Realtime publishers

Creating Unified IT Monitoring and Management in Your Environment

Don Jones

sponsored by

Chapter 2: Eliminating the Silos in IT Management 16
Too Many Tools Means Too Few Solutions 16
Domain-Specific Tools Don't Facilitate Cooperation 19
The Cloud Question: Unifying On-Premise and Off-Premise Monitoring 21
Missing Pieces 23
Not All of IT Is a Problem: Ordering, Routing, and Providing Services 27
Coming Up Next… 28

Copyright Statement

Chapter 2: Eliminating the Silos in IT Management

In the previous chapter, I proposed that one of the biggest problems in modern IT is the fact that we manage our environment in technology-specific silos: database administrators are in charge of databases, Windows admins are in charge of their machines, VMware admins run the virtualization infrastructure, and so forth. I'm not actually proposing that we change that exact practice—having domain-specific experts on the team is definitely a benefit. However, having these domain-specific experts each using their own unique, domain-specific tool definitely creates problems. In this chapter, we'll explore some of those problems, and see what we can do to solve them and create a more efficient, unified IT environment.

Too Many Tools Means Too Few Solutions

"Comparing apples to oranges" is an apt phrase when it comes to how we manage performance, troubleshooting, and other core processes in IT. Tell an Exchange Server administrator that there's a performance problem with the messaging system, and he'll likely jump right into Windows' Performance Monitor, perhaps with a pre-created counter set that focuses on disk throughput, processor utilization, RPC request count, and so forth—as shown in Figure 2.1.

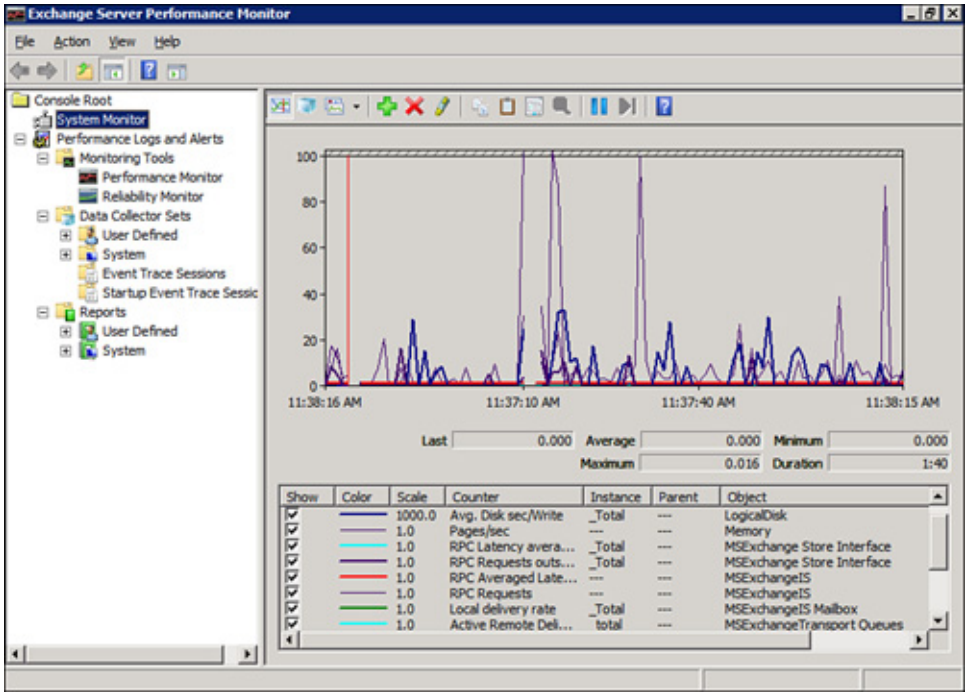

Figure 2.1: Monitoring Exchange.

If the Exchange administrator can't find anything wrong with the server, he might pass the problem over to someone else. Perhaps it will be the Active Directory administrator because Active Directory plays such a crucial role in Exchange's operation and performance. Out comes the Active Directory administrator's favorite performance tool, perhaps similar to the one shown in Figure 2.2. This is truly a domain-specific tool, with special displays and measurements that relate specifically to Active Directory.

Figure 2.2: Monitoring Active Directory.

If Active Directory looks fine, then the problem might be passed over to the network infrastructure specialist. Out comes another tool, this one designed to look at the performance of the organization's routers (see Figure 2.3).

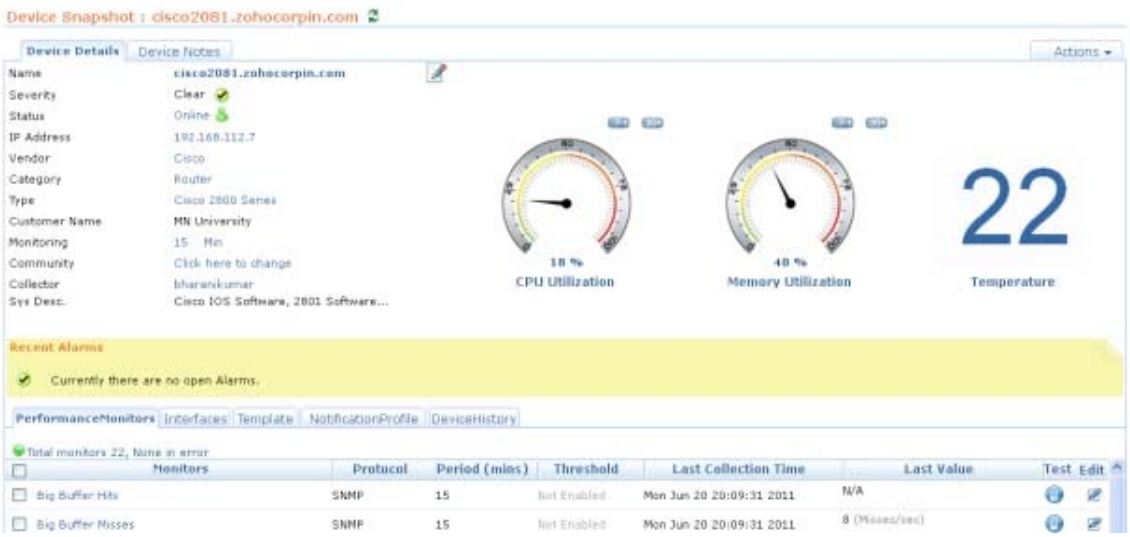

Figure 2.3: Monitoring router performance.

Combined, all of these tools have led these three specialists to the same decision: Everything's working fine. In spite of the fact that Exchange is clearly, from the users' point of view, *not* working fine, there's no evidence that points to a problem.

Simply put, this is a "too many tools, too few answers" problem. In today's complex IT environments, performance—along with other characteristics like availability and scalability—are the result of many components interacting with each other and working together. You can't manage IT by simply looking at one component; you have to look at entire *systems* of interacting, interdependent components.

Our reliance on domain-specific tools holds us back from finding the answers to our IT problems. That reliance also holds us back when it comes time to grow the environment, manage service level agreements (SLAs), and other core tasks. I've actually seen instances where domain-specific tools acted almost as blinders, preventing an expert who should have been able to solve a problem, or at least identify it, from doing so as quickly as he or she might have done.

Case Study

Heather is a database administrator for her organization. She's responsible for the entire database server, including the database software, the operating system (OS), and the physical hardware.

One day she receives a ticket indicating that users are experiencing sharply reduced performance from the application that uses her database. She whips out her monitoring tools, and doesn't see a problem. The server's CPU is idling along, disk throughput is well within norms, and memory consumption is looking good. In fact, she notices that the amount of workload being sent to the server is lower than she's used to seeing. That makes her suspect the network is having traffic jams, so she re-assigns the ticket to the company's infrastructure team. That team quickly re-assigns the ticket right back to her, assuring her that the network is looking a bit congested, but it's all traffic coming from *her* server.

Heather looks again, and sees that the server's network interface *is* humming along with a bit more traffic than usual. Digging deeper, she finally realizes that the server is experiencing a high level of CRC errors, and is thus having to retransmit a huge number of packets. Clients experience this problem as a general slowdown because it takes longer for undamaged packets to reach their computers.

Heather's focus on her specific domain expertise led her to "toss the problem over the wall" to the infrastructure team, wasting time. Because she wasn't accustomed to looking at her server's network interface, she didn't check it as part of her routine performance troubleshooting process.

Domain-Specific Tools Don't Facilitate Cooperation

If the components of our complex IT systems are cooperative and interdependent, our IT professionals are often anything but. In other words, IT management tends to encourage the silos that are built around specific technology domains. There's the database administration group, the Active Directory group, the infrastructure group, and so forth. Even companies that practice "matrix management," in which multiple domain experts are grouped into a functional team, still tend to accept the silos around each technical domain.

There are two major reasons that these silos persist, and almost any IT professional can describe them to you:

- "I don't know anything about *that*." Each domain expert is an expert in *his* technical area. The database administrator isn't proficient at monitoring or managing routers, and doesn't especially want to work with them anyway. There's little real value in extensive technical cross-training for most organizations, simply because their staff doesn't have the time. Devoting time to secondary and tertiary disciplines reduces the amount of time available for their primary job responsibilities.
- "I don't want anyone messing with my stuff." IT professionals want to do a good job, and they're keenly aware that most problems come about as the result of change. Allow someone to change something, and you're asking for trouble. If someone changes something in your part of the environment, and you don't know about their activity, you'll have a harder time fixing any resulting problems.

Both of these reasons are completely valid, and I'm in no way suggesting that everyone on the IT team become an expert in every technology that the organization must support. However, the attitudes reflected in these two perspectives require some minor adjustment.

One reason I keep coming back to domain-specific tools is because they *encourage* this kind of walled-garden separation, and do nothing to encourage even the most cursory cooperation between IT specialists. Cooperation, when it exists, comes about through good human working relationships—and those relationships often struggle with the fact that each specialist is looking at a different set of data and working from a different "sheet of music," so to speak. I've been in environments and seen administrators spend hours arguing about whose "fault" something was, each pointing to their own domain-specific tools as "evidence."

Case Study

Dan is an Active Directory administrator for his company, and is responsible for around two dozen domain controllers, each of which runs in a virtual machine. Peg is responsible for the organization's virtual server infrastructure, and manages the physical hosts that run all of the virtual machines.

One afternoon, Peg gets a call from Dan. Dan's troubleshooting a performance problem on some of the domain controllers, and suspects that something is consuming resources on the virtualization host that his domain controllers need.

Peg opens her virtual server console and assures Dan that the servers aren't maxed out on either physical CPU or memory, and that disk throughput is well within expected levels. Dan counters by pointing to his Active Directory monitoring tools, which show maxed-out processor and memory statistics, and lengthening disk queues that indicate data isn't being written to and read from disk as quickly as it should be. Peg insists that the physical servers are fine. Dan asks if the virtual machines settings have been reconfigured to provide fewer resources to them, and Peg tells him no.

The two go back and forth like this for hours. They're each looking at different tools, which are telling them completely different things. Because they're not able to speak a common technology language, they're not able to work together to solve the problem.

We *don't* need to have every IT staffer be an expert in every IT technology; we *do* need to make it easier for specialists to cooperate with one another on things like performance, scalability, availability, and so forth. That's difficult to do with domain-specific tools. The router administrator doesn't *want* a set of database performance-monitoring tools, and the database administrator doesn't especially want the router admin to *have* those tools. Having domain-specific tools for someone else's technical specialization is exactly how the two attitudes I described earlier come into play.

Ultimately, the problem can be solved by having a unified tool set. Get everyone's performance information onto the same screen. That way, everyone is playing from the same rule book, looking at the same data—and that data reflects the entire, interdependent environment. Everyone will be able to see where the problem lies, *then* they can pull out the domain-specific tools to start fixing the actual problem area, if needed.

The Cloud Question: Unifying On-Premise and Off-Premise Monitoring

This concept of a unified monitoring console becomes even more important as organizations begin shifting more of their IT infrastructure into "the cloud."

The Cloud Is Nothing New

I have to admit that I'm not a big fan of "the cloud" as a term. It's very sales-and-marketing flavored, and the fact is that it isn't a terribly new concept.

Organizations have outsourced IT elements for years. Probably the most-outsourced component is Web hosting, either outsourcing single Web sites into a shared-hosting environment, or outsourcing collocated servers into someone else's data center.

> For the purposes of this discussion, "the cloud" simply refers to some IT element being outsourced in a way that abstracts the underlying infrastructure. For example, if you have collocated servers in a hosting company's data center, you don't usually have details about their internal network architecture, their Internet connectivity, their routers, and so forth—the data center is the piece you're paying to have abstracted for you. In a modern cloud computing model like Windows Azure or Amazon Elastic Cloud, you don't have any idea what physical hosts are running your virtual machines—that physical server level is what you're paying to have abstracted, along with supporting elements like storage, networking, and so on. For a Software as a Service (SaaS) offering, you don't even know what virtual machines might be involved in running the software because you're paying to have the entire underlying infrastructure abstracted.

Regardless which bits of your infrastructure wind up in some outsourced service provider's hands, those bits are still *a part of your business.* Critical business applications and processes rely on those bits functioning. You simply have less control over them, and typically have less insight into how well they're running at any given time.

This is where domain-specific tools fall apart completely. Sure, part of the whole point of outsourcing is to let someone else worry about performance—but outsourced IT still supports *your* business, so you at least need the ability to see how the performance of outsourced elements is affecting the rest of your environment. If nothing else, you need the ability to authoritatively "point the finger" at the specific cause of a problem—even if that cause is an outsourced IT element, and you can't directly effect a solution. This is where unified monitoring truly earns a place within the IT environment. For example, Figure 2.4 shows a very simple "unified dashboard" that shows the overall status of several components of the infrastructure—including several outsourced components, such as Amazon Web Services.

Figure 2.4: Unified monitoring dashboard.

The idea is to be able to tell, at a glance, where performance is failing, to drill through for more details, and then to either start fixing the problem—if it exists on your end of the cloud—or escalate the problem to someone who can.

Let's be very clear on one thing: Any organization that's outsourcing *any* portion of its business IT environment and cannot monitor the basic performance of those outsourced elements is going to be in big trouble when something eventually goes wrong. Sure, you have SLAs with your outsourcing partners—but *read* those SLAs. Typically, they only commit to a refund of whatever fees you pay if the SLA isn't met. That does nothing to compensate you for lost business that results from the unmet SLA. It's in your best interests, then, to keep a close watch on performance. That way, when it *starts* to go bad, you can immediately contact your outsourcing partner and get someone working on a fix so that the impact on your business can at least be minimized.

Missing Pieces

There's another problem when it comes to performance monitoring and management, scalability planning, and so forth: missing pieces. Our technology-centric approach to IT tends to give us a myopic view of our environment. For example, consider the diagram in Figure 2.5. This is a typical (if simplified) diagram that any IT administrator might create to help visualize the components of a particular application.

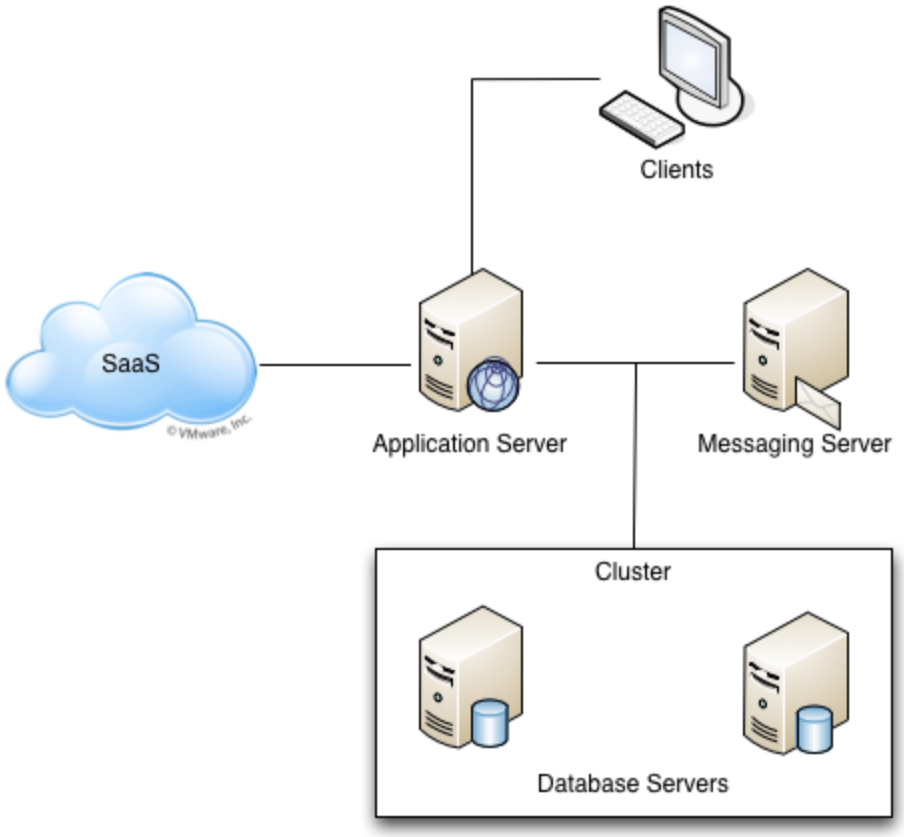

Figure 2.5: Application diagram.

The problem is that there are obviously missing pieces. For example, where's the infrastructure? Whoever created this diagram clearly doesn't have to deal with the infrastructure—routers and switches and so forth—so they didn't include it. It's assumed, almost abstracted like an outsourced component of the infrastructure. Maybe Figure 2.6 is a more accurate depiction of the environment.

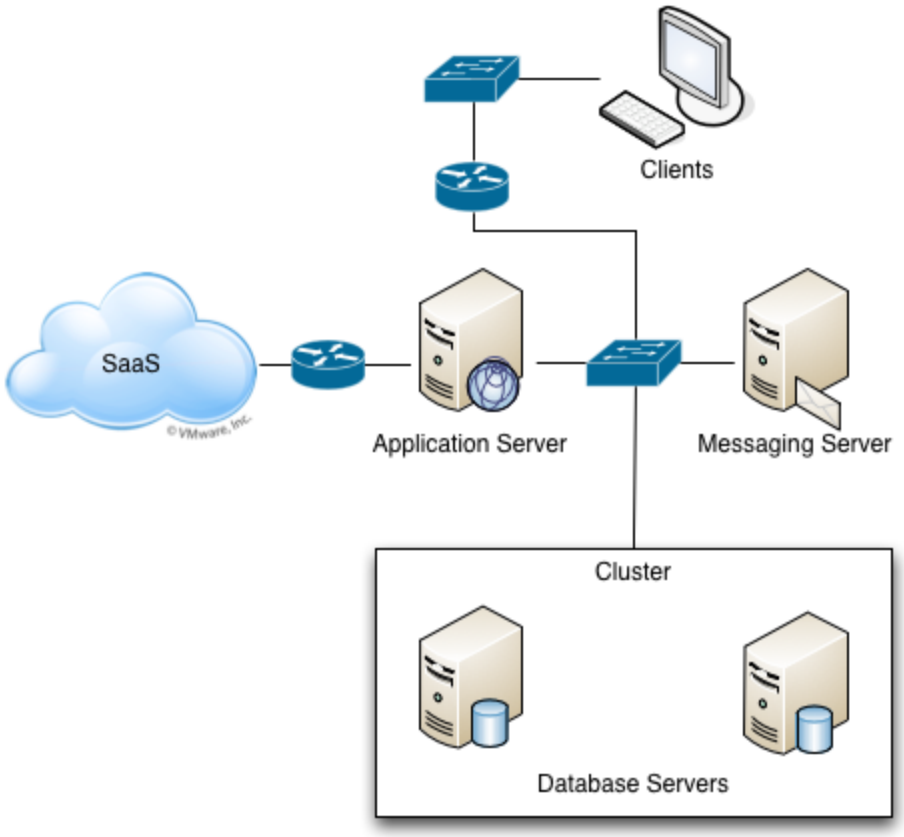

Figure 2.6: Expanded application diagram.

And even with this diagram, there are still probably missing pieces. This reality is probably one of the biggest dangers in IT management today: We forget about pieces that are outside our purview.

Again, this is where a unified monitoring system can create an advantage. Rather than focusing on a single area of technology—like servers—it can be technology-agnostic, focusing on *everything.* There's no need to leave something out simply because it doesn't fit within the tool's domain of expertise; *everything* can be included.

In fact, an even better approach is to focus on unified monitoring tools that can actually go out and *find* the components in the environment. Software doesn't have to make the same assumptions, or have the same technology prejudices, as humans. A unified monitoring console doesn't care if you happen to be a Hyper-V expert, or if you prefer Cisco routers over some other brand. It can simply take the environment as it is, discovering the various components and constructing a real, accurate, and complete diagram of the environment. It can then start monitoring those components (perhaps prompting you for credentials for each component, if needed), enabling you to get that complete, all-in-one, unified dashboard. I've *been* in environments where not using this kind of auto-discovery became a real problem.

Case Study

Terry is responsible for the infrastructure components that support his company's primary business application. Those components include routers, switches, database servers, virtualization hosts, messaging servers, and even an outsourced SaaS sales management application. Terry's heard about the unified monitoring idea, and his organization has invested in a service that provides unified monitoring for the environment. Terry's carefully configured each and every component so that everything shows up in the monitoring solution's dashboard.

One afternoon, the entire application goes down. Terry leaps to the unified monitoring console, and sees several "alarm" indications. He drills down and discovers that the connection to the SaaS application is unavailable. Drilling further, he sees that the router for that connection is working fine, and that the firewall is up and responsive. He's at a complete loss.

Several hours of manual troubleshooting and wire-tracing reveal something about the environment that Terry didn't know: There's a router on the *other* side of the firewall as well, and it's failed. Normal Internet communications are still working because those travel through a different connection, but the connection that carries the SaaS application's traffic is offline. The "extra" router is actually a legacy component that pretty much everyone had forgotten about.

A monitoring solution capable of automated discovery wouldn't have "forgotten," though. It could have detected the extra router and included it in Terry's dashboard, making it much easier for him to spot the problem. In fact, it might have prompted him to replace or remove that router much earlier, once he realized it existed.

Discovery can also help identify components that don't fit neatly within our technology silos, and that don't "belong" to anyone. Infrastructure components like routers and switches are commonly-used examples of these "orphan" components because not every organization maintains a dedicated infrastructure specialist to support these devices. However, legacy applications and servers, specialty equipment, and other components can all be overlooked when they're not anyone's specific area of responsibility. Discovery helps keep us from overlooking them.

Not All of IT Is a Problem: Ordering, Routing, and Providing Services

Most organizations tend to get into the habit of thinking of their IT department as "fire fighters." IT exists to solve problems. That isn't true, of course, and any organization probably (hopefully) depends more on IT to carry out day-to-day tasks and requests more than they rely on them to solve problems. But the day-to-day tasks are easy to overlook, whereas "fire fighting" gets everyone's attention.

The result of this way of thinking is that IT management tends to focus on tools that help make problem-solving easier. Unified monitoring is exactly that kind of tool: If nothing ever went wrong, we wouldn't need it. It's there to make problem-solving faster, primarily in the areas of performance and availability. Right?

Not quite. *Truly* unified *management* also entails making day-to-day IT tasks easier for everyone involved. Users, for example, need to order and receive routine services, from simple password resets and account unlocks to new hardware and software requests. I'll make what some consider to be a bold statement and say that those routine requests should be treated in the exact same way as a problem. Look at any IT management framework, such as ITIL, and you'll find that concept runs throughout: Routine IT requests should be part of a unified *management* process, which also includes problem-solving.

Consider some of these broad functional capabilities that a unified management (versus mere "monitoring") can offer both to problem-solving activities and to routine IT services:

- Workflow—When problems arise, following a structured process, or workflow, can help make problem-solving more consistent and efficient. Similarly, structured workflows can help make routine IT services more efficient and consistent. The workflows will be different for problem-solving and for various routine services, but having the ability to manage and monitor workflows can be a real benefit.
- Approvals—Workflows should include approvals. This capability is most obvious for routine services like hardware and software requests, security requests, and so on—but it can be just as important for problem solving. Not every problem can be fixed by changing a setting or rebooting a device; sometimes you'll need to make a more significant change, and having the ability to formally route approval to make that change is a benefit.

- Routing. The specialist who fixes a problem is usually the last one to hear about it. Front-line resources, such as your Help desk and your end users, are the first "responders." Being able to select a problem category and have a ticket routed to the right individual helps speed problem resolution. The same is true for routine services: Things get done quicker when the right person has the request. Automated routing capabilities can help get the right person on the job more quickly and more accurately.
- Self-service. Reducing phone calls and manual email juggling is crucial to achieving better efficiency. Self-service can help do that for both problems and routine requests. When users experience a problem, self-service can allow them to submit tickets as well as help them solve the problem on their own, through a knowledge base. When users need routine service, self-service helps them submit that request without having to engage additional IT services.
- Service catalog. Part of self-service is the ability to create an "online store" for services that users can request.

There are more capabilities, of course, but we'll cover them in upcoming chapters. These are simply some of the basic capabilities that we need in order to make both routine IT requests *and* problem-solving more consistent and efficient.

Coming Up Next…

This chapter has been about breaking down the silos between technology specialties, or at least building doorways between them. That helps to solve one of the major problems in modern IT monitoring and management. The next chapter will tackle a somewhat more complicated problem: Keeping everyone in the management loop. It's about improving communications. Unfortunately, communications are too often a voluntary, secondary exercise—we have to make an *effort* to communicate, and when we're really feeling the pressure, it's easy to want to put that effort elsewhere. So we need to adopt processes and tools that make communications more automatic, helping keep everyone in the loop *without* requiring a massive secondary effort to do so.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.