Chapter 21

Centralization and Decentralization

This chapter seeks to help an SA decide how much centralization is appropriate, for a particular site or service, and how to transition between more and less centralization.

Centralization means having one focus of control. One might have two DNS servers in every department of a company, but they all might be controlled by a single entity. Alternatively, **decentralized** systems distribute control to many parts. In our DNS example, each of those departments might maintain and control its own DNS server, being responsible for maintaining the skill set to stay on top of the technology as it changes, to architect the systems as it sees fit, and to monitor the service. Centralization refers to nontechnical control also. Companies can structure IT in a centralized or decentralized manner.

Centralization is an attempt to improve efficiency by taking advantage of potential economies of scale: improving the average; it may also improve reliability by minimizing opportunities for error. Decentralization is an attempt to improve speed and flexibility by reorganizing to increase local control and execution of a service: improving the best case. Neither is always better, and neither is always possible in the purest sense. When each is done well, it can also realize the benefits of the other: odd paradox, isn't it?

Decentralization means breaking away from the prevailing hegemony, revolting against the frustrating bureaucratic ways of old. Traditionally, it means someone has become so frustrated with a centralized service that "do it yourself" has the potential of being better. In the modern environment decentralization is often a deliberate response to the faster pace of business and to customer expectations of increased autonomy.

Centralization means pulling groups together to create order and enforce process. It is cooperation for the greater good. It is a leveling process. It seeks to remove the frustrating waste of money on duplicate systems, extra work, and manual processes. New technology paradigms often bring opportunities for centralization. For example, although it may make sense for each department to have slightly different processes for handling paper forms, no one department could fund building a pervasive web-based forms system. Therefore, a disruptive technology, such as the web, creates an opportunity to replace many old systems with a single, more efficient, centralized system. Conversely, standards-based web technology can enable a high degree of local autonomy under the aegis of a centralized system, such as delegated administration.

21.1 The Basics

At large companies in particular, it seems as if every couple of years, management decides to centralize everything that is decentralized and vice versa. Smaller organizations encounter similar changes driven by mergers or opening of new campuses or field offices. In this section, we discuss guiding principles you should consider before making such broad changes. We then discuss some services that are good candidates for centralization and decentralization.

21.1.1 Guiding Principles

There are several guiding principles related to centralization and decentralization. They are similar to what anyone making large, structural changes should consider.

- **Problem-Solving:** *Know what specific problem you are trying to solve.* Clearly define what problem you are trying to fix. “Reliability is inconsistent because each division has different brands of hardware.” “Services break when network connections to sales offices are down.” Again, write down the specific problem or problems and communicate these to your team. Use this list as a reality check later in the project to make sure that you haven’t lost sight of the goal. If you are not solving a specific problem, or responding to a direct management request, stop right here. Why are you about to make these changes? Are you sure this is a real priority?
- **Motivation:** *Understand your motivation for making the change.* Maybe you are seeking to save money, increase speed or become more flexible.

Maybe your reasons are political: You are protecting your empire or your boss, making your group look good, or putting someone's personal business philosophy into action. Maybe you are doing it simply to make your own life easier; that's valid too. Write down your motivation and remind yourself of it from time to time to verify that you haven't strayed.

- **Experience Counts:** *Use your best judgment.* Sometimes, you must use experience and a hunch rather than specific scientific measurements. For example, we've found that when centralizing email servers, our experience has developed these rules of thumb: Small companies—five departments with 100 people—tend to need one email server. Larger companies can survive with an email server per thousands of people, especially if there is one large headquarters and many smaller sales offices. When the company grows to the point of having more than one site, each site tends to require its own email server but is unlikely to require its own Internet gateway. Extremely large or geographically diverse companies start to require multiple Internet gateways at different locations.
- **Involvement:** *Listen to the customers' concerns.* Consult with customers to understand their expectations: Retain the good aspects and fix the bad ones. Focus on the qualities that they mention, not the implementation. People might say that they like the fact that "Karen was always right there when we needed new desktop PCs installed." That is an implementation. The new system might not include on-site personnel. What should be retained is that the new service has to be responsive—as responsive as having Karen standing right there. That may mean the use of overnight delivery services or preconfigured and "ready to eat" systems[1] stashed in the building, or whatever it takes to meet that expectation. Alternatively, you must do expectation setting if the new system is not going to deliver on old expectations. Maybe people will have to plan ahead and ask for workstations a day in advance.
- **Be Realistic:** *Be circumspect about unrealistic promises.* You should thoroughly investigate any claims that you will save money by decentralizing, add flexibility by centralization, or have an entirely new system without pain: The opposite is usually the case. If a vendor promises that a new product will perform miracles but requires you to centralize

[1] Do not attempt to eat a computer. "Ready to eat" systems are hot spares that will be fully functional when powered up: absolutely no configuration files to modify and so on.

(or decentralize) how something is currently organized, maybe the benefits come from the organizational change, not the product!

- **Balance:** *Centralize as much as makes sense for today, with an eye toward the future.* You must find the balance between centralization and decentralization. There are time considerations: Building the perfect system will take forever. You must set realistic goals yet keep an eye to future needs. For example, in 6 months, the new system will be complete and then will be expected to process a million widgets per day. However, a different architecture will be required to process 2 million widgets per day, the rate that will be needed a year later, and will require considerably more development time. You must balance the advantage of having a new system in 6 months—with the problem of needing to start building the next-generation system immediately—versus the advantage of waiting longer for a system that will not need to be replaced so soon.
- **Access:** *The more centralized something is, the more likely it is that some customers will need a special feature or some kind of customization.* An old business proverb is: "All of our customers are the same: They each have unique requirements." One size never fits all. You can't do a reasonable job of centralizing without being flexible; you'll doom the project if you try. Instead, look for a small number of models. Some customers require autonomy. Some may require performing their own updates, which means creating a system of access control so that customers can modify their own segments without affecting others.
- **No Pressure:** *It's like rolling out any new service.* Although more emotional impact may be involved than with other changes, both centralization and decentralization projects have issues similar to building a new service. That said, new services require careful coordination, planning, and understanding of customer needs to succeed.
- **110 Percent:** *You have only one chance to make a good first impression.* A new system is never trusted until proven a success, and the first experience with the new system will set the mood for all future customer interactions. Get it right the first time, even if it means spending more money up front or taking extra time for testing. Choose test customers carefully, making sure they trust you to fix any bugs found while testing, and won't gossip about it at the coffee machine. Provide superior service the first month, and people will forgive later mistakes. Mess up right at the start, and rebuilding a reputation is nearly impossible.

- **Veto Power:** *Listen to the customers, but remember that management has the control.* The organizational structure can influence the level of centralization that is appropriate or possible. The largest impediment to centralization often is management decisions or politics. Lack of trust makes it difficult to centralize. If the SA team has not proved itself, management may be unwilling to support the large change. Management may not be willing to fund the changes, which usually indicates that the change is not important to them. For example, if the company hasn't funded a central infrastructure group, SAs will end up decentralized. It may be better to have a central infrastructure group; lacking management support, however, the fallback is to have each group make the best subinfrastructure it can—ideally, coordinating formally or informally to set standards, purchase in bulk, and so on. Either way, the end goal is to provide excellent service to your customers.

21.1.2 Candidates for Centralization

SAs continually find new opportunities to centralize processes and services. Centralization does not innately improve efficiency. It brings about the opportunity to introduce new economies of scale to a process. What improves efficiency is standardization, which is usually a by-product of centralization. The two go hand in hand.

The cost savings of centralization come from the presumption that there will be less overhead than the sum of the individual overheads of each decentralized item. Centralization can create a simpler, easier-to-manage architecture. One SA can manage a lot more machines if the processes for each are the same.

To the previous owners of the service being centralized, centralization is about giving up control. Divisions that previously provided their own service now have to rely on a centralized group for service. SAs who previously did tasks themselves, their own way, now have to make requests of someone else who has his or her own way to do things. The SAs will want to know whether the new service provider can do things better.

Before taking control away from a previous SA or customer, ask yourself what the customer's psychological response will be. Will there be attempts to sabotage the effort? How can you convince people that the new system will be better than the old system? How will damage control and rumor control be accomplished? What's the best way to make a good first impression?

The best way to succeed in a centralization program is to pick the right services for centralization. Here are some good candidates.

- **Distributed Systems:** *Management of distributed systems.* Historically, each department of an organization configured and ran its own web servers. As the technology got more sophisticated, less customization of each web server was required. Eventually, there was no reason not to have each web server configured exactly the same way, and the need for rapid updates of new binaries was becoming a security issue. The motivation was to save money by not requiring each department to have a high level of web server expertise. The problem being fixed was the lack of similar configurations on each server. A system was designed to maintain a central configuration repository that would update each of the servers in a controlled and secure manner. The customers affected were the departmental SAs, who were eager to give up a task that they didn't always understand. By centralizing web services, the organization could also afford to have one or more SAs become better-trained in that particular service, to provide better in-house customer support.
- **Consolidation:** *Consolidate services onto fewer hosts.* In the past, for reliability's sake, one service was put on each physical host. However, as technology progresses, it can be beneficial to have many services on one machine. The motivation is to decrease cost. The problem being fixed is that every host has overhead costs, such as power, cooling, administration, machine room space, and maintenance contracts. Usually, a single, a more powerful machine costs less to operate than several smaller hosts. As services are consolidated, care must be taken to group customers with similar needs.

  Since the late 1990s, **storage consolidation** has been a big buzzword. By building one large storage-area network that each server accesses, there is less "stranded storage"—partially-full disks—on each server. Often, storage consolidation involves decommissioning older, slower, or soon-to-fail disks and moving the data onto the SAN, providing better performance and reliability.

  Server virtualization, a more recent trend, involves using virtual hosts to save hardware and license costs. For example, financial institutions used to have expensive servers and multiple backup machines to run a calculation at a particular time of the day, such as making end-of-day transactions after the stock market closes. Instead, a virtual machine can be spun up shortly before the market closes; the machine runs its tasks, then spins down. Once it is done, the server is free to run other virtual machines that do other periodic tasks.

By using a global file system, such as a SAN, a **virtualization cluster** can be built. Since the virtual machine images—the data stored on disk that defines the state of a virtual machine—can be accessed from many hardware servers, advanced virtualization management software can migrate virtual machines between physical machines with almost unnoticable switch-over time. Many times, sites realize that they need many machines, each performing a particular function, none of which requires enough CPU horsepower to justify the cost of dedicated hardware. Instead, the virtual machines can share a farm, or cluster, of physical machines, as needed. Since virtual machines can migrate between different hardware nodes, workload can be rebalanced. Virtual machines can be moved off an overloaded physical machine. Maintenance becomes easier too. If one physical machine is showing signs of hardware problems, virtual machines can be migrated off it onto a spare machine with no loss of service; the physical machine can then be repaired or upgraded.

- **Administration:** *System administration.* When redesigning your organization (see Chapter 30), your motivation may be to reduce cost, improve speed, or provide services consistently throughout the enterprise. The problem may be the extra cost of having technical management for each team or that the distributed model resulted in some divisions' having poorer service than others. Centralizing the SA team can fix these problems.

  To provide customization and the "warm fuzzies" of personal attention, subteams might focus on particular customer segments. An excellent example of this is a large hardware company's team of "CAD ambassadors," an SA group that specializes in cross-departmental support of CAD/CAM tools throughout the company. However, a common mistake is to take this to an extreme. We've seen at least one amazingly huge company that centralized to the point that "customer liaisons" were hired to maintain a relationship with the customer groups, and the customers hired liaisons to the centralized SA staff. Soon, these liaisons numbered more than 100. At that point, the savings in reduced overhead were surely diminished. A regular reminder and dedication to the original motivation may have prevented that problem.

- **Specialization:** *Expertise.* In decentralized organizations, a few of the groups are likely to have more expertise in particular areas than other groups do. This is fine if they maintain casual relationships and help one another. However, certain expertise can become critical to business, and therefore an informal arrangement becomes an unacceptable business risk. In that case, it may make sense to consolidate that expertise into one group. The motivation is to ensure that all divisions have access to a minimum level of expertise in one specific area or areas. The problem is that the lack of this expertise causes uneven service levels, for example, if one division had unreliable DNS but others didn't or if one division had superior Internet email service, whereas others were still using UUCP-style addresses. (If you are too young to remember UUCP-style addresses, just count your blessings.) That would be intolerable!

  Establishing a centralized group for one particular service can bring uniformity and improve the average across the entire company. Some examples of this include such highly specialized skills as maintaining an Internet gateway, a software depot, various security issues—VPN service, intrusion detection, security-hole scanning, and so on—DNS, and email service. A common pattern at larger firms is to create a "Care Services" or "Infrastructure" team to consolidate expertise in these areas and provide infrastructure across the organization.
- **Left Hand, Right Hand:** *Infrastructure decisions.* The creation of infrastructure and platform standards can be done centrally. This is a subcase of centralizing expertise. The motivation at one company was that that infrastructure costs were high and interoperability between divisions was low. There were many specific problems to be solved. Every division had a team of people researching new technologies and making decisions independently. Each team's research duplicated the effort of the others. Volume-purchasing contracts could not be signed, because each individual division was too small to qualify. Repair costs were high because so many different spare parts had to be purchased. When divisions did make compatible purchasing decisions, multiple spare parts were still being purchased because there was no coordination or cooperation. The solution was to reduce the duplication in effort by having one standards committee for infrastructure and platform standards. Previously, new technology was often adopted in pockets around the company because some divisions were less averse to risk; these became

the divisions that performed product trials or became early adopters of new technology.

This last example brings up another benefit of centralization. The increased purchasing power should mean that better equipment can be purchased for the same cost. Vendors may provide better service, as well as preferred pricing, when they deal with a centralized purchasing group that reflects the true volume of orders from that one source. Sometimes, money can be saved through centralization. Other times, it is better to use the savings to invest in better equipment.

- **Commodity:** *If it has become a commodity, consider centralization.* A good time to consider centralizing something is when the technology path it has taken has made it a commodity. Network printing, file service, email servers, and even workstation maintenance used to be unique, rare technologies. However, now these things are commodities and excellent candidates for centralization.

Case Study: Big, Honkin' File Servers

Tom's customers and even fellow SAs fought long and hard against the concept of large, centralized file servers. The customers complained about the loss of control and produced, in Tom's opinion, ill-conceived pricing models that demonstrated that the old UNIX-based file servers were the better way to go. What they were really fighting was the notion that network file service was no longer very special; it had become a commodity and therefore an excellent candidate for centralization. Eventually, an apples-to-apples comparison was done. This included a total cost-of-ownership model that included the SA time and energy to maintain the old-style systems. The value of some unique features of the dedicated file servers, such as file system snapshot, was difficult to quantify. However, even when the cost model showed the systems to cost about the same per gigabyte of usable storage, the dedicated file servers had an advantage over the old systems: consistency and support. The old systems were a mishmash of various manufacturers for the host; for the RAID controllers; and for the disk drives, cables, network interfaces, and, in some cases, even the racks they sat in! Each of these items usually required a level of expertise and training to maintain efficiently, and no single vendor would support these Frankenstein monsters. Usually, when the SA who purchased a particular RAID device left the group, the expertise left with the person. Standardizing on a particular product resulted in a higher level of service because the savings were used to purchase top-of-the line systems that had fewer problems than inexpensive competitors. Also, having a single phone number to call for support was a blessing.

Printing is another commodity service that has many opportunities for centralization, both in the design of the service itself and when purchasing supplies. Section 24.1.1 provides more examples.

21.1.3 Candidates for Decentralization

Decentralization does not automatically improve response times. When done correctly, it creates an opportunity to do so. Even when the new process is less efficient or is inefficient in different ways, people may be satisfied simply to be in control. We've found that people are more tolerant of a mediocre process if they feel they control it.

Decentralization often trades cost efficiency for something even more valuable. In these examples, we decentralize to democratize control, gain fault tolerance, acquire the ability to have a customized solution, or remove ourselves from clue-lacking central authorities. ("They're idiots, but they're *our* division's idiots.") One must seek to retain what was good about the old system while fixing what was bad.

Decentralization democratizes control. The new people gaining control may require training; this includes both the customers and the SAs. The goal may be autonomy, the ability to control one's own destiny, or the ability to be functional when disconnected from the network. This latter feature is also referred to as **compartmentalization,** the ability to achieve different reliability levels for different segments of the community. Here are some good candidates for decentralization.

- *Fault tolerance.* The duplication of effort that happens with decentralization can remove single points of failure. A company with growing field offices required all employees to read email off servers located in the headquarters. There were numerous complaints that during network outages, people couldn't read or even compose email, because composition required access to directory servers that were also at the headquarters. Divisions in other time zones were particularly upset that maintenance times at the headquarters were their prime working hours. The motivation was to increase reliability, in particular access during outages. The problem was that people couldn't use email when WAN links were down. The solution was to install local LDAP caches and email servers in each of the major locations. (It was convenient and effective to also use this host for DNS, authentication, and other services.) Although mail would not be transmitted site to site during an outage, customers could access their email store, local email could be delivered, and messages that needed to be relayed to other sites would transparently queue until the WAN link recovered. This could have been a management disaster if each site was expected to have the expertise required to configure and maintain such systems or if different sites created different standards. Instead, it was a big success because management was centralized. Each site received preconfigured hardware and software that simply needed to be plugged in. Updates were done via a centralized system. Even backups could be performed remotely, if required.

- *Customization.* Sometimes, certain customer groups have a business requirement to be on the bleeding edge of technology, whereas others require stability. A research group required early access to technology, usually before it was approved by corporate infrastructure standards committees. The motivation was largely political because the group maintained a certain status by being ahead of others within the company, as well as in the industry. There was also business motivation: The group's projects were far-reaching and futuristic, and the group needed to "live in the future" if it was going to build systems that would work well in the networks of the future. The problem was that the group was being prevented from deviating from corporate standards. The solution was to establish a group-specific SA team. The team members participated in the committees that created the corporate standards and were able to provide valuable feedback because they had experience with technologies that the remainder of the committee was just considering. Providing this advice also maintained the groups elite status. Their participation also guaranteed that they would be able to establish interoperability guidelines between their "rogue" systems and the corporate standards. This local SA team was able to meet the local requirements that were different from those of the rest of the company. They could provide special features and select a different balance of stability versus cutting-edge features.
- *Meeting your customers' needs.* Sometimes, the centralized services group may be unable to meet the demands placed on it by some of the departments in the company. Before abandoning the centralized service, try to understand the reason for the failures of the central group to meet your customers' needs. Try to work with the customers to find a solution that works for both SAs and customers, such as the one described earlier. Your ultimate responsibility is to meet your customers'

needs and to make them successful. If you cannot make the relationship with the central group work, your company may have to decentralize the necessary services so that you can meet your group's needs. Make sure that you have management support to make this move; be aware of the pitfalls of decentralization, and try to avoid them. Remember why you moved to a centralized model, and periodically reevaluate whether it still makes sense.

Advocates of decentralization sometimes argue that centralized services are single points of failure. However, when centralization is done right, the savings can be reinvested into technology that increases fault tolerance. Often, the result of decentralization is many single points of failure spread all over the company; redundancy is reduced. For example, when individual groups build their own LANs, they might have the training only to set up a very basic, simple LAN infrastructure. The people doing the work have other responsibilities that keep them from being able to become experts in modern LAN technology. When LAN deployment is centralized, people who specialize in networking and do it full-time are in charge. They have the time to deploy redundancy protocols and proactive monitoring that will enable them to fix problems within a defined SLA. The savings from volume discounts often will pay for much of the redundancy. The increased reliability from a professional SLA-driven design and operation process benefits the company in many ways.

Another point in support of decentralization is that there are benefits to having diversity in your systems. For example, different OSs have different security problems. It can be beneficial to have only a fraction of your systems taken out by a virus. A major software company had a highly publicized DNS outage because all of its DNS servers were running the same OS and the same release of DNS software. If the company had used a variety of OSs and DNS software, one of them might not have been susceptible to the security hole that was being leveraged. If you are the centralized provider, accept that this may sometimes be necessary.

21.2 The Icing

Centralization and decentralization can be major overhauls. If you are asking people to accept the pain of converting to a new system, you should be proposing a system that is not only cheaper but also better for them.

There is an old adage that often appears on buttons and bumper stickers: "Cheap, fast, good. Pick two." This pointed statement reveals a time-tested truism. In general, you must sacrifice one of those three items to achieve the other two. In fact, if someone tries to claim that they provide all three simultaneously, look under the tablecloth of their slick demo and check for hidden wires. This section describes some examples that achieved or promised to achieve all three. Some, like the purchasing example in 21.2.1, were a great overall success. Others had mixed results.

21.2.1 Consolidate Purchasing

In this example, centralization resulted in better products more quickly delivered for less money. An SA group was able to position itself to approve all computer-related purchasing for its division. In fact, the group was able to have the purchasing agent who handled such purchases moved into its group so they could work closely on contracts, maintenance agreements, and so on. As a result, the group was able to monitor what was being purchased. Particular purchases, such as servers, would alert the SAs to contact customers to find out what special requirements the server would create: Did it need machine room space, special networking, or configuration? This solved a problem whereby customers would blindside the SAs with requests for major projects. Now the SAs could contact them and schedule these large projects.

As a side benefit, the group was able to do a better job of asset management. Because all purchasing was done through one system, there was one place where serial numbers of new equipment were captured. Previous attempts at tracking assets had failed because it depended on such data to be collected by individuals who had other priorities.

Centralized purchasing's biggest benefit was the fact that the SAs now had knowledge of what was being purchased. They noticed certain products being purchased often and arranged volume purchasing deals for them. Certain software packages were preordered in bulk. Imagine the customers' surprise when they tried to purchase a particular package and instead received a note saying that their department would be billed for one-fiftieth of a 50-license package purchased earlier that year and were given a password they could use to download the software package and manuals. That certainly beat waiting for it to be delivered!

The most pervasive savings came from centralizing the PC purchasing process. Previously, customers had ordered their own PCs and spent days looking through catalogs, selecting each individual component to their particular needs. The result was that the PC repair center had to handle many types of motherboards, cards, and software drivers. Although a customer might take pride in saving the $10 by selecting a nonstandard video card, he or she would not appreciate the cost of a technician at the PC repair department spending half a day to get it working. With the repair group unable to stock such a wide variety of spare parts, the customers were extremely unhappy with having to wait weeks for replacement parts to arrive.

The average time for a PC to be delivered under the old system had been 6 weeks. It would take a week to determine what was to be ordered and push it through the purchasing process. The vendor would spend a couple of weeks building the PC to the specifications and delivering it. Finally, another week would pass before the SAs had time to load the OS, with possibly an additional week if there were difficulties. A company cannot be fast paced if every PC requires more than a month to be delivered. To make matters worse, new employees had to wait weeks before they had a PC. This was a morale killer and reflected badly on the company. The temporary solution was that management would beg the SAs to cobble together a PC out of spare parts to be used until the person's real PC was delivered. Thus, twice as much work was being done because two complete PC deliveries were required.

The centralized purchasing group was able to solve these problems. The group found that by standardizing the PC configuration, a volume discount could be used to reduce cost. In fact, the group was able to negotiate a good price even though it had negotiated four configurations: server, desktop, ultralight laptop, and ultrapowerful laptop. Fearing that people would still opt for a custom configuration, the group used some of the savings to ensure that the standard configuration would be more powerful, with better audio and video than any previously purchased custom PC. Even if the group matched the old price, the savings to the PC repair department would be considerable. The ability to stock spare parts would be a reduction in lost productivity by customers waiting for repairs.

The purchasing group realized that it wouldn't be able to push a standard on the customers, who could simply opt for a fully custom solution if the standard configuration wasn't to their liking. Therefore, purchasing made sure to pull people to their standard by making it amazingly good. With the volume discounts, the price was so low and quality so high that most of the remaining customization options would result in a less powerful machine for more money. How could anyone not choose the standard? Using pull instead of push is using the carrot, not the stick, get a mule to move forward.

One more benefit was achieved. Because the flow of new machines being purchased was relatively constant, the purchasing group was able to preorder batches of machines that would be preloaded with their OS by the SAs. New employees would have a top-notch PC installed on their desk the day before they arrived. They would be productive starting on the very first day.

Ordering time for PCs was reduced from 6 weeks to 6 minutes. When faced with the choice between ordering the exact PC they wanted and waiting 6 weeks or waiting 6 minutes and getting a PC that was often more powerful than they required, for less money, it was difficult to reject the offer.

Any company that is rapidly growing, purchasing a lot of computer-related items, or deploying PCs should consider these techniques. Other advice on rapid PC deployment can be found in Chapter 3. More information about how PC vendors price their product lines is in Section 4.1.3.

21.2.2 Outsourcing

Outsourcing is often a form of centralization. **Outsourcing** is a process by which an external company is paid to provide certain technical functions for a company. Some commonly outsourced tasks are running the corporate PC helpdesk, remote access, WAN and LAN services, and computer deployment operations. Some specific tasks, such as building the infrastructure to support a particular application—web site, e-commerce site, enterprise resource planning (ERP) system—are outsourced, though probably vendors refer to that process "professional services" instead.

The process of outsourcing usually involves centralization to reduce redundant services and to standardize processes. Outsourcing can save money by eliminating the political battles that were preventing such efficiencies. When executives are unable to overcome politics through good management, outsourcing can be a beneficial distraction.

Advocates emphasize that outsourcing lets a company focus on its core competency rather than on all the technological infrastructure required to support that core. Some companies become bogged down in supporting their infrastructure, to the detriment of their business goals. In that situation, outsourcing can be an appealing solution.

The key in outsourcing is to know what you want and to make sure that it is specified in the contract. The outsourcing company isn't required to do anything that isn't in the contract. Although the salespeople may paint an exciting picture, once the contract is signed, you should expect only what is specified in ink. This can be a particular problem when the outsourced services had been provided previously in-house.

We've seen three related problems with signing an outsourcing contract. Together, they create an interesting paradox. Outsourcing to gain new technical competence means that the people you are negotiating with have more technical competence than you do. This gives the outsourcing firm the power seat in the negotiations. Second, to accurately state your requirements in the contract, you must have a good understanding of your technical needs; however, if your executive management had a good handle on what was needed and was skilled at communicating this, you wouldn't need outsourcing. Finally, companies sometimes don't decide to outsource until their computing infrastructure has deteriorated to the point that outsourcing is being done as an emergency measure and are thus too rushed or desperate to keep the upper hand in negotiations. These companies don't know what they want or how to ask for it and are in too much of a panicked rush to do adequate research. As you can imagine, this spells trouble. It is worth noting that none of these situations that specifically allow for technology knowledge refer to the buying company.

You should research the outsourcing process, discuss the process with peers at other companies, and talk with customer references. Make sure that the contract specifies the entire life cycle of services—design, installation, maintenance and support, decommissioning, data integrity, and disaster recovery—SLAs penalties for not meeting performance metrics, and a process for adding and removing services from the contract. Negotiating an outsourcing contract is extremely difficult, requiring skills far more sophisticated than our introduction to negotiating (Section 32.2.1).

Some outsourcing contracts are priced below cost in order for the vendor to be considered a preferred bidder on project work; it's on this project work that outsourcing deals make money for the supplier. Contracts usually specify what is "in scope" of the contract and casually mention a standard rate for all other "out-of-scope" work. The standard rate is often very high, and the outsource organization hopes to identify as much out-of-scope work as possible. Clients don't usually think to send such work out to bid to receive a better rate.

There are outsourcing consultants who will lead you through the negotiating process. Be sure to retain one who has no financial ties to the outsourcing firms that you are considering.

Don’t Hide Negotiations

When one Fortune 500 company outsourced its computing support infrastructure, the executive management feared a large backlash by both the computing professionals within the company and the office workers being supported. Therefore, the deal was done quickly and without consulting the people who were providing the support. As a result, the company was missing key elements, such as data backups, quality metrics, and a clear service-level specification. The company had no way to renegotiate the contract without incurring severe penalties. When it added backups after the fact, the out-of-scope charges in the contract were huge.

Don’t negotiate an outsource contract in secret; get buy-in from the affected customers.

When you outsource anything, your job becomes quality assurance. Some people think that after outsourcing, the contract will simply take care of itself. In reality, you must now learn now to monitor SLAs to make sure that you get all the value out of the contract. It is common for items such as documentation or service/network architecture diagrams to be specified on the contract, but not delivered until explicitly requested.

Critically Examine Metrics

Executives at one company were very proud of their decision to outsource when, after a year of service, the metrics indicated that calls to the helpdesk were completed, on average, in 5 minutes. This sounded good, but why were employees still complaining about the service they received? Someone thought to ask how this statistic could be true when so many calls included sending a technician to the person’s desk. A moderate percentage of calls like that would destroy such an excellent average. It turned out that the desk-side support technicians had their own queue of requests, which had their own time-to-completion metrics. A call to the helpdesk was considered closed when a ticket was passed on to the desk-side technician’s queue, thus artificially improving the helpdesk’s metrics. Always ask for a detailed explanation of any metrics you receive from a vendor, so that you can clearly relate them to SLAs.

While you are trying to get the most out of your contract, the outsourcing company is trying to do the same. If the contract is for “up to $5 million over 5 years,” you can be assured that the account executive is not going to let you spend only $4.5 million. Most outsourcing companies hold weekly meetings to determine whether they are on schedule for using the entire value of the contract as quickly as possible; they penalize their sales team for coming in "under contract." Does the contract charge $1,000 per server per month? "How can we convince them that a new service they've requested needs a dedicated host rather than loading it onto an existing machine?" will be asked at every turn. Here's the best part: If they can get you to spend all $5 million of a 5-year contract in only 4.5 years, those last 6 months usually won't be at the discounted rate you negotiated. How can anyone predict what their IT needs will be that far out? This is the most dangerous aspect of long-term outsourcing contracts.

Make sure that your contract specifies an exit strategy. When starting a long-term contract, the outsourcing company usually retains the right to hire your current computing staff. However, the contract never says that you get them back if you decide that outsourcing isn't for you. Many contracts fail to guarantee that your former staff will remain on-site for the duration of the contract. The company may decide to use their skills at another site! Even switching to a different outsourcing company is difficult, because the old company certainly isn't going to hand over its employees to the competition. Make sure that the contract specifies what will happen in these situations so that you do not get trapped. Switching back to in-house service is extremely difficult. Eliminate any noncompete clauses that would prevent you from hiring back people.

Our coverage of outsourcing is admittedly centric to our experiences as SAs. Many books give other points of view. Some are general books about outsourcing (Gay and Essinger 2000, Rothery and Robertson 1995); by contrast, Williams (1998) gives a CIO's view of the process. Mylott (1995) discusses the outsourcing process with a focus on managing the transfer of MIS duties. Group Staff Outsource (1996) has a general overview of outsourcing. Kuong (2000) discusses the specific issue of provisioning outsourced web application service provider services. Jennings and Passaro (1999) is an interesting read if you want to go into the outsourcing business yourself. Finally, Chapman and Andrade (1997) discuss how to get out of an outsourcing contract and offer an excellent sample of outsourcing horror stories. We pick up the topic of outsourcing again in Section 30.1.8.

The first edition of this book was written during the outsourcing craze of the late 1990s. We had numerous warnings about the negative prospects of outsourcing, many of which came true. Now the craze is over, but **off-shoring** is the new craze. Everything old is new again.

21.3 Conclusion

Centralization and decentralization are complicated topics. Neither is always the right solution. Technical issues, such as server administration, as well as nontechnical issues, such as organizational structure, can be centralized or decentralized.

Both topics are about making changes. When making such pervasive changes, we recommend that you consider these guiding principles: know what specific problem you are solving; understand your motivation for making the change; centralize as much as makes sense for today; recognize that as in rolling out any new service, it requires careful planning; and, most important, listen to the customers.

It is useful to learn from other people's experiences. The USENIX LISA conference has published many case studies (Epp and Baines 1992; Ondishko 1989; Schafer 1992b; and Schwartz, Cottrell, and Dart 1994). Harlander (1994) and Miller and Morris (1996) describe useful tools and the lessons learned from using them.

Centralizing purchasing can be an excellent way to control costs, and our example showed that it can be done not by preventing people from getting what they want, but by helping them make purchases in a more cost-effective manner.

We ended with a discussion of outsourcing. Outsourcing can be a major force for centralization and will be a large part of system administration for a very long time, even under different names.

❖ **Centralization Rules of Thumb** Every site is different, but we have found that, as an informal rule of thumb, centralization of the following services is preferred once a company grows large enough to have multiple divisions:

- Network security
- Internet connectivity
- WAN services
- LAN deployment and operations
- Email services
- ActiveDirectory/LDAP
- Desktop deployment and life-cycle management

- Storage within a data center
- Web services with external access
- IP address allocation and DNS management

Exercises

1. How centralized or decentralized is your current environment? Give examples.
2. Give an example of a service or an aspect of your organization that should be centralized. Relate guiding principles in Section 21.1.1 to such a project.
3. Give an example of a service or an aspect of your organization that should be decentralized. Relate guiding principles in Section 21.1.1 to such a project.
4. In Section 21.1.3, we describe decentralizing email servers to achieve better reliability. How would you construct a similar architecture for print servers?
5. Describe a small centralization project that would improve your current system.
6. Share your favorite outsourcing horror story.