having vSphere installed. As much as VMware would love for that to be the case, it's not. In every instance, there's a driving factor, a force, a purpose behind the implementation. There's a reason the company or organization is implementing VMware vSphere. That reason, naturally, varies from customer to customer and organization to organization.

Here are some example reasons taken from our own experience in the virtualization industry:

**Consolidation** The company or organization has too many physical servers and needs to reduce that number. The need to reduce the number of physical servers can be driven by any number of reasons, including a need to reduce data-center space usage, a need to cut power and cooling costs, or an attempt to reduce hardware refresh costs.

**New Application Rollout** The company or organization is deploying a new application or a new service in its data center, and it has chosen to use virtualization as the vehicle to accomplish that deployment. This may be a deployment of a new version of an application; for example, a company currently using Exchange 2007 may decide to roll out Exchange 2010 in a virtualized environment on VMware vSphere. As another example, a company deploying SAP may choose to do so on VMware vSphere. The reasons for choosing to deploy on a virtualized environment are too numerous to list here, but they can include increased utilization, simplified deployment, and better support for a disaster recovery/business continuity (DR/BC) solution.

**Disaster Recovery/Business Continuity (DR/BC)** The company or organization is in the midst of developing or enhancing its DR/BC solution and has chosen to use virtualization as a key component of that solution. Perhaps the company is using array-based replication and wishes to use VMware vSphere and VMware Site Recovery Manager (SRM) to provide a more automated DR/BC solution. The choice to use virtualization as a component of a DR/BC solution is almost always a financial one; the company or organization wishes to reduce the amount of downtime (thus minimizing losses due to downtime) or reduce the cost of implementing the solution.

**Virtual Desktop Infrastructure** The company or organization wishes to deploy a virtual desktop infrastructure (VDI) in order to gain desktop mobility, a better remote-access solution, increased security, or reduced desktop-management costs. Whatever the motivation, the reason for the VMware vSphere environment is to support that VDI deployment.

As you can see, the reasons for adopting virtualization are as varied as the companies and organizations. There is no one reason a company will adopt virtualization, but there will be a reason. There are often multiple reasons. These reasons become the basis for the functional requirements of the design. The reasons are the *things* the design must *do*. Functional requirements formalize the reasons why the company or organization is adopting VMware vSphere and turn them into actionable items that you'll use to drive all the other decisions in the design.

Think about some of the examples we just provided. Does the organization plan to virtualize a new rollout of Microsoft Exchange Server 2010? If so, then the VMware vSphere design had better accommodate that functional requirement. The design must specifically accommodate Microsoft Exchange Server 2010 and its configuration needs, supportability requirements, and resource constraints. If you fail to properly account for the fact that Microsoft Exchange Server 2010 will run in this virtualized environment, then you've failed to consider one of the design's functional requirements — and, in all likelihood, the implementation will be a failure. The design will fail to *do* the *thing* the company needs it to do: run Microsoft Exchange Server 2010.

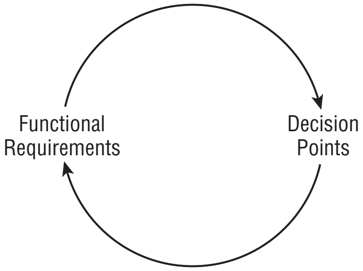
With this in mind, you can look back at Figure 1.2 and better understand how the functional requirements both surround and unify the facets of VMware vSphere design. Continuing in our example of an organization that is deploying Exchange Server 2010 in a virtualized environment, the functional requirements that derive from that reason affect a number of different areas:

- The server hardware selected needs to be capable of running the virtual machines configured with enough resources to run Microsoft Exchange Server 2010.
- The virtual machines that run Exchange will, most likely, need to be configured with more RAM, more virtual CPUs (vCPUs), and more available disk space.
- The configuration of Exchange Server 2010 will affect cluster configurations like the use of vSphere High Availability (HA), vSphere Distributed Resource Scheduler (DRS), and vSphere Fault Tolerance (FT).
- The cluster configuration, such as the ability (or inability) to use vSphere FT, will in turn affect the networking configuration of the VMware ESXi hosts in the environment.
- Operational procedures need to be included in the design as a result of the use (or lack of use) of features like vSphere HA, vSphere DRS, and vSphere FT.

The list can go on and on, but at this point you should get the idea. The functional requirements affect almost every decision point in every facet of the design; as a result, they lie at the core of creating a VMware vSphere design. Any design that doesn't directly address the organization's functional requirements is a poor design, and the implementation won't be a success. Any consultant or VMware vSphere architect who attempts to design a vSphere environment without knowledge of the functional requirements will fail. After all, the functional requirements are the targets the design is aiming to hit; how can the design hit those targets if the targets aren't known and understood?

Interestingly, although the functional requirements directly affect the decision points — things like what servers to use, the form factor of the servers, the number and type of network interface cards (NICs), and so on — these decision points also affect the functional requirements. An inherent interdependency exists between the functional requirements and the decisions, as shown in Figure 1.3.

**FIGURE 1.3**
Functional requirements and design decision points are interdependent.



**NOTE** Although we've been focusing primarily on requirements in this discussion, formal VMware design documentation typically refers to four primary factors that drive your design: requirements, risks, constraints, and assumptions. We've already discussed *requirements*; these represent the specific features or functions the design must provide or satisfy. *Constraints* are decision points — such as the type of server you'll use, the type of storage you'll use, or the way in which you'll connect to an existing network — that have already been made and can't be changed. *Risks* represent specific areas where the design may not satisfy the requirements or the constraints. For example, if the design has a certain storage-capacity requirement but also has a constraint that prevents the design from meeting that capacity requirement, this is a risk. Finally, *assumptions* are requirements or constraints inserted into the design by the vSphere architect in order to fill in missing information. For example, in order to plan for future growth, certain growth requirements need to be known. If these requirements aren't known, the architect can use an assumption to fill in the blanks when creating the design.

Keep these four factors in mind as you continue to review the process of vSphere design.

As a result of this interdependency, you'll find that creating a design is often an iterative process. Based on the functional requirements, you make a decision. Then, based on that decision, you ensure that the decision is capable of supporting the functional requirements. If so, you proceed with other decision points. If not, you revise the decision point based on the functional requirements. This iterative process again underscores the importance of the functional requirements in the creation of the design.

At the beginning of this section, we told you that design is the process of determining the way in which the different elements that make up a VMware vSphere environment should be assembled and configured to create a virtual infrastructure that is strong yet flexible. When we factor in the key role that functional requirements play in unifying the technical, organizational, and operational facets of a design, perhaps a better definition is that design is the process of determining the way in which the different elements that make up a VMware vSphere environment should be assembled and configured in order to satisfy the functional requirements. Or, in simpler terms, design is making the VMware vSphere environment *do* the *things* it needs to do.

Now that you have a better understanding of what VMware vSphere design is and why it's important, in the next section we'll take a closer look at the facets of design.

## The Facets of vSphere Design

As we described in the previous section and illustrated in Figure 1.1, your design must address three facets, or the design is incomplete. These three facets — technical, organizational, and operational — are unified by the functional requirements; but within each facet, a wide variety of decision points must be specified in the design.

The best way to understand how these facets differ from each other is to look at the types of decisions that fall in each facet. This is graphically depicted in Figure 1.4.

In each facet of the design, you'll make decisions based on the functional requirements, followed by an iterative review (as illustrated in Figure 1.3) to ensure that the functional requirements are still met based on the decision. In this section of this chapter, we'll take a deeper and more detailed look at these facets, examining some of the decision points that are involved.

We'll start with the technical facet.

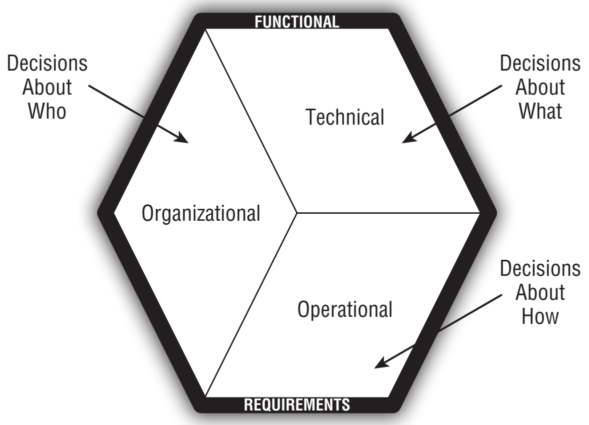


**FIGURE 1.4** Each facet of the design primarily addresses a different type of decision, such as who, what, or how.

## The Technical Facet

The *technical facet* is the facet that IT people most closely identify with design. It involves the pieces and parts of technology that make up the final environment: things like what servers to use, what quantity of random access memory (RAM) the servers will have, what configuration the storage array will use for its datastores, and what the networking configuration will look like. You may also see the technical facet referred to as the *physical design*, although it incorporates certain logical aspects as well. These are all decisions about what will or won't be included in the design, and these decisions fall into the technical facet, as illustrated in Figure 1.5.

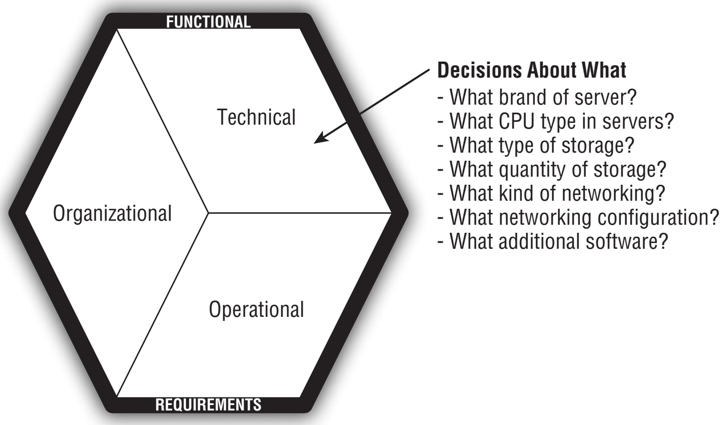

FUNCTIONAL
Technical
Organizational
Operational
REQUIREMENTS
Decisions About What
- What brand of server?
- What CPU type in servers?
- What type of storage?
- What quantity of storage?
- What kind of networking?
- What networking configuration?
- What additional software?

**FIGURE 1.5** The technical facet includes the "what" decisions that are familiar to many IT professionals.

It's important to be sure the technical facet is as complete as possible, so the design should include — at the very least — decisions in the following technical areas:

- The number and type of servers in the environment
- The number, type, and speed of the CPUs in the servers
- The amount of RAM in the servers
- The type of connectivity to the shared storage
- The type or configuration of the shared storage
- The number of physical NIC ports available
- The manufacturer and model of the NICs in the servers
- The exact configuration of the virtual switches (vSwitches) and distributed vSwitches in the environment
- The amount of power required by the equipment
- The amount of cooling required by the equipment
- The amount of rack space or floor space required by the equipment

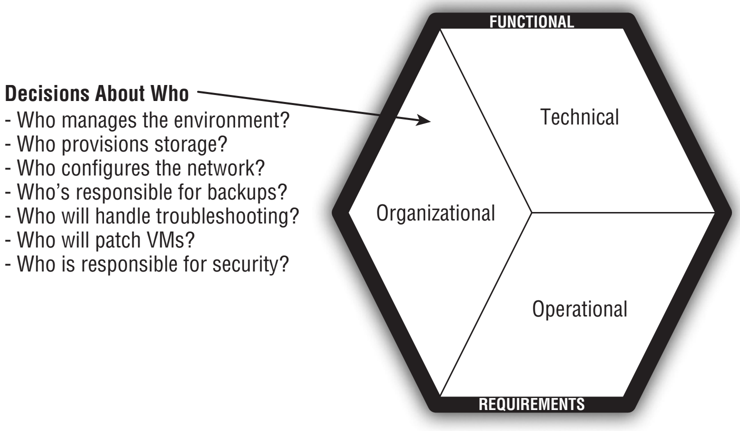
This is, of course, just a small list to get you started thinking about the detail you should provide when crafting a design for a VMware vSphere environment. Subsequent chapters examine each of these areas in much more detail. For example, VMware vSphere networking is covered in detail in Chapter 5, "Designing Your Network"; Chapter 6, "Storage," discusses shared storage in more depth.

A complete and thorough design addresses more than just the technical facet, though. Your design should also address the organizational facet.

## The Organizational Facet

Although the technical facet is important, equally as important is the *organizational facet*. It's concerned with questions centered on "who," as you can see in Figure 1.6.

**FIGURE 1.6**
The "who"-focused decisions of a VMware vSphere design fall into the organizational facet.



Decisions About Who
- Who manages the environment?
- Who provisions storage?
- Who configures the network?
- Who's responsible for backups?
- Who will handle troubleshooting?
- Who will patch VMs?
- Who is responsible for security?

You might initially think that these "who"-focused questions aren't important or aren't your responsibility. Aren't these the sort of decisions that should be made by the customer? In a certain respect, yes — these decisions are driven by the functional requirements every bit as much as the "what" questions in the technical facet. As you'll see later in this chapter, in the section "The Process of Design," gathering the functional requirements from the customer or organization (if it's your own organization) means gathering information about who is responsible for the various tasks within a virtualized infrastructure.

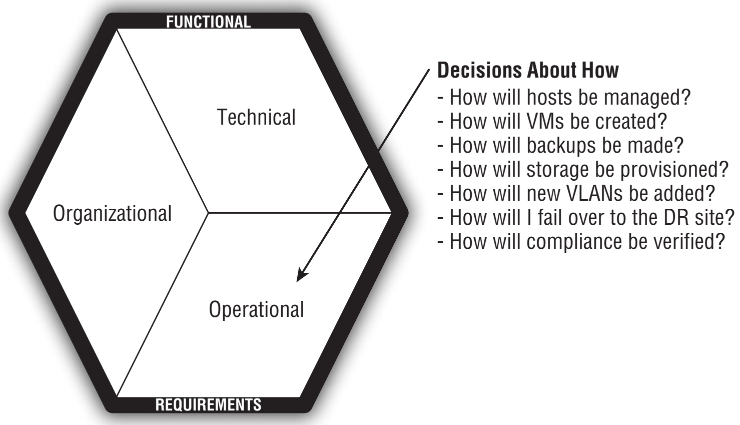
The other thing to consider regarding these "who"-focused questions, though, is the fact that the customer or company may not know or understand who will be responsible for certain aspects of the design. For organizations that are new to virtualization, the convergence of server administrators, network administrators, storage administrators, and security administrators often means they're confused and don't understand who can or should be responsible for the different areas of the new VMware vSphere environment. By embedding the answers to these questions in your design, you can help the customer (or your own organization) better understand how these responsibilities should be divided and who should be responsible for each area.

The final facet of VMware vSphere design addresses an equally important but often overlooked type of question: how should the environment be operated? This is the operational facet, and we discuss it in the next section.

## The Operational Facet

The *operational facet* of a VMware vSphere design answers questions focused on "how," such as those illustrated in Figure 1.7.

**FIGURE 1.7** Decisions about how you'll operate a VMware vSphere environment fall into the operational facet.


FUNCTIONAL
Technical
Organizational
Operational
REQUIREMENTS
Decisions About How
- How will hosts be managed?
- How will VMs be created?
- How will backups be made?
- How will storage be provisioned?
- How will new VLANs be added?
- How will I fail over to the DR site?
- How will compliance be verified?

As with the organizational facet, you might ask, "Why would I need to define operational procedures in a VMware vSphere design? Shouldn't these sorts of operations be tasks the organization already knows how to do?"

In the event of an organization that is new to virtualization, the answer to that question is no. Virtualization changes the way the data center operates — and the customer or company implementing virtualization for the first time must have these operational decisions spelled out in the design.

Even for organizations that are familiar with virtualization, the operational facet is still a critical part of a complete and comprehensive VMware vSphere design. It's possible, even likely, that the "what" decisions made in the technical facet of this design are different than the "what" decisions made in the technical facet of earlier designs. Server models change. Storage vendors introduce new products with new functionality — and new operational requirements. Networking vendors change the way their products work over time. All of these factors add up to a need to include operational information in every design to ensure that the organization implementing the design has the information it needs to operate the environment.

As an example, consider an organization that adopted virtualization a couple of years ago. It deployed VMware Infrastructure 3 (VI3) on HP ProLiant rack-mounted servers attached to a NetApp storage array via Fibre Channel (FC). Now, the company is implementing VMware vSphere 5.1 on Cisco Unified Computing Systems (UCS) attached to a Dell Compellent storage array via FC over Ethernet (FCoE). Do you think the operational procedures from the last implementation will be the same as the operational procedures for this new implementation? No, of course not. Just as technology changes over time, operations change over time as well. This is why the operational aspect is important not only for new VMware vSphere users but also for existing users.

Grouping design decisions into these three facets — technical, organizational, and operational — helps us, as architects, ensure that our design is complete and that it addresses all the aspects of creating a vSphere environment that satisfies the functional requirements. What these facets don't do, though, is help us to determine the specific way in which to apply the functional requirements. What are the guiding qualities or principles that help us make the decisions within each of these facets? That's the topic of the next section.

## The Principles of Design

In this section, we'll discuss the principles of design — the guiding ideas or thoughts that shape and influence the decisions we make within each facet (technical, organizational, and operational) in order to satisfy the functional requirements.

At a very high level, there are five basic principles of design:

- Availability
- Manageability
- Performance
- Recoverability
- Security

We'll use the acronym AMPRS to refer to these design principles. These principles are, for the most part, pretty self-explanatory, but we'll look at each of them in a bit more detail next.

### Availability

As you make decisions that will create a vSphere design, one principle to consider is availability. Availability encompasses a number of areas, including uptime and downtime, reliability, redundancy, and resiliency. Note that some aspects of performance are also on occasion associated with availability; for example, in the context of a service-level agreement (SLA), availability might also encompass application response times or application latency. We'll have a separate discussion on performance later in this chapter, but be aware of the close relationship between these two principles. With regard to the facets of design, the principle of availability typically has the greatest effect on decisions in the technical facet.

In some cases, the functional requirements explicitly call out availability; for example, a functional requirement may state that the vSphere design must provide 99% availability. In this case, availability is explicitly noted by the functional requirements and therefore must be incorporated into the design. In other cases, the functional requirements may not explicitly state availability demands. In these situations, the architect has to include an appropriate level of availability in the design as the various design decisions are made. The functional requirements may only state that 10 gigabit (Gb) Ethernet is required, in which case a single 10 Gb Ethernet switch will satisfy the functional requirements. However, a single 10 Gb Ethernet switch presents a single point of failure (SPoF). Is that an appropriate level of availability for this design, when availability has not been explicitly called out?

In situations like this, an architect uses an *assumption*. An assumption provides justification for the vSphere designer's decisions within the broader framework of functional requirements and design constraints (see the sidebar in the "What Is Design?" section). When availability isn't explicitly stated, the architect can provide an assumption that the environment will be made as available as possible within the financial limitations of the project.

## Manageability

Architects must also consider manageability. The principle of manageability most directly affects decisions within the operational facet because this principle involves the ongoing management or maintenance of the environment. Manageability encompasses a number of related ideas:

- Compatibility (can it be managed as part of the design?)
- Usability (how easy is it to manage?)
- Interoperability (does it integrate with other management structures in the environment?)
- Scalability (how well will it work as the environment grows?)

## Performance

Performance is often called out in the functional requirements, and it can affect decisions in the technical and operational facets. For the technical facet, it can affect all manner of design decisions, from the type of servers to the kind of network switches to the storage solution that is selected. With regard to the operational facet, it's most frequently seen as an SLA that defines performance metrics, such as response times, transactions per second, and maximum number of users supported. (As we mentioned earlier, keep in mind that in some cases certain performance metrics are also associated with availability.)

Naturally, even if no performance requirement is explicitly stated, architects designing a vSphere environment should consider performance when making design decisions.

## Recoverability

The principle (or quality) of recoverability includes such concepts as mean time to recover (or repair, abbreviated MTTR), maintainability, and DR/BC. Naturally, this makes recoverability related to availability; but whereas availability is more focused on preventing an outage, recoverability targets how quickly (and with how much effort) the environment can be restored/recover from a failure.

Organizations use a couple different metrics to measure or gauge recoverability. These metrics are recovery point objective (RPO; a measure of how much data loss can be sustained in the event of a major failure or disaster) and recovery time objective (RTO; how quickly an environment can be restored to working operation). Typically, your functional requirements will include RPO/RTO metrics that must be met by the design.

### Security

Every facet of the design — technical, organizational, and operational — is affected by security, and every design decision should consider security.

These five principles of design guide and shape all the design decisions. There are multiple ways to fulfill the functional requirements, but as a vSphere architect you must evaluate each option against these five principles. Does the option positively or negatively impact the availability of the design? What about the design's manageability? Or performance? What about recoverability and security? These principles provide guidance and direction on the best way to satisfy the functional requirements for a particular design.

Before we wrap up this chapter and start a more detailed look at VMware ESXi in Chapter 2, "The ESXi Hypervisor," we want to discuss one more area: the process of VMware vSphere design. It's the focus of the next section.

## The Process of Design

Now that we've discussed the *facets* of design and the *principles* of design, it's time to discuss the *process* of design. In this section, we'll cover how you go about creating a VMware vSphere design, some of the tasks involved, and some of the tools you can use to complete those tasks.

We'll start with what is, as we've said before, one of the most important areas: functional requirements.

### Gathering and Defining Functional Requirements

Functional requirements form the basis, the driver, for almost everything in the design. Most other decisions in the design are based on or affected by the functional requirements, so it's incredibly important to be as thorough and complete as possible during the process of gathering and defining them.

In many situations, some of the functional requirements are provided to you. For example, if an organization is adopting VMware vSphere in order to support a consolidation initiative, the business might clearly specify a functional requirement in a statement like this: "The virtualization environment must support the consolidation of 250 physical server workloads." No additional effort is required on your part to define this requirement. (But additional effort is clearly required to implement that functional requirement in the design.)

It's uncommon, in our experience, to have situations where all the functional requirements are provided. In these cases, you'll have to gather information to define the functional requirements. There are two primary ways to gather the information necessary to define the design's functional requirements:

- Reviewing documentation
- Performing interviews

**NOTE** You may see VMware define this type of approach to design — performing interviews to gather information — as the *consultative approach*.

### Reviewing Documentation

In some cases, the customer or organization implementing VMware vSphere has documentation that outlines the functional requirements. Remembering that virtualization is implemented in order to accomplish a goal (to "do something"), documentation is often created that outlines what the organization is attempting to achieve. For example, perhaps the organization is implementing virtualization as part of a desktop virtualization initiative. In that case, some of the functional requirements of the VMware vSphere environment can be derived directly from the documentation prepared for the desktop virtualization project. If the desktop virtualization documentation specifies that the VMware vSphere environment will automatically restart desktop VMs in the event of a host failure, that should immediately sound a mental alarm — your vSphere environment will need to use vSphere HA in order to meet that functional requirement. And because you'll use vSphere HA, you'll also need to use clusters, which means you'll require redundant management connections, which affects the networking design … and so on.

In another example, suppose the organization is migrating into a new data center and has compiled a list of all the applications to be migrated. You can use that documentation to understand the applications' needs and determine the functional requirements necessary to support those needs. Perhaps the application documentation indicates that the I/O profile is primarily writes instead of reads and that the application needs to sustain a specific number of transactions per second (TPS). That information translates into storage requirements that dictate the RAID level, array type, storage protocol, and capacity in I/O operations per second (IOPS).

Although reviewing documentation can be helpful, it's unlikely that you'll find all the information you need in a company's documentation. If the organization is like a lot of others, documentation is sparse and incomplete. In these instances, you'll need to gather information by going straight to the source: the people in the organization.

### Performing Interviews

Interviewing individuals in the organization or company that is implementing VMware vSphere is the second major way to gather the information necessary to understand the functional requirements.

Generally speaking, unless you've already gotten the information you need from somewhere else (and even then, you might want to conduct interviews to ensure that you haven't missed something), you'll want to interview the following people in the organization:

- Desktop support staff
- Server administrators
- Network administrators
- Storage administrators
- Security managers
- Compliance/legal staff

- Application owners
- Business leaders
- Project managers or project sponsors/owners
- Executive/managerial sponsors
- Architects

Not all designs or situations require you to speak with all these individuals, so be selective but thorough.

These individuals can provide you with information about the applications currently supported in the environment, the requirements of the applications, SLAs that are in place, dependencies between different applications or services in the data center, plans for future trends or projects, compliance or regulatory requirements, business-level requirements, financial objectives, operational aspects and workflow, and other facts that can be used to derive the functional requirements for the design.

## Assessing the Environment

After you've gathered the information necessary to determine the design's functional requirements, it's then necessary to assess the current environment. Assessing the environment fills a couple of purposes:

- The results of the assessment can, in some instances, verify or clarify the information provided during the information-gathering process of defining the functional requirements. People are just people and are known to make mistakes or accidentally omit information. By performing an assessment of the environment, you can verify that the applications you were told were present are, in fact, present.
- The assessment provides useful information necessary to complete the technical facet of the design. An assessment reveals the current types and configurations of the servers in the environment, the current network configurations, and the current storage configurations. All this information is crucial to putting together a new structure that will properly interoperate with the existing structure. If the organization is currently using iSCSI, then you know that implementing FC might create interoperability issues. Having this knowledge through an assessment of the current environment helps you tailor the technical facet of the design appropriately.

You can use a number of different tools or methods to assess the environment. If the organization already has a robust management system in place, it may have the inventory, configuration, and performance information you need. If not, you'll have to start digging through the environment, gathering information from such sources as these:

- Active Directory
- LDAP directories
- Network-management tools
- Enterprise-wide logging solutions

- IP addressing documentation
- Network equipment configurations
- Server performance data
- Server configuration data

You can imagine that in anything larger than most small environments, assessing the existing environment manually like this can be time-consuming and potentially error-prone. Fortunately, VMware and other vendors have released assessment tools that help gather this information in an automated fashion, to save you time and help you avoid missing critical data. Even the virtualization community has stepped up, providing scripts and other tools that gather information about existing physical and/or virtual environments.

Examples of some tools that have been created by vendors and community members include the following:

- VMware Capacity Planner
- Various community-supplied health-check scripts
- NetIQ PlateSpin Recon (formerly Novell PlateSpin PowerRecon)
- CiRBA

We'll discuss some of these tools in Chapter 10, "Monitoring and Capacity Planning." Because part of capacity planning involves previrtualization assessment, these tools are also useful in assessing an organization's existing environment in preparation for completing a design.

At this point, you're armed with some functional requirements, the information necessary to define other functional requirements, and knowledge of the existing environment. In some instances, before you're ready to assemble the design, you may first need to perform a gap analysis.

## Performing a Gap Analysis

Not every vSphere design is a *greenfield implementation*, meaning that not every design is a brand-new implementation in an environment where vSphere wasn't being used previously. In environments where an organization had already adopted VMware vSphere and is now upgrading or expanding the environment, or migrating to a new environment, you may need to perform a gap analysis.

A *gap analysis* is the process of comparing an environment's current state to its desired future state in order to determine what is needed to achieve that future state. Perhaps the current environment doesn't provide the scalability that is needed. A gap analysis can determine what portions of the design may limit the design's growth and point the way toward removing (or, at the very least, increasing sufficiently) those limits.

Once you have determined your functional requirements, gained knowledge about the existing environment, and performed a gap analysis (where necessary), you're ready to start assembling the design.

## Assembling the Design

Assembling the design is the iterative process we described earlier and depicted in Figure 1.3. While assembling the design, you make decisions within each of the three facets. Those decisions, focused on the what/who/how theme, are based on the functional requirements you've been given or have defined and are guided by the five design principles (AMPRS). Each decision has a cascading effect, forcing a series of what we call *downstream* decisions, as shown in Figure 1.8.

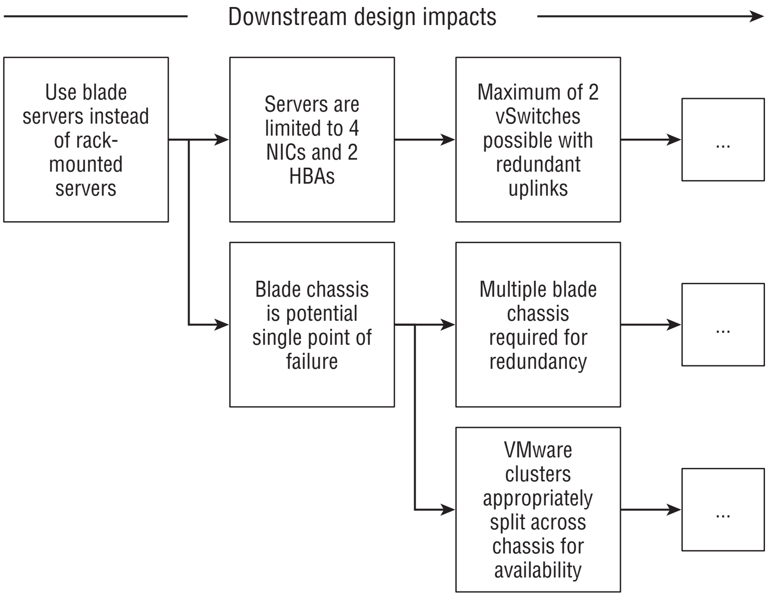

Downstream design impacts
Use blade servers instead of rack-mounted servers
Servers are limited to 4 NICs and 2 HBAs
Maximum of 2 vSwitches possible with redundant uplinks
...
Blade chassis is potential single point of failure
Multiple blade chassis required for redundancy
...
VMware clusters appropriately split across chassis for availability
...

**FIGURE 1.8** Each decision forces other decisions in a complex decision tree. The results of each decision must be compared against the functional requirements.

When you make a decision in the design, you then need to examine the result of that decision — and all the downstream decisions resulting from that decision — to ensure that you're still meeting all the functional requirements. If so, you continue; if not, you need to change that decision or violate one or more of the functional requirements.

### Violating Functional Requirements May Be Necessary

Sometimes, organizations have an unrealistic view of the functional requirements. In situations like this, it may be necessary to violate a functional requirement. As long as you can show why the functional requirement is violated and can provide a potential remediation for the violation, it's up to the organization to determine whether that functional requirement really is a requirement or the design can be accepted and implemented as described.

It's here, while assembling the design, that you define standards and best practices for the VMware vSphere environment:

**Standards** A good design defines standards for host names, networking configuration, storage layouts, resource allocation, virtual machine configurations, and so on. Standards are important because standardization reduces complexity. When complexity is reduced, operations are simplified, and costs are generally reduced. Without standards, environments become too complex to operate efficiently — and inefficiency is the bane of a VMware vSphere environment.

**Best Practices** A good design both defines and enforces best practices, where those best practices align with the organization's needs and functional requirements. The term *best practices* doesn't just mean the recommendations made by vendors for configuring their products; it also means the operational processes that the organization should follow. For example, a best practice states that all Windows-based VMs should have their file systems properly aligned in the virtual disk. Yes, that's a recommendation made by VMware, but it's also an operational recommendation made by a good design to ensure the efficient and stable operation of the environment. A good design includes other best practices that are specific to this particular implementation, such as the structure of new VMs, or the configuration for new datastores.

### Don't Accept Best Practices Blindly

When it comes to vendor best practices, don't accept them blindly. Examine those best practices and try to understand the reasons behind them. If the storage vendor says it's a best practice to use a particular multipathing policy, take a deeper look to understand why that multipathing policy is recommended. If a server vendor defines certain BIOS settings as best practices, try to figure out why those settings are used. Doing so will give you a deeper, more complete understanding of the environment and make you better prepared to provide implementation-specific best practices.

Assembling the design is a time-consuming and detailed process. Most, if not all, of the remaining chapters in this book focus on specific technical areas and the decisions you must make when building your VMware vSphere design.

## Documenting the Design

When the design has been assembled — that is, after you've made all the decisions that need to be made in each facet of the design, and you've compared the results of those decisions (and all the downstream decisions) against the functional requirements to ensure that the requirements are still being met — then you're ready to document the design.

This portion of the design process may happen concurrently with the assembly of the design and is, for the most part, straightforward. You should ensure that your documentation addresses each of the three facets of the design. In many instances, IT professionals tend to forget about the organizational and operational facets, but these are just as important as the technical facet and deserve equal treatment in your final design documentation.

In particular, be sure the design documentation includes at least the following documents:

- A description of the functional requirements that were provided to you or defined by you
- A comprehensive description of the technical facet decisions (the "what" decisions) made in the design

- A complete review of the organizational facet decisions (the "who" decisions) made in the design
- A thorough review of the operational facet decisions (the "how" decisions) found in the design
- Build documents — sometimes referred to as *blueprints* — that describe the specifics involved in building or constructing the design specified in your documentation
- Test plans describing how you can verify that the design satisfies the functional requirements
- A high-level architectural design document that ties together all these elements and tells the story of the design

### Performing the Implementation

After the design is complete and has been accepted by the organization, you may also have the opportunity to perform the implementation. Doing so affords you the opportunity to use your own documentation to build the environment.

If you aren't performing the implementation, then someone else will have to build your design. This is why it's important to be thorough and complete with the design documentation — when someone else comes along to build your design, that person won't have access to the thought processes inside your head. Be sure to provide as much information as possible in the design documentation so the build process is simple and straightforward.

## Summary

Throughout this book, we'll draw on our experience and knowledge to help you understand the complexities of VMware vSphere design. In this chapter, we've discussed the three key facets of design — the technical facet, the organizational facet, and the operational facet — and the importance of each. We've shared with you the five guiding principles of design, known as AMPRS: availability, manageability, performance, recoverability, and security. We've also discussed the process of design and some of the tasks involved in creating a design. In coming chapters, we'll take a more detailed look at the different areas of VMware vSphere design, starting in Chapter 2 with an examination of VMware ESXi.