

# 3

## Crafting Your mySAP Solution Vision

### What Is a Solution Vision?

With the understanding you have gained in terms of what an SAP project looks like, we are ready to begin refining and later communicating a vision of the future-state of your SAP solution, a *solution vision*. Think of this as the “eyes closed” phase of the project, where wishful thinking is tempered by constraints like budget and headcount realities to sketch a design that meets both business and financial requirements, as you see in Figure 3.1.

In the narrow role I have personally played helping my customers to craft such a vision, I give the following advice to executive and senior IT decision makers, and other members of the SAP project steering committee:

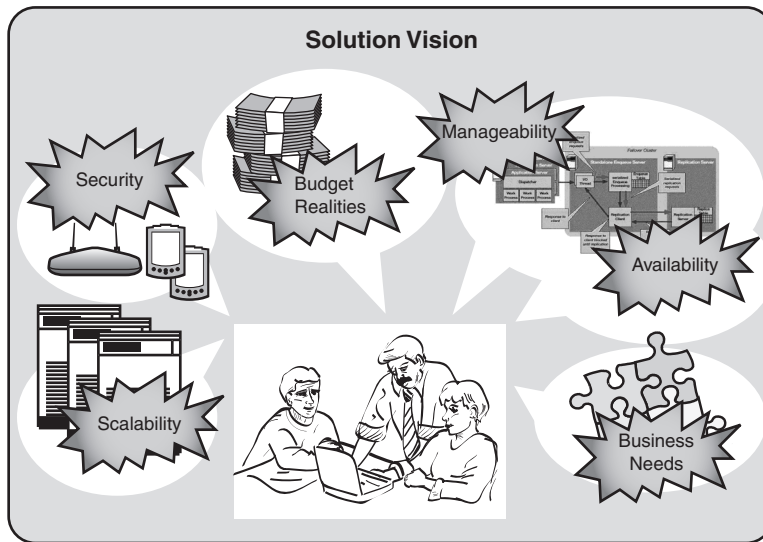
- Generally, focus on your core business and how an enterprise solution will better enable that core business to be successful.
- Identify the shortcomings of the systems and processes in place today. For example, is it difficult, expensive, or cumbersome to customize the system? Are employees forced to duplicate entries in multiple systems, or access different systems for different customers? Is the system subject to downtime because of hardware and other solution stack issues?
- Clearly define the value that you believe those systems should provide to the business. That is, should the system be available 24x7? Should it be accessible over the Internet or your company

### IN THIS CHAPTER

- What Is a Solution Vision?
- SAP System Landscape Design and Planning Approaches
- Capturing Your mySAP Solution Vision
- Outsourcing—Another Method of Achieving Your Solution Vision
- Tools and Techniques

intranet? Should it tie together different functional and business areas, or enable real-time decision making?

- With the data collected in response to the questions in the previous two bullet points and with your real business requirements nailed down in Chapter 2, step outside of the “box” and consider alternatives and “nice-to-haves.” Enlist the assistance of your own long-time end users and the insight of your current IT staff to begin assembling a new solution vision that describes what the system *should* be capable of.
- Solicit the advice of mySAP.com experts to assist you in identifying real-world product and technology constraints, thereby helping you to refine and document the *characteristics and capabilities* of a mySAP solution that can be customized and implemented for *your* company in support of *your* business objectives.



**FIGURE 3.1** The solution vision melds company needs and constraints into an achievable business vision facilitated by technology.

Regarding this last point, mySAP.com experts can be enlisted from many places. I suggest creating a focus team of pre-sales consultants from SAP AG itself, from a Big 4 or other capable implementation partner, and from an enterprise hardware and services partner. In this way, most of the SAP Solution Stack is represented while still minimizing the number of players.

After the vision is initially captured, disseminate it in draft form so as to begin a *review process* of sorts—share it with stakeholders, like senior members of the business groups who will use the solution in their day-to-day dealings, the customer-facing groups who will be positioned to better serve your customers, and the Information Technology professionals who will be tasked with supporting the solution. Formally gather and document all of this feedback, so as to better revise and refine your solution vision. To this end, I recommend updating the Solutions Characteristics Matrix first discussed in Chapter 2. Ensure that senior and executive-level management concur with the vision as it evolves, and that buy-in is achieved at all levels of your organization. Only after all of this is accomplished can the real work of planning how to actually “get there” commence.

### Impact on the Business

As the business groups begin sharing their thoughts and insight regarding the solution vision, keep in mind the following:

- Business processes will almost certainly have changed with the introduction of your new mySAP system, typically reflecting tighter integration and best practices.
- Therefore, employee roles will change, and jobs may potentially be at stake.
- Finally, the tools and interfaces used by each employee in the normal course of their job will change.

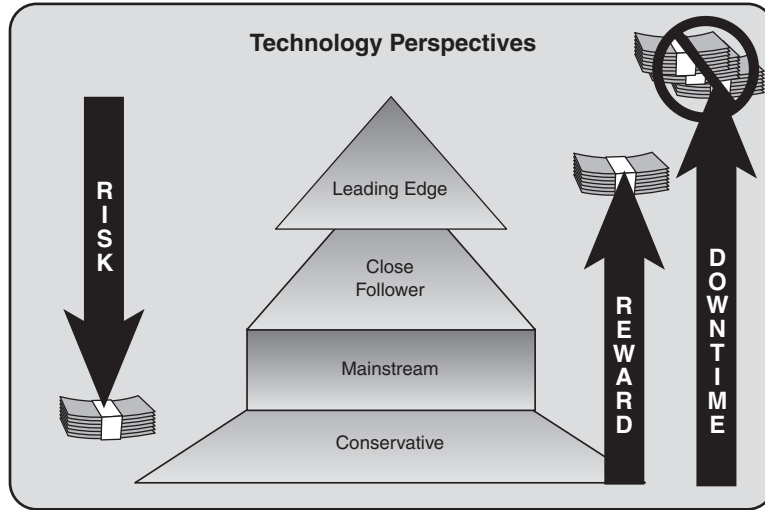
Back to the solution review process—as different folks inevitably demonstrate resistance to the project, consider the points in the preceding list, especially whether their jobs are impacted and to what degree. And just as importantly, consider each individual’s personal resistance to change. These two factors represent the key rationale behind exposing only senior members of the different business and functional groups to the new solution vision. With your senior and loyal employees on board and embracing potential changes as their own, you will be positioned as best you can against pockets of resistance lower in the organization.

### Technology Perspective

Before specific software packages and hardware components are purchased, or services contracts are signed, a company must come to grips with its *technology perspective*, which is simply how it views its investment in information technology resources. Why? Because this shapes the architecture, or the very foundation, of a computing solution. Some companies look at IT spending from a long-term perspective, and try to purchase assets with a useful life of perhaps many years. Other companies subscribe to the belief that regular hardware and software refreshes will provide a competitive advantage or a performance advantage over time. Still others

seek to stay on one side of the spectrum or the other, investing conservatively in time-proven solution stack components, or on the other hand investing in the latest and greatest high-availability and performance offerings. And finally, others prefer to outsource technology and its requisite support structure. I like to understand how a company thinks in this regard before attempting to architect a mySAP hardware and software solution; it is important for everyone to understand how risk tends to increase as investments in new technology increase, too, promising greater potential reward in exchange. This is illustrated in Figure 3.2 and detailed in the following different technology perspectives:

- **Conservative**—As the least risky of all approaches, a company that has a conservative technology perspective places availability above all else. They seek mature technology, mature practices, and tried-and-true solutions that *work*, day in and day out. What they potentially sacrifice, then (though in their eyes this is not a sacrifice at all), is anything new—new approaches to accessing their system, new methods of improving availability or manageability, new solution architectures, and so on.
- **Mainstream**—Like their conservative brethren, these companies prefer established platforms and products to newer ones. However, the key word here is “company”—they want to have a lot of company when it comes to how they solve their business problems through the use of IT resources. Mainstream companies want to be able to point to a slew of other companies and feel confident that they are not alone, that most of the industry is doing things in a manner similar to theirs.
- **Close follower**—My favorite companies to work with tend to be close followers. They seem to leverage their IT investments in proven technology, but with exceptions. That is, although maximizing uptime always remains central, close followers are unafraid to try a few new things to gain a competitive advantage or otherwise position themselves better for the future. Therefore, they take an occasional calculated risk and invest in new products, new technologies, and new approaches.
- **Leading edge**—This is the riskiest of all approaches, hence the more popular label “bleeding edge” assigned to this technology perspective. A leading-edge approach places more value on competitive positioning than anything else—it’s all about getting a jump on the competition in terms of minimizing cost, reducing downtime (through recent technology advances), increasing response times of customer-driven business transactions, maximizing accessibility (for example, through Internet-based vendor/partner access to your order-status system), and so on. Therefore, a leading-edge company must be prepared to spend much more time managing change, as they tend to introduce new products and approaches without the benefit of a “history.” In fact, because of this, leading-edge companies are the same ones that tend to find and work through technology problems first.



**FIGURE 3.2** These four key technology perspectives illustrate how greater risks are related to potential reward as well as increased downtime.

When your technology perspective is clear, we can start looking at individual solution components and how all this fits together to create a custom system landscape for each particular solution. Note that I did not include outsourcing in the preceding list—the topic of outsourcing as a technology perspective is covered in the last section of this chapter.

### Considering mySAP Components to Be Implemented

As the different business requirements are hammered out and in turn mapped to the solution vision, inevitably a discussion around various mySAP offerings emerges. Take care to distinguish between *current* mySAP component capabilities, and new features that will soon be released in new versions of a particular mySAP solution. Over the last few years, SAP has aggressively released new versions of current mySAP components, re-badged existing components and technologies, and added quite a few *new* components. So as you begin discussing specific solutions like Advanced Planner and Optimizer, SRM E-Procurement, or Enterprise Portal and the SAP Exchange Infrastructure, it is very important to bring in an expert versed in both the solution's current capabilities and shortcomings, and what lies ahead on the road map.

### Considering SAP System Landscape Requirements

As with the mySAP components to be implemented, it's also important to determine the SAP system landscape requirements necessary to achieve your solution vision:

- Do you need a formal training system for end users?
- What about your IT staff—will a Technical Sandbox be required to help them gain a certain comfort level with new technology?
- Will your functional and development/programming team need a Business Sandbox with which to learn and test?
- Will a dedicated load-testing system need to be maintained that is identical to the Production system?

All these questions must be answered soon. This is why figuring out details related to your SAP system landscape plays such a big part in this chapter. In essence, though, evaluating the following will help you answer SAP landscape-specific questions as we move forward:

- The relative strength or weakness of an organization often determines whether an SAP system landscape component is warranted. For example, a “weak” IT team—a team uneducated or unfamiliar with a particular technology platform—will benefit greatly from a Technical Sandbox. Similarly, a development team less than familiar with a unique mySAP component/development tool combination will require a Business Sandbox.
- High availability drives much of the SAP system landscape design, too. The original “SAP 3-System Landscape” discussed in many books and articles over the years evolved out of the need for improved availability, for instance. But your particular needs may drive the creation of a more robust architecture where additional testing is possible.
- The ability to recover quickly from a disaster drives the creation of a Disaster Recovery system. The term “quickly” is relative of course, but a backup tape-based restore performed on a newly installed hardware platform usually represents a worst-case baseline.
- If performance is critical, adding a Staging system to a Development/Test/Production landscape can provide the resources necessary for load-testing or stress-testing changes prior to implementing them in Production (or prior to a change management package or “wave” being promoted to Production).
- If the idea of *simplification* is important to you, there are strategies and approaches designed to do just that—simplify your SAP system landscape.

Other factors like critical security concerns, the ability to manage a particular solution, and so on will drive the adoption of incremental systems, too. All these factors and characteristics are discussed in detail in the next section.

## SAP System Landscape Design and Planning Approaches

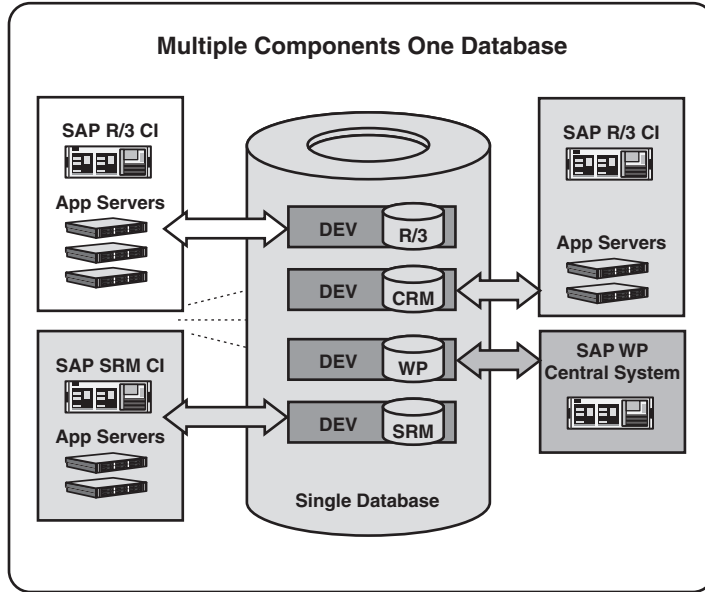
Remember, a system landscape exists for each mySAP solution—if you deploy R/3, APO, CRM, and PLM, you will in effect be creating four different SAP system landscapes, one for each product. The focus in this and the following few sections, though, is on what *one* of these system landscapes looks like from a design and planning perspective, for example R/3 alone or APO alone.

In the most general form, an SAP system landscape consists of SAP instances (installations of the SAP database and application software) and SAP servers. In the Microsoft world of SAP implementations, there is a one-to-one correlation between instances and servers nearly all the time. That is, the Development instance resides on a dedicated Development server, the Test instance resides on a dedicated Test server, and so on. In the world of UNIX implementations, though, multiple instances can be often found on a single “larger” server. For example, both Development and Test instances can reside on a single server. And multiple application instances can be installed on a single server as well.

Until last year’s release of SAP’s *Multiple Components, One Database* (MCOD) initiative, there was a one-to-one correlation between instances and database systems, too, regardless of the Operating System platform. MCOD is beginning to change this, such that a single “larger” database can be leveraged for multiple instances. However, an important difference between MCOD and multiple instances/one server exists—MCOD ties the same type of databases within *different* SAP system landscapes together. With MCOD, all Development databases used by your R/3, SRM, CRM, and Workplace implementations can be one and the same. Similarly, all Test databases across R/3, SRM, CRM, and Workplace can be bundled together, too, as illustrated in Figure 3.3. Note, however, that in many cases SAP AG frowns on mixing OLTP and OLAP systems, or combining different databases within the same system landscape. In that regard, forcing an MCOD database server to host your R/3 system’s Development and Test databases would therefore be unsupported and contrary to best practices.

As we move forward with our basic understanding of SAP system landscapes, and seek to understand how your SAP solution vision impacts and is impacted by your landscape decisions, my hope is to achieve the following:

- Note the relative importance and relationship of technology perspectives to our solution vision
- Understand why each system landscape is important to fulfilling our vision
- Note how the presence or absence of a particular system within a landscape impacts the other systems and ultimately the overall solution vision



**FIGURE 3.3** A properly architected sample MCOD deployment is displayed for a typical mySAP enterprise consisting of R/3, CRM, SRM, and Workplace.

All these design and planning approaches I cover tend to come into play in one manner or another across all mySAP implementations. It's how they are weighted or addressed that makes one system landscape different from the next.

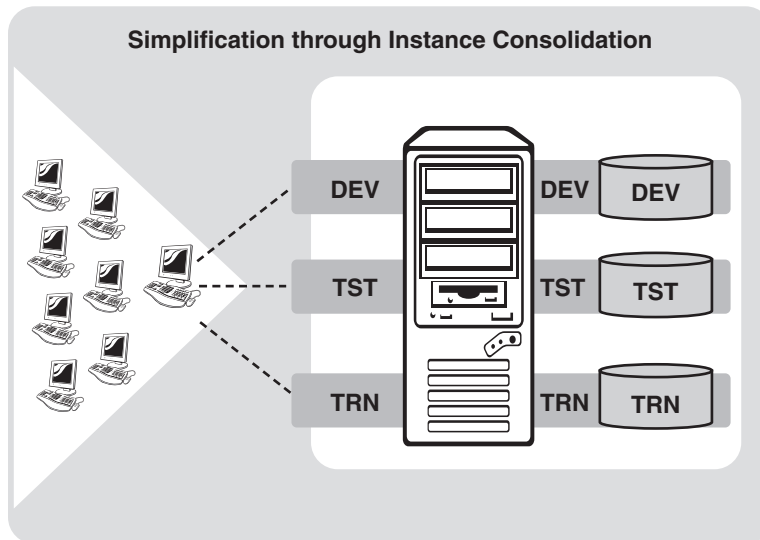
### Simplifying Your SAP System Landscape

After spending time with hundreds of customers and SAP implementations, I think it is safe to say that when all things are equal, the desire to *simplify* emerges as an important driver. Simplification takes many forms, too. In the case of the SAP system landscape and how it fulfills our SAP solution vision, the desire to simplify manifests itself in any number of ways:

- First, the pure number of instances will be reduced to the fewest necessary to get the job done “right” for a particular company. An organization focused on simplifying administrative, change management, systems management, operations, and other tasks will deploy a three-system or even a two-system landscape, whereas similar organizations without the same simplification goals can deploy more. There are trade-offs, of course. A system landscape without a dedicated test instance will, for example, be forced to perform testing in the same system used for development. Because of these kinds of limitations, simplification achieved through instance reductions is not as common as it has been in the past.



- Instead, a more popular approach to simplification seeks to reduce the number of physical servers in a particular system landscape, by installing multiple instances on a single server. Consolidation of instances is becoming quite common in SAP customer environments today, as displayed in Figure 3.4.
- Similarly, deploying a shared disk subsystem and tape backup/restore solution also simplifies a very complex piece of the SAP Solution Stack. This is why my colleagues and I have spent so much time in the last two years designing and implementing Storage Area Networks, or SANs—they provide outstanding performance while simultaneously reducing system landscape complexity and allowing expensive resources like enterprise tape libraries to be shared between systems.
- Another customer of mine shared with me why they went with the WebGUI as opposed to the classic SAPGUI approach to system accessibility—to simplify desktop support and maintenance requirements.
- Companies that value simplification will also standardize on a particular solution stack option or approach, too, as this simplifies support and maintenance, and minimizes the need for a variety of onsite/reserved spare parts, the time spent in change management activities, and more.



**FIGURE 3.4** Multiple SAP instances can be installed and configured on a single physical server, oftentimes reducing both acquisition and systems management costs down the road.

Although simplification tends to work in one direction by encouraging a “do more with less” philosophy, our next topic goes the other route in that it purposefully introduces complexity and differences between various systems within a system landscape—high availability.

## High Availability and the SAP System Landscape

When it comes to high availability, many technology professionals automatically think about what it means to improve the availability of a particular system or hardware component—thoughts of basic HA offerings like clustering or redundancy come to mind. With regard to the broader topic of how your solution vision impacts your SAP system landscape, though, high availability equates to the following:

- Business-driven requirements—HA offerings and approaches are normally implemented to satisfy specific business-oriented needs, and therefore form an integral part of your overall SAP solution vision.
- Complexity—HA complicates the SAP system landscape, as HA offerings and approaches tend to only really exist or apply to the Production system and at minimum (hopefully!) another similarly configured system within the landscape.
- Increased support needs—Because HA offerings are inherently complex, a very real need exists to prepare your SAP support organization in how to install it, update it, manage it, and troubleshoot HA issues.

► To read more about how business requirements relate to high availability, see “Availability Planning—Documenting Requirements and Key Drivers,” p. 167 in Chapter 6.

## Disaster Recovery Considerations

All companies implement a method of addressing Disaster Recovery (DR), whether or not they actually realize it. Even companies that do not add a dedicated DR system to their system landscape address disaster recovery. That is, their de facto disaster recovery plan simply reflects the challenges and timeframes surrounding rebuilding their SAP system from scratch, restoring from their latest tape backup, and imposing upon their end users to manually rekey any new business transactions lost between the last successful tape backup and the point at which the disaster occurred. This doesn’t sound like much of a “plan,” of course, but it does represent a baseline against which all other disaster recovery approaches and solutions can be weighed.

A host of DR approaches are discussed throughout Chapter 6, from those involving disk subsystem data replication solutions, to various clustering solutions, to database

and mySAP-specific tactics. But when it comes to sifting the potential layout of your SAP system landscape through your solution vision, two general approaches fall out:

- Creating and maintaining a dedicated Disaster Recovery system within your overall system landscape.
- Outsourcing your Disaster Recovery system to an outsourcing provider.

Both approaches are valid, and the first is more traditional. But I believe that the time and expense related to setting up, configuring, keeping current, and managing your own DR system explains the recent increase in outsourcing I've seen over the last two years.

- ▶ To review some of the tasks and considerations inherent to addressing DR internally as opposed to outsourcing it, see “SAP General Availability and DR Best Practices,” p. 207 in Chapter 6.

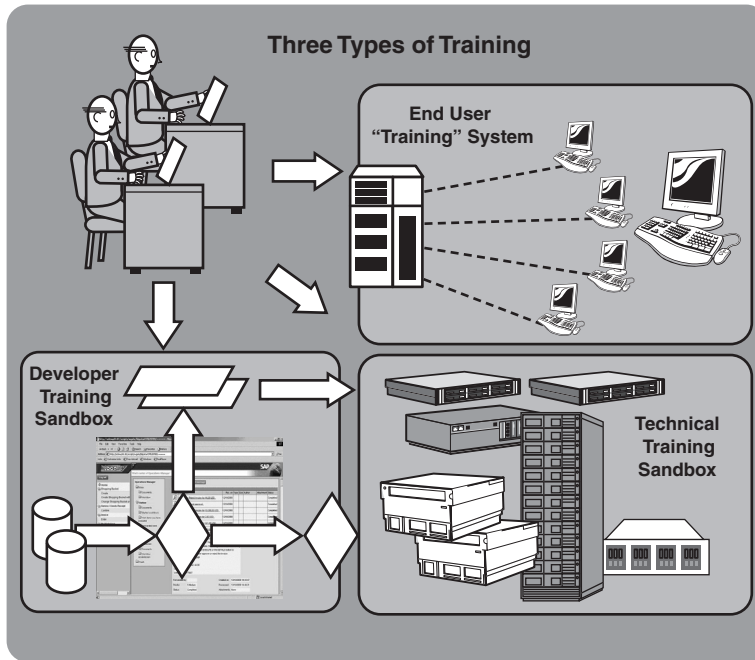
Companies that outsource the DR component of their SAP system landscape help to preserve their data, and access to this data, in that the outsourcer operates a completely independent data center, typically in a very different geographic location. For smaller and mid-size companies with only a single data center, the expense relief is tremendous. On the other hand, if the DR solution is maintained “in-house,” so to speak, it will need to be housed in a separate facility. This alone is sure to drive complexity, cost, and even the architecture and makeup of both the SAP system landscape and its individual systems.

## Addressing Training Requirements

The SAP system landscape is directly impacted by the potential need to train SAP end users as well as the system's developers and technical support staff. Three different systems come into play, as illustrated in Figure 3.5:

- A dedicated *Training* system is often implemented to assist in teaching users new to a particular mySAP component *how* to actually use the system. This amounts to business-process training as well as SAP user interface training (an excellent alternative to creating multiple training clients on the Test system, which is busy fulfilling integration responsibilities prior to Go-Live—the exact time when end users *need* to be trained!). To provide the most value to its students, the Training system needs to be an exact copy of Production.
- A dedicated *Technical Sandbox* system is extremely useful in helping the SAP Technical Support Organization (SAP TSO) get up to speed on the entire SAP Solution Stack, especially with regard to new components and complex HA offerings (rather than attempting to get time on other systems for what could amount to *crash-and-burn* testing).

- A dedicated *Business Sandbox* or *Development Sandbox* system allows developers unfamiliar with a particular mySAP component, or faced with integrating multiple components and other legacy systems, the opportunity to do so in a pure testing environment (rather than the real Development system).



**FIGURE 3.5** These SAP training systems support the different needs of different organizations, from end users, to developers and programmers, to technical implementation specialists.

- For details as to how the SAP system landscape satisfies the training needs of both the SAP Technical Support Organization and the production system’s end users, see “Training and the Role of the SAP System Landscape,” p. 314 in Chapter 9.

It can represent quite a challenge for the “customers” of one of these training systems to convince everyone that such a system is truly required. In my own experience, I have seen the lack of a Technical Sandbox really hurt an organization in terms of downtime due to botched infrastructure upgrades and changes to DR processes.

Another colleague of mine has more than once had to strongly push for the adoption of a Training system, too. Such a system allows for extensive informal user testing and practice outside of formally delivered training. He believes that this extra

level of hands-on self-directed training is critical because your end-user community is best positioned of *all* groups to find business-process operational errors and limitations. And of course it is desirable to correct these issues well before Go-Live. But a consultant or even a senior super user is typically not positioned to push the adoption and use of a dedicated Training system. More often than not, it takes the SAP Steering Committee, the project's experienced management team, and the prodding of a knowledgeable SAP Solution Architect to do so. I cannot stress this enough—the risk is huge, in that you do not want to find out too late that not every business scenario works as it did during integration testing (for example, all types of contracts, all types of material movements, all kinds of accounting entries, and so on).

## The Performance-Driven System Landscape

When it comes to evaluating your solution vision against the layout of your SAP system landscape, it is important to ensure that the performance of the systems meets the needs of their different end-user communities. Most of the time, of course, the focus is on designing, installing, and configuring a well-performing Production system. Performance considerations usually relate back to what an end user will experience while on the system, including

- Business transaction *response times*, or how long it takes to refresh your SAPGUI after pressing the Enter key, for example.
  - How quickly a background or “batch” job will execute, otherwise known as *throughput*.
  - How quickly a report or other query will make it through the system and actually be printed, sometimes called *latency*.
- To read more about verifying that a Production system can meet performance expectations, see “Key SAP Stress-Testing Considerations,” p. 580 in Chapter 16.

However, these same performance considerations apply to all of the other systems within the SAP system landscape, too. The Development system, for example, needs to exhibit excellent performance even while 25 or 50 or more developers are banging away at keyboards trying to build your custom mySAP solution. Similarly, your Test system needs to provide the performance necessary to get through integration testing. Even the Training system needs to provide adequate user response times so as to make the actual training experience more than something to be avoided.

High-performance considerations cover the gamut, touching every facet of every system within the landscape. This means that *everything*—from the performance of the network connecting each system, to each server's CPU, RAM, and disk configuration, to each system's OS, database, and mySAP component—must be addressed.

Starting off on the right foot (with properly sized and configured hardware and software elements) is paramount, of course, but tuning all these solution stack pieces to create a cohesive well-running machine is just as important to achieving excellent performance. Like the weakest link in a chain, a single underperforming solution component will only throttle back the maximum performance otherwise obtainable from your system.

## Driving Scalability into Your System Landscape

The need for scalability, like high availability and excellent performance, is addressed primarily through the sizing process. Scalability does not pay off up front in terms of improved system availability or better user response times, though. Rather, scalability is all about paying for “headroom” in your system, headroom that is not actually needed at present but might be required in the near future. In other words, scalability addresses future planned and unplanned growth in your system.

This growth can manifest itself in a number of ways. In my experience in the real world, I have seen the results of *unplanned* growth hurt companies where scalability was never addressed, as in the following cases:

- The number of end users increased at one of my new accounts, not due to more hiring than was anticipated when their mySAP.com solution was crafted, but because they unexpectedly acquired their competitor and doubled in size. We had six months to project the delta needed in terms of database and application server processing power and RAM requirements, followed by stress-testing the new design and finally implementing it.
- More than one of my customers' databases grew so fast that they outstripped the results of their comprehensive three-year database sizing methodology in the first year! In most cases, the system we put in place for these customers was scalable—more disk drives could be added, smaller drives could be swapped with larger ones, and so on. In three cases in particular, though, the database growth was so explosive that a whole new disk subsystem platform needed to be brought in, and the recently acquired current platform retired (or redeployed) years earlier than expected.
- When databases grow quickly, the tape backup/restore solution implemented often grows less effective as well. I have seen this most often in relatively small SAP implementations, where an initial investment in tape backup technology needed to be tossed in favor of tape solutions that backed up more data per tape cartridge, and did so fast enough to not exceed the customer's backup window (time allotted to perform a backup, which usually equates to planned downtime in the case of offline full backups).

- It's been a while, but I also had a customer outgrow their network, too. Today, with switched networks and Gigabit Ethernet providing more than adequate bandwidth to every mySAP.com server component, and cheap 10- and 100-Megabit Ethernet prevalent across end-user workstations, there's no excuse for lacking network scalability.

Outstripping the capabilities of your current system such that a new platform is needed probably represents a worst-case scenario. Not only does the current Production component need to be replaced, but to support sound change control principles, so does the same component in your Test, Staging, and/or Technical Sandbox environments.

This is why hardware and software vendors tout things like “highly scalable system architectures,” “enterprise versions” of particular Operating Systems and Database Systems, and so on—though not necessarily needed up front, the headroom that these approaches provide helps an organization feel more comfortable if they wind up growing faster than they expected. And hardware vendors in particular can position their SAP clients for improved scalability by practicing the following:

- Specify server platforms that allow additional CPUs and RAM to be added as needed. In other words, avoid “maxing out” the box.
- Alternatively, design SAP solutions such that they take advantage of SAP's support for *horizontal scalability*. This is one of my favorite approaches when it comes to SAP Application, Web AS, J2EE middleware, and ITS servers—I prefer to max them out with regard to processors, with the understanding that an incremental number of servers can be added at any time should the environment grow to require it (interestingly, although SAP has successfully tested a system running more than 160 application servers, it is rare to find customer implementations with more than 10 or 12).
- Architect a solution for the appropriate level of vertical scalability. In other words, if a two-tier “Central System” (where all SAP software components execute on the same physical server) approach to sizing meets today's requirements, perhaps a three-tier solution will provide for unknown scalability requirements. In a three-tiered architecture, one database server and multiple application servers are configured as a single system image.
- Architect a highly scalable database platform. As my real-world examples earlier in this list illustrate, this tends to be where a lack of scalability causes the most problems.

Hardware and software vendors alike spend a great deal of time “proving” how scalable their offerings are. As a first step, I suggest that benchmarks, customer references/feedback, and the results of tests published through white papers and other

technical documents be reviewed by prospective mySAP customers. I also suggest that you begin considering new approaches to scalability. For example, HP's iCOD offering touts "capacity on demand." When a customer buys a server, for instance, it is fully populated with CPUs. The customer pays for only what is needed in the near term, however. Later, if it is determined that more processing power is required, the customer takes advantage of the in-place processors by merely applying for a license; no intrusive field upgrade or service call is required and therefore the need for planned downtime is drastically reduced.

### The TCO-Driven System Landscape

More than anything else, Total Cost of Ownership (TCO) drives what a solution vision actually looks like at the end of the day, when a mySAP solution is implemented and really being used. Discussions on TCO might instead be labeled Return on Investment (ROI), or might fall under the heading of investment protection. Regardless, a focus on lowering TCO seeks to find less expensive solution-stack alternatives that still meet the needs of the business.

- ▶ To read more about the relationship between TCO and your SAP solution vision, see "How the SAP Solution Vision Drives TCO," p. 127 in Chapter 5.

When all other things are equal, the following points apply from a hardware perspective:

- A hardware vendor's use of common components like CPUs and memory boards allows flexible sharing of resources between different SAP system landscapes and in some cases hardware platforms, too.
- Similarly, common disk drive form factors reduce cost of ownership by increasing reusability.
- Support for hot-pluggable and/or hot-add hardware components eliminates or worst-case minimizes downtime (can include hard drives, tape drives, power supplies, fans, and even RAM and processors).
- Support for redundant components, like power supplies, disk drives, fans, and so on, also eliminates or minimizes downtime.
- The ability to run mixed-speed CPUs or RAM in a particular platform protects that investment—CPUs and RAM do not have to be tossed aside when additional processing power or memory is required.

Outsourcing your entire SAP infrastructure/operations team is another potential method of reducing TCO. In fact, outsourcing can represent the biggest potential TCO factor that a company will consider. At this level, though, outsourcing becomes more of a strategic business solution that impacts a lot more than simply TCO. True,



outsourcing can cut labor costs by 50%, and enhance flexibility of a technical support organization to easily change as business requirements change, but there are drawbacks and disadvantages as well (discussed later in this chapter).

Another solution vision approach that impacts the SAP system landscape from both a configuration and TCO perspective is the use of an *Application Service Provider (ASP)*. ASPs can drive lower TCO by virtue of their application-specific expertise, above and beyond that provided by in-house staff and traditional outsourcing providers. For example:

- An ASP can offer a preconfigured solution stack for the particular mySAP solution you want to implement. This is one reason why they look so good from a TCO perspective—design, deployment, manageability, operations, and other cost factors are substantially reduced due to a high level of both standardization and core competencies in the services they provide.
- ASPs were more or less born out of the dot-com era, and by virtue of this, their data centers enjoy the benefit of fat redundant pipes to the Internet. Thus, mySAP.com applications are well positioned to take advantage of this flexible and powerful accessibility option.
- ASPs offer interesting financing alternatives, in that they partner with various SAP technology partners to make leasing, pay-as-you-go, and other payment methods available.

The ASP provider market shrank over the last few years. The mySAP-focused companies that weathered these hard times seem even better prepared and well-positioned to host SAP solutions, however.

## Security Considerations

I know of *no* company that does not envision protecting its corporate computing assets. From a solution-stack security perspective, not all software vendors are created equal, however. Oracle touts its unbreakable database, UNIX vendors tout the robust security features of their operating environments, and so on. In my eyes, security features are very important, but good security is more often about managing and testing changes to a solution stack, by carefully identifying security holes and other weaknesses in new solution stack components before these components ever find themselves in Production.

However, companies that embrace and act upon the idea of protecting computing assets will prove to be better partners in the long run. This is why I believe that Oracle's focus on security will pay big dividends in terms of slowing the adoption of competing databases. And it is why I believe that the *Trusted PC* joint Intel, AMD, and Microsoft vision (once labeled by Microsoft as Palladium, and now referred to as

the “next-generation secure computing base for Windows”) will prove fruitful as well. Its goal is to build security into servers and PCs at a microprocessor level. New initiatives coming out of the Trusted Computing Platform Alliance promise to better secure our processing platforms, ensuring that only authorized applications and program executables can ever be executed by the system, and that all data housed on the system is encrypted so that it is useless to others. To this end, Microsoft considers Trusted PC a significant part of its Trustworthy Computing strategy—we should see something commercially available in this regard by 2004 that applies to server as well as desktop and other computing platforms.

### Manageability Considerations

No customers of mine have ever started their initial SAP implementation planning discussions with me by saying:

“George, all that high-availability and performance stuff is fine, but what we *really* want is a nice *manageable* system. Can you do that for us?”

By the time Go-Live looms just over the horizon, though, every single one of them—without exception—has indicated a growing concern for manageability. Sure, it’s there on the project plan, and any number of products can be used to support managing your mySAP environment. But the whole field of manageability is more complex *and more work* than you would imagine. Consider the following:

- Each layer in the SAP Solution Stack must be managed; the risk of not keeping an eye on a particular layer or solution component affects the uptime of the entire system.
  - Because each layer is so different from the others, it’s nearly impossible to find a single management product that can actually monitor and report on more than a few layers, much less the entire stack.
  - Therefore, the next best thing becomes trying to find a product that can at least interoperate successfully with other products.
  - At the end of the day, three, four, or even more tools and utilities must ultimately be fused together to provide a holistic view of a mySAP solution stack. This is challenging, to say the least!
- To learn exactly how challenging piecing together a management approach can be, see “Systems Management Techniques for SAP,” p. 511 in Chapter 14.

Because of the challenges inherent to managing hardware and software products from a lot of different solution stack vendors, some of my customers have purposely chosen less than “best-of-breed” products for their SAP solutions, so as to minimize

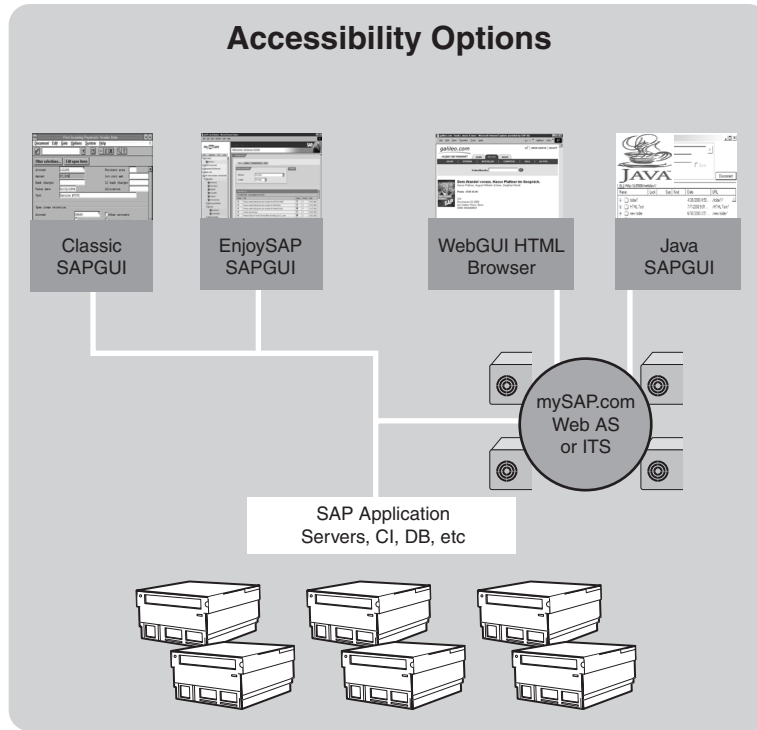
the number of software partners involved. Or they have decided to reduce the number of partners and vendors altogether by selecting one of the big enterprise hardware/services vendors. The obvious partners are clear—HP, IBM, and Sun. For example, if you go with HP and choose to implement an rp8400-based server platform with an HP StorageWorks SAN, running HP-UX 11i, and managed by HP OpenView, the challenges inherent to managing four different vendors' products just dropped tremendously. Similar arguments could be made for going with an IBM or Sun solution stack, too—IBM even throws a couple of databases into the mix.

## The System Landscape and Accessibility

The last area I want to cover with regard to solution vision and the SAP system landscape is accessibility. Many companies over the last three or four years have started with a vision of dumping all application-specific interfaces in favor of browser-enabled solutions, so as to ease the burdens and costs associated with desktop/laptop management while opening up new accessibility approaches like hand-helds and other wireless devices. SAP has supported that vision since 1996, with the advent of Internet connectivity in R/3 3.1G. But only in the last few years have I really seen this take off.

SAP AG offers quite a few accessibility options today when it comes to mySAP solutions. The classic SAPGUI and its revamped and more capable EnjoySAP SAPGUI represent one end of the spectrum. This approach is safe, very mainstream, and very easy to implement. And the SAPGUI we have today is extremely comprehensive, supporting all mySAP components through a single interface, which is unlike the approach a few years ago where each so-called “New Dimension” product like BW or APO required its own GUI. But the SAPGUI still represents a typical application-specific approach to accessibility; each end user installs the client on their desktop or laptop, or runs the SAPGUI from a network share, and off they go.

Other accessibility approaches are available, however, as you see in Figure 3.6. The original WebGUI, for example, is based on HTML and provides connectivity via Microsoft's Internet Explorer and so on. And a more recent addition, the JavaGUI, allows native Java-based access to SAP. Both of these approaches fulfill an Internet-based approach to connectivity, and subsequently simplify the desktop (assuming Internet connectivity is a standard desktop offering at your particular company, of course).



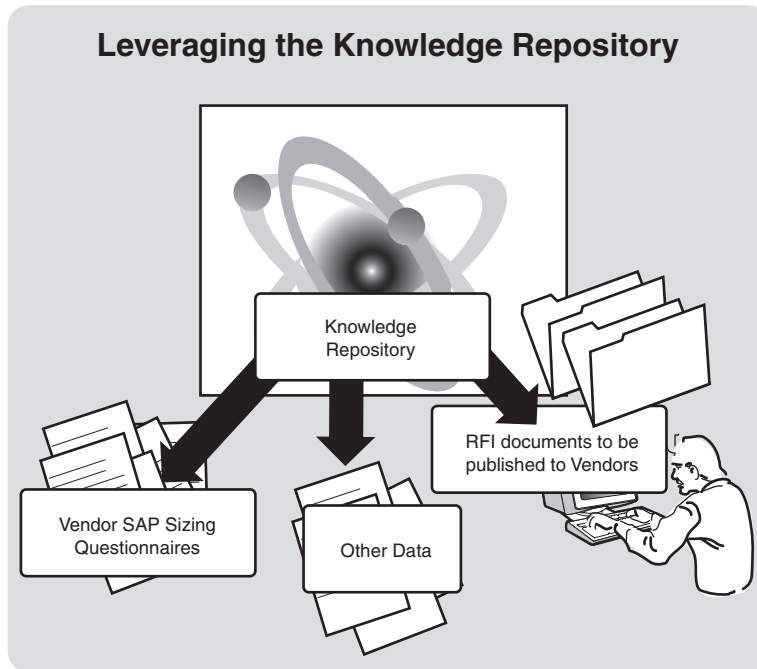
**FIGURE 3.6** Access to mySAP solutions is quite varied today, ranging from classic and updated SAPGUI options to newer Web-enabled versions.

## Capturing Your mySAP Solution Vision

As you work through all the different system landscape characteristics, considerations, and options, it is necessary to document why a particular approach or product was selected, and how it impacts the vision of the project. This documentation eventually finds its way into a *Knowledge Repository*, which is simply a documentation vehicle where assumptions, constraints, and so on are all maintained. This information serves as a set of boundary conditions and assumptions, useful later as you eventually engage various solution stack partners in fashioning your SAP solution, as you see in Figure 3.7.

In this way, whether you elect to publish a *Request for Information* (RFI) or complete a number of SAP Sizing Questionnaires, all of your hard data and related explanatory reasons for implementing each mySAP component in a particular way will be at your fingertips. The ability to share all this data consistently with all the prospective

hardware and software vendors is important—doing so enables a much better apples-to-apples comparison between different vendors’ solution approaches later on, as you will see in Chapter 7. Further, as new information comes to light, or questions are posed by these prospective vendors, the Knowledge Repository will naturally lend itself to collecting and managing this incremental data of evolving constraints, assumptions, requirements, needs, and so on.



**FIGURE 3.7** The Knowledge Repository will eventually provide input into either an RFI process or in answering SAP sizing questionnaires provided by various SAP Solution Stack vendors.

### Leveraging SAP Sizing Questionnaires

Although the SAP Sizing Questionnaire is covered in detail in Chapter 7 and elsewhere, a quick discussion is in order here as it pertains to vision. Each vendor’s SAP Sizing “Q” can potentially bring to light different areas of concern that you may not have yet considered. Microsoft’s SAP Sizing Q is focused on their products and capabilities, HP’s on their products and capabilities, and so on. As each organization’s SAP Competency Center updates and republishes its respective Sizing Q, it tends to promote new high-availability, performance, manageability, and similar offerings.

Thus, the Sizing Q in and of itself can prove valuable in terms of educating prospective SAP customers. This may in turn help you to fill in gaps in your vision, or facilitate getting you better acquainted with IT and business drivers that otherwise might not be identified until much later in the implementation.

## Developing a Request for Information

Rather than completing a whole lot of different SAP Sizing Qs, and going through everything that such an approach would entail after the fact, many companies choose instead to author and publish a Request for Information (RFI). I still promote the idea of going through various SAP Sizing Questionnaires in the name of education, of course, but using an RFI approach can be a much cleaner method for moving a project along.

A good RFI takes time to develop, though—a lot of time, usually. I have included a sample RFI (and related appendixes) on the Planning CD, not only to have something to walk through here, but also for you to use as a template of sorts if need be. Keep in mind that my 20-page sample RFI is quite short compared to what you might need to publish, though. My RFI addresses only two products, R/3 and APO, and contains very little in the way of legal terms and conditions. Your RFP could very well exceed 100 pages if you choose to implement a number of mySAP components or if you employ an ambitious legal department.

An RFI should include or take into account the following:

- General information, such as your company name and contact information, background data, and why the RFI is being published.
- Instructions to each potential RFI respondent as to how to complete the RFI. This includes how to address questions, details on the proposal process, confidentiality details, any disclaimers, and other administrative details.
- Terms and conditions, including the scope of the project, payment terms, minimum integration requirements related to existing/legacy systems, and so on.
- Requirements that must be met by the respondent, including any vendor and account management information you want to capture, the need for other SAP References, details surrounding the pricing model (including lease versus financing discussions), how to factor in maintenance windows or other planned downtime windows required of the proposed solution, hardware quotes, requested professional services quotes (for installation, migration of data, training, and so on), and any other information that may help you make a decision (such as each respondent's relationship with SAP, or the various database vendors, or a particular disk subsystem vendor, and so on).

In addition to covering the basics, an RFI normally contains or references various appendixes, too, which are designed to either share supplemental information or to enforce a certain type of formatted response. Let's walk through the seven appendixes I've included on the Planning CD:

- Existing Equipment Matrix—Documents what is in place today that must either be replaced by the new system(s) discussed in the RFI, or integrated with these same systems. This can also include a breakdown of existing SAP instances, or other enterprise application installations, that are currently productive. And if a current disk subsystem solution is already in place and expected to be leveraged by the respondent, details must be provided here as well.
- Software to be Implemented—Here, the Solution Stack as you imagine it at this point is shared, including expected versions of each mySAP component, database systems, operating system releases, enterprise management packages, and so on.
- Implementation Timetable—Represents an organization's hard requirements or possibly just a "best guess" as to when the new mySAP solution needs to be implemented.
- Reference Form—Applicable to absolutely *all* RFI respondents, though most often focused on potential hardware, software, and implementation partners.
- Cost Submission Worksheet—Consists of a standardized worksheet that forces an RFI respondent to price the project *your* way. This allows you to perform true apples-to-apples comparisons after all RFIs have been turned in.
- Staffing Matrix—Encourages each RFI respondent to give thought to staffing in terms of pre-engagement, during the engagement, and post-Go-Live.
- Sample Agreement with Terms and Conditions—Allows everyone responding to the RFI to understand up front what kind of legal constraints and financial commitments are expected.

Not all of these appendixes absolutely must be published with your RFI. In my experience, however, the kind of information provided in these seven appendixes is exactly what is needed by a respondent to either craft an SAP solution that really meets your needs, or to provide you with enough data to make an intelligent decision as to whether a vendor has what it takes to be your partner.

In Chapter 7, we will go through the remainder of the RFI process in detail, including how to compare and evaluate RFI responses, how to create a short list of RFI respondent candidates, approaches to making your final decisions, and more.

## Revisiting the SAP Infrastructure Implementation Budget

With the information gleaned from refining our solution vision and building the basis for an RFI, we should be in a pretty good position to review our SAP Infrastructure Implementation Budget (SIPP) again. The SIPP needs to be updated to reflect new cost models, new business requirement-driven technology drivers, the need for incremental or other skillsets, and any other information that in effect changes our budget.

Based on these new budget numbers—one-time implementation costs as well as recurring license, administrative, operations, and similar costs—you should be in a position to truly consider outsourcing as a viable alternative to hosting everything internally. This is discussed next.

## Outsourcing—Another Method of Achieving Your Solution Vision

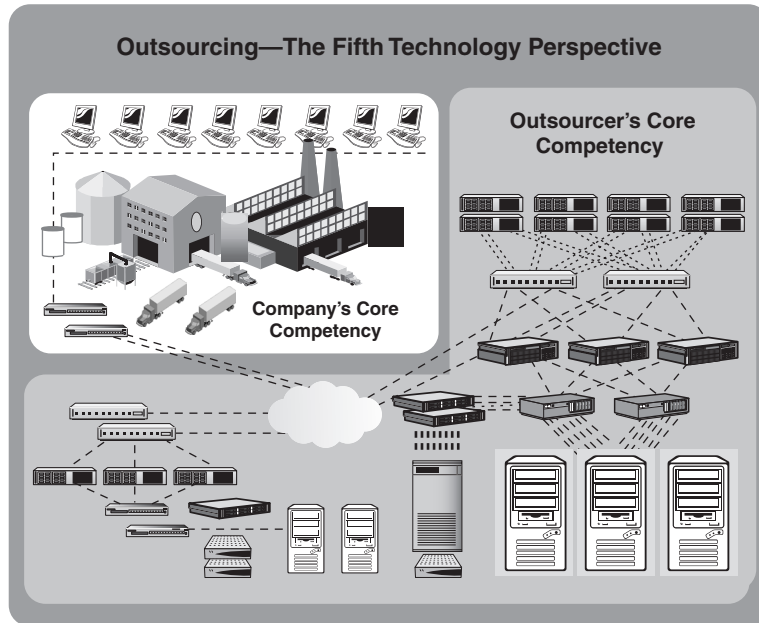
Outsourcing is what I consider the fifth technology perspective, after the Conservative, Mainstream, Close Follower, and Leading Edge approaches. Most of this book assumes that your company owns and manages the SAP infrastructure necessary to implement your SAP solution. I also assume that the members of your SAP Technical Support Organization are employed or contracted by you, and not by a third-party outsourcing firm. In these final few sections of Chapter 3, however, I open the door to considering *outsourcing* these key assets instead, as illustrated in Figure 3.8.

What drives organizations to outsource? In a recent IDC study, the volatility of our global economy was labeled as the primary consideration. The study put forth the following ideas:

- Making large investments in computing infrastructure is not wise in today's economy.
- A company should instead let experts in the field of enterprise computing resource management make these investments, leveraging *their* core competencies in these areas.
- All non-core functions should be considered for outsourcing, allowing an organization to instead invest time and resources in its own core competencies.

This is really no different than in the past, when companies turned to outsourcing firms to cut costs. But today things are a little different, and cost is less a factor than pure *adaptability*, which is the ability of a company to make changes quickly so as to stay competitive or position itself better with their customers, vendors, suppliers, and so on. In a nutshell, adaptability equates to strategic benefits, rather than the simpler and more tactical cost-cutting benefits realized a decade ago through traditional outsourcing.





**FIGURE 3.8** From both a business and IT point of view, outsourcing represents a unique technology perspective.

Intelligent outsourcing represents one method of becoming more flexible and adaptive, while still cutting costs. Outsourcing can mitigate risks relative to economic uncertainty as well, especially when the outsourcing agreement leverages the core competencies of each party. The really good outsourcing organizations, confident in their ability to execute, are more than willing to assume incremental risk. And with other risk-reward elements coming into play, such as those around meeting service-level agreements and availability targets, the best outsourcers are so convinced that they can do a better job of managing your resources and minimizing your downtime than you can that they're betting their revenue stream—your company's monthly check to them—on it.

With all of this in mind, exactly what should you outsource and what should you keep in-house? The short answer includes anything that is technology-intensive or complex from a process perspective. This easily explains why outsourcing SAP Disaster Recovery responsibilities is growing in popularity—DR meets both criteria in a big way.

### **Prerequisites of ITO—Information Technology Outsourcing**

Although companies today can outsource technology or business processes, my focus in the remainder of this chapter is on Information Technology Outsourcing (ITO);

Business Process Outsourcing (BPO) is the label given to process-oriented outsourcing. A quick list of ITO prerequisites is in order before we move on, however. To really benefit from an ITO relationship with an outsourcing partner, consider the following “must haves”:

- The outsourcer must be flexible and able to adapt to your needs, both short-term and long-term. Thus, a clear understanding of the iterative nature of successful outsourcing is needed, as both tactical and strategic needs will morph over time. A rigid engagement and change-management model will leave you worse off than before.
- Your company’s goals and objectives must align with the outsourcer’s capabilities—if the outsourcer does not specialize in mySAP, or is uncomfortable providing references that otherwise prove their capabilities, walk away.
- A well-defined and articulated set of expectations must be communicated to the outsourcer. For example, your service-level agreements, requirements surrounding any systems management information you want to see on a regular basis, and so on, all must be clearly communicated up front.
- For global outsourcing arrangements, a good cultural fit is very important, too. At minimum, understanding your outsourcer’s culture is essential. But it’s really helpful to understand specific traits and tendencies. For example, in some cultures people tend to avoid sharing bad news with their clients, or in other cultures, it is not acceptable to answer a question with a simple “no” without providing details as to why.

If both parties meet these prerequisites, and you are comfortable with your potential outsourcer, you are a good fit for at least *considering* outsourcing.

### **Potential Benefits of Outsourcing SAP Infrastructure**

The benefits you should reap from an ITO outsourcing relationship, compared to retaining control of your SAP assets internally, include the following:

- Less downtime and better availability. This includes both planned and unplanned downtime, as the outsourcer can presumably leverage their economies of scale, superior maintenance processes, and access to talented mySAP personnel.
- The same or greater level of flexibility. As your business needs change, so too should the system that supports these needs. This should manifest itself in a number of ways, including a full life cycle offering and “one-stop shopping.”
- Better consistency from a personnel perspective. Although employee and contractor turnover is not what it has been in the past—it’s quite reasonable

today—a successful outsourcing provider should still be able to retain its scarce technical resources longer than you can.

- Simplified budgeting and financial management of assets.
- High-quality approach and delivery.
- Reduced IT and supporting costs and little or no up-front capital expenditures on hardware and infrastructure equate to *more cash*. This is especially true when it comes to using offshore outsourcing partners.

To this last point, offshore outsourcing has been described as “counter-recessionary” simply because offshore costing models are so dramatically lower than U.S.-based models—recession or not, you are saving more money than otherwise possible by holding on to SAP assets internally. And with so many other countries beginning to compete successfully with India, which has dominated offshore outsourcing over the last five years (and currently owns 85% of all offshore outsourcing, according to Meta Group), the cost models will only continue to improve over time.

### **The Shortcomings of Outsourcing in the Real World**

Historically, it has been difficult to find more than a few success stories where the company was so enthralled with their outsourcing partner that they could not help but tell everyone. My direct experience with outsourcing is pretty shallow, but from the stories my colleagues have shared with me, the following points seem to hold true:

- Loss of control seems to be the biggest concern. This relates directly back to the flexibility and adaptability that outsourcers today tout as compelling benefits.
- Less than overwhelming cost savings is another. Organizations that expect an order-of-magnitude cost reduction may be disappointed. Numbers like this are possible, true, but only if your own organization is so fat and bloated with overhead that you simply couldn't help reducing your IT bill in half.
- No perceived difference in the amount of time it takes to resolve system problems. This is especially true if your company's IT organization does their job quite well, leaving little room for improvement for an outsourcing partner.
- Outsource contract timelines vary considerably. One of my large SAP customers was persuaded to sign a seven-year outsourcing agreement a few years back. Seven years! That's an eternity in the world of IT, and they are “locked in” until the contract expires, lest they turn over a hefty penalty for early termination.
- Another customer of mine signed an outsourcing deal structured such that incremental processing power required by the customer during the life of the contract would be billed “per server.” Less than a year later, they began to

understand what that meant, as the outsourcer increased its revenue stream by meeting their new mySAP requirements with many two-processor servers instead of fewer larger servers.

- I've been told by customers that they sometimes feel “nickel and dimed to death” by their outsourcing provider. Every unplanned change to their environment, every new addition to the SAP system landscape, and so on add up to incremental and costly fees that were never envisioned by the original contracting team.

In looking back at the preceding list, it seems to me that many of the stumbling blocks stem from contractual issues rather than true outsourcing shortcomings. That is, performance problems were hard to find, and it seemed as though service-level agreements and general system availability were not issues, either.

### Analyzing Outsourcing Versus Doing It Yourself

Just like hosting and managing your infrastructure internally, outsourcing touches every facet of your end users' experience with their mySAP solution. However, if an outsourcer can provide the same or better levels of service, responsiveness, and system availability, while successfully retaining the skillsets and expertise needed to keep an SAP solution humming along, and do all of this more cost-effectively than you, by all means outsourcing should be considered the forerunner in achieving your SAP solution vision.

The next step is to verify not only that the outsourcer is built upon a foundation of sound business fundamentals, but that it can demonstrate the following abilities:

- Can be effectively held accountable to deliver what it promises, through penalties and similar service-level-based fees
- Can show you proof of how it has accepted responsibility for its mistakes and shortcomings in the past
- Can point to a clear and time-proven methodology for planning, deploying, upgrading, supporting, and otherwise managing the mySAP enterprise computing resources of *other* customer organizations
- Can show you how its own processes and procedures are subject to continuous improvement

Why are these so important? Because they give an organization a way of comparing themselves to the best that outsourcing can provide. And because there is really no cost savings that will ever make it acceptable to circumvent these basic business fundamentals! In other words, flexibility, service, system availability, and authentic customer-service values mean a whole lot more to your end users than price ever will.

## ASP Hosting for SAP

Another approach to managing resources outside the boundaries of your internally housed data center is through an Application Service Provider (ASP). What exactly is an ASP? According to IDC, ASPs provide a contractual service offering to deploy, host, manage, and rent access to an application from a centrally managed facility. ASPs are responsible for either directly or indirectly providing all of the specific activities and expertise aimed at a managing software application or set of applications. Different ASPs tend to focus their services in different application areas—hosting traditional file, print, and Web services, and enterprise applications like mySAP make up the bulk of these. This is what tends to differentiate ASPs from general ITO and BPO outsourcing providers.

According to Gartner Group, ASPs deliver application functionality and associated services across a network to multiple customers by way of a “pay as you go” pricing model. As in traditional outsourcing, the value proposition clearly has to do with providing access to customer applications without the systems, staffing, and manageability challenges. After dramatic consolidation over the last two years, the strong remaining ASPs are growing again because they provide the following:

- Access to expensive and skilled IT professionals
- Knowledge in hosting mySAP and other applications
- Better reliability than most customer organizations enjoy, in regards to network and other infrastructure resources
- Alternative and flexible financing arrangements available
- Ability to include value-adds like e-trading, home pages for different organizations, and other Internet-focused offerings

Of course, like ITO outsourcing providers, an ASP’s specific knowledge and experience in supporting mySAP solutions, its reputation, an installed base of customer references, and overall financial stability are important considerations prior to securing their services.

## HP’s Utility Data Center

Another very different method of accessing and provisioning scarce hardware resources is promised by HP. Announced in November 2002, HP’s Utility Data Center (UDC) does not seek to simply partition hardware platforms in a different way, or enable software-based workload management. According to HP Software’s Chief Technology Officer, Rick Hayes-Roth, “UDC is a software and services package that creates a data center infrastructure that administrators will wire once, then reconfigure dynamically.” Ultimately, a UDC data center administrator transparently redi-

rects computing resources to any application that needs it, thereby bringing true dynamic *provisioning* to the SAP Data Center by enabling different mySAP applications to access needed point-in-time resources.

In doing so, UDC promises to reduce data center costs by as much as 50%, a compelling proposition not only to large corporate enterprises but perhaps even more so to outsourcing vendors, managed service providers, infrastructure application service providers, and the like. To achieve these goals, UDC customers actually receive the following:

- HP consulting services, to architect the data center infrastructure.
- HP services, which are leveraged to keep the infrastructure up and running.
- HP OpenView's Integrated Services Management (ISM) software, necessary to manage and control resources under the umbrella of UDC.
- Cisco-based networking solutions, tying all assets into one computing fabric.
- An end-to-end solution vision that can easily evolve as an organization evolves. UDC supports multiple operating systems (HP-UX, Windows, Linux, and other UNIX variants), for example, and by tying these all together, enables the creation of huge compute and storage pools.

UDC will deliver on the ROI that all customers hope for but never fully achieve with current solution deployment models, simply because over-sizing and otherwise building scalability into each mySAP enterprise component, for example, is expensive!

There will be challenges that must first be overcome by companies wanting to adopt UDC, however. The most unusual will be related to the fact that UDC is not a single product or approach—instead, it's a marriage of hardware, software, people, and service-level agreements. Bringing all of this under the umbrella of ISM will be nothing compared to the time it takes to simply review current assets and then plan for a UDC solution.

## Tools and Techniques

In addition to the sample RFI and related appendixes, I have also included a form for documenting the SAP solution vision, and electronic versions of each figure found in this chapter, in PowerPoint format.

## Summary

In this chapter, I discussed the need and importance of crafting a solution vision prior to designing or implementing mySAP. Different technology perspectives were

covered, and the importance of refining the solution vision in regard to SAP system landscape requirements and constraints was covered as well. After the vision began to take form, I reviewed a few methods of capturing all of the data that came together to create the vision—constraints, assumptions, boundary conditions, and so on—including using a knowledge repository, developing an RFI, preparing for the SAP sizing process, and more. I then wrapped up Chapter 3 with a discussion on outsourcing, including leveraging ASPs and newer approaches like HP's Utility Data Center offering to improve systems manageability, provide better resource provisioning, and ultimately create a better customer experience for mySAP end users. In the next chapter, I build upon this foundation we have created to begin identifying and filling key SAP Technical Support Organization staffing roles.

