To Migrate to Not to Migrate — Proper Preparation Is the Key to Making the Right Decision and Ensuring a Successful Migration

George W. Anderson

George W. Anderson is a senior SAP Basis Consultant & Project Manager for HP's Consulting & Integration Services organization, where he focuses on implementing and optimizing HP and other technology vendor platforms for SAP customers. George is a certified SAP Technical Consultant and PMI PMP and a veteran of numerous SAP implementations, upgrades and performance-oriented projects.

(complete bio appears on page 68)

There are many reasons you might want to consider what SAP terms a *heterogeneous SAP OS/DB migration*, or simply an *OS/DB migration*. For IT shops that are happy with their current system's performance, reliability, and scalability, the reason might simply be cost — everything else equal, significant cost savings can be realized by moving to an industry standard platform, for example. By the same token, though, IT shops struggling to meet performance and scalability service level agreements (SLAs) or tasked with integrating new systems may be tempted to undertake an OS/DB migration. IT teams seeking improved agility or pursuing simplification and consolidation strategies may do the same. In addition, business teams driving functional upgrades to any number of mySAP solutions may push a current platform's performance needs beyond that which can be addressed by performance tuning. And management teams tasked with integrating users from additional business units, melding acquisitions into current systems, or introducing complex business processes encompassing new workloads may simply need the horsepower and scalability that only a migration to a new technology platform brings.

The process of deciding whether a migration is indeed in order, however, is as challenging and complex as conducting the migration itself, if not more so. While the technical migration process itself has a clear goal, navigating the business factors and needs involved, and marrying them to IT factors and needs, are not as clear cut. Not to mention the stakes at hand — poorly tested or executed migrations invite repercussions that can impact the viability of an entire company, as financial, inventory/warehousing, supply chain, payroll, and countless other systems may be affected by such a fundamental change. Previous articles have introduced the migration process and discussed the

technical export and import procedures in depth.[1] This article helps IT and business teams alike make the right decision on whether and how to migrate by walking you through the critical areas that will make or break the success of your proposed migration:

- Understanding the needs of key migration stakeholders
- Analyzing the costs of a proposed migration
- Weighing the risks of a proposed migration
- Choosing a method to perform a proposed migration

In addition to showing you how to navigate these key pain points, I also review some critical but commonly missed steps that my colleagues and I have experienced on actual migration projects, show you how to mitigate some of the risks involved in going live on a new platform (by selecting the right partners, building a solid migration team, performing functional and load testing, and optimizing performance), and discuss tips and other special considerations associated with migrating very large databases.

And even if you are not an executive, business leader, or IT or business manager, you'll find a wealth of information designed to uncover key decision points or solve fundamental technical issues. Those tasked with validating the financial viability of a proposed migration, for instance, will benefit from this article, as will the members of the technical team faced with completing a migration.

Before we dive right into the thick of things, however, let's get reacquainted with the big picture. In the next section, we'll look at why SAP customers migrate, which offers insight into potential drivers of your own migration, along with what constitutes an OS/DB migration, which is critical to making the right up-front decisions for a successful migration.

[1] For a detailed look at what it takes to conduct an OS/DB migration, see Bert Vanstechelman's article, "A Step-by-Step Guide to Planning for a Successful OS/DB Migration Project" (*SAP Professional Journal*, November/December 2002). To review the technical process of migration, look to the recent articles written by Michael A. Moore in the March/April and September/October 2004 issues of *SAP Professional Journal*.

What Is an OS/DB Migration?

In simplest terms, an OS/DB migration involves migrating from one operating system or database platform to another, amounting to a technology upgrade rather than a similarly complex, but fundamentally different, application upgrade. In the real world, most OS/DB migration projects mandate server and disk subsystem upgrades as well, making a seemingly simple technology refresh that much more complex.

So why pursue such an undertaking? What are the compelling reasons? And what exactly is involved in a migration? We'll look at the answers to these questions in the next few sections.

Why Migrate?

What drives migration activity within SAP's installed base? While the need for better performance and increased scalability has historically driven migration activity, it should come as no surprise that total cost of ownership (TCO) is the primary driver of late.

First, I see technology-oriented TCO factors, such as the need to otherwise drastically upgrade or retire existing servers, driving the adoption of new strategic OS platforms. In my experience, OS migrations are primarily between different versions of Unix, between Unix and Microsoft Windows, and to a lesser extent between Unix and IBM midrange/mainframe systems. These migrations are brought on by the need to reduce the cost of the very foundation upon which SAP runs. As industry-standard Microsoft Windows and Linux computing platforms now deliver the kind of performance and scalability once only obtainable from proprietary Unix and mainframe platforms, for example, part of the transition naturally involves moving to a new operating system.

Second, many of my own customers have considered migrating from their legacy RDBMS platforms to Microsoft SQL Server, IBM DB2 Universal Database (UDB), and to some extent the SAP-certified open source database mySQL MaxDB in order to reduce

TCO. Given the strides made by Microsoft, IBM, and SAP in developing relational databases that boast both solid performance and lower costs, such DB migrations can make sense at a number of levels. For example, with its huge SAP market share, and at something like half to a quarter the cost of competing RDBMS licenses, it's hard to ignore the savings Microsoft SQL Server can bring to the SAP IT table, not to mention its straightforward administration and support requirements. With the growth of Linux as a credible platform for SAP and other mission-critical applications, I have seen a fair amount of migration activity around mySQL MaxDB as well (especially in Europe and Asia); at a fraction the cost of some of the most popular database releases in the world, small to mid-sized SAP shops reluctant to leave their familiar Unix platforms have an appealing alternative at their disposal. Finally, the latest IBM DB2 UDB release (Version 8.2, also known as "DB2 Stinger") boasts improved hooks into SAP, and in my experience is getting more attention from large SAP customers as well as targeted mid-sized shops. With all of this focus on lowering the total cost of deploying and maintaining SAP, it's only reasonable to assume that very few CIOs would be uninterested in at least *exploring* the various available database options.

In addition to decreasing hard costs, there are many other reasons to migrate between technology platforms for SAP. A company may seek to change vendors for strategic reasons, for example, or it might seek to pursue an end state that requires fewer people, less system administration time, less-costly maintenance or service and support, and so on. In the same way, the need for a dramatic improvement in availability, scalability, security, or performance may be driving organizations to transition operating platforms.

Consider these and other reasons for pursuing an SAP OS/DB migration:

- To retire an existing older platform in favor of transitioning to a newer platform
- To significantly improve online transaction response times through a technology refresh
- To significantly increase report and batch throughput performance in response to shrinking batch-job execution windows
- To reduce downtime and increase availability by migrating to a platform that supports better or more appropriate high availability solutions
- To move to a processing platform that supports greater disaster recovery capabilities or improved business continuity solutions
- To strategically embrace a platform that can reduce headcount related to database administration, SAP Basis and technical administration, enterprise computer operations, and even core infrastructure support
- To better leverage existing staff by increasing full-time employee bandwidth, therefore enabling the SAP IT organization to pursue value-added projects *without* increasing headcount
- To standardize on fewer hardware platforms that support multiple operating systems, database releases, and applications — in support of IT consolidation initiatives, for example — such as HP's Integrity or IBM's pSeries platforms
- To standardize on hardware platforms that provide greater agility and flexibility, including support for hard and soft partitioning, component-level provisioning, and simpler yet more robust upgrade and migration roadmaps
- To enable superior integration below the SAP application layer, or to augment business process management made possible through SAP NetWeaver, SAP Exchange Infrastructure (SAP XI),[2] SAP Business Information Warehouse (SAP BW), and others

The central theme across the board often boils down to *cost* first and foremost — whether by reducing hard costs, increasing soft dollar savings, or through cost avoidance. That is, in the world we

[2] For a detailed introduction to SAP XI, see the article series by Manish Agarwal in the March/April and May/June 2005 issues of *SAP Professional Journal*.

find ourselves today, more often than not the central issue in IT is simply how to save more money while maintaining the service levels IT has signed up to meet on behalf of the business. Hard dollar savings, like those associated with database licensing and hardware acquisition, are naturally the easiest to quantify.

When cost is less an issue, other factors quickly come into play. For instance, there will always be systems that beg to be migrated for better month-end scalability or to take advantage of higher-performing platforms — these requirements are about meeting business needs, presumably better balancing costs and performance, or costs and scalability, in the process. In these cases, cost takes a backseat to performance, scalability, reliability, disaster recoverability, manageability, security, enhanced integration, increased agility/flexibility, and so on.

Despite the driving factors, the message is clear: OS/DB migration projects continue to be pervasive across SAP implementations because they make sense to both the CFO and the CIO (we'll look at understanding the needs of executives in an upcoming section). It is a good idea to look closely at platform alternatives — based on lifecycle refresh requirements or in response to changing business requirements that drive core technology changes, for example — at least every three years, or whenever the current technology platform is nearing the end of its vendor-provided support, to determine if a migration makes sense. Work with your current vendor's SAP Competency Center and at least one or two other vendors to draft solution sizings customized for your specific environment (more on this later). Investigate alternative or complementary technologies. Look into new methods of hosting your current and future SAP workloads. Then price out these alternatives, paying attention to initial acquisition as well as ongoing support costs. At the outside, infrequent application upgrades should drive at least some of this activity. Indeed, every SAP organization in production today has ample reason to evaluate the extent to which an OS/DB migration might prove strategic. After all, CFOs chartered with maximizing shareholder value and CIOs tasked with enabling cost-effective business processes both have it in their best interests to reduce TCO.

Much therefore hangs in the balance relative to pursuing an OS/DB migration; the importance and volume of raw data sitting in the thousands of tables that make up a typical SAP database represents the lifeblood of a company. Yet every dollar needlessly spent in preserving this lifeblood is a dollar that otherwise would have found its way to the bottom line. Thus, a balance must be stricken between all stakeholders — IT and end users, those tasked with financial oversight and those tasked with providing support to the business, and so on. And because stakeholder needs naturally affect this balance over time — as the needs of end users change, or company financials are squeezed during lean times — a holistic review is truly mandated every few years.

So what exactly does it mean to perform an OS/DB migration? We'll look at this next.

What Constitutes an OS/DB Migration?

Fortunately for its customers, SAP migrations do not necessitate revamping the application itself. That is, the version or release of the SAP application sitting atop the OS and DB does *not* change in an OS/DB migration. Sure, it might change soon after, but not within the scope of the OS/DB migration project. Such a change would constitute a project in its own right, an SAP functional or application upgrade (discussion of which is beyond the scope of this article). So when we toss around the term "OS/DB migration," we effectively eliminate a major complicating factor from the equation — the current release of SAP R/3 or your mySAP Business Suite or SAP NetWeaver component gratefully remains the same throughout the process.

Other components of your unique SAP solution, such as Internet and accessibility options including SAP Internet Transaction Server (ITS) or Citrix MetaFrame, normally remain untouched as well. And middleware and other bolt-ons don't change either, though exceptions exist — for example, in the case where a particular bolt-on product requires an update or upgrade to work with a particular vendor's operating system or database. In this situation, it can be

assumed that the integration point is "outside," or "underneath," the SAP application layer. Such a change might reflect the introduction of a 64-bit version of a bolt-on in a 32-bit environment or the need for an OS-specific scripting language or DB-specific database call — all of which unfortunately represents real-world issues that are bound to occur.

> ✓ ***Note!***
>
> *Although the version of your SAP product or component remains the same during a migration, it's important to keep in mind that the existing release must still be reinstalled on the new target platform. Further, given that many OS/DB migrations stem from a review of the company's current SAP platform, and the subsequent need to retool this platform for increased horsepower or availability, it's inaccurate to say that OS/DB migrations and functional upgrades are mutually exclusive. Indeed, they often coexist, though the bulk of their timelines typically do not overlap. Instead, a functional upgrade oftentimes precedes or follows an OS/DB migration.*

What an OS/DB Migration Is Not!

Whether the OS or DB or both are changed, these fundamental platform shifts are referred to not only as SAP OS/DB migrations but also as *heterogeneous system copies*, a reasonable though rather technical label. You'll find the very generic term *system copy* used as well, or rather misused — I suggest avoiding it, because a system copy refers to pretty much any method employed to create or copy an SAP database. The term OS/DB migration, on the other hand, usually implies a system copy performed exclusively through the use of a select set of SAP-approved migration tools and practices found in the SAP OS/DB Migration Kit.

So, just as important as what constitutes a heterogeneous migration is what does *not*. Consider the following scenarios:

- Updating or upgrading an in-place operating system, say from Microsoft Windows NT 4.0 Enterprise Edition to Windows 2000 Advanced Server or Windows Server 2003 is *not* a heterogeneous migration — it's simply an OS upgrade.
- Upgrading from Oracle 8i to 9i or 10G is *not* a heterogeneous migration — it's a database upgrade.
- Moving from a 32-bit OS to a 64-bit version of that OS is *not* a heterogeneous migration either — again, it's simply an OS upgrade.
- Exchanging one server platform for another that supports the same OS does *not* constitute a heterogeneous migration — it's a technology refresh at best. This is most applicable to Microsoft Windows and Linux shops moving from one Intel-based server to another, for example, or IT shops replacing their Sun gear with a Fujitsu solution that also supports Sun Solaris.
- Introducing a new disk subsystem (therefore requiring a database copy from the old storage system to the new one) does *not* constitute a heterogeneous migration, either — it's also a technology refresh.

These scenarios are very appropriately called *homogeneous system copies* and are inherently less complex to pull off than heterogeneous system copies — they don't require a certified SAP migration consultant, a mandatory review of the source and target systems, purchase of the SAP migration service, and so on (we'll look at more of this later when we discuss different migration methods and approaches). And, as you would suspect, homogeneous system copies are also more common than OS/DB migrations.[3]

Now that you have a solid understanding of what an OS/DB migration is (and just as important, what it

[3] For more on homogeneous system copies, see the *SAP Professional Journal* articles "A Homogeneous System Copy in 60 Minutes? It Can Be Done!" (January/February 2002) and "Database System Copies Made Easy — A Guide for Copying an Entire R/3 System to a New Windows Platform" (May/June 2003).

is *not*), along with what drives migrations, over the remainder of the article I will show you how to set the stage for a successful migration by evaluating the needs of key stakeholders, the costs and risks involved in a proposed migration, and the various methods and approaches available to perform the migration itself.

Understanding the Needs of Key Migration Stakeholders

From a stakeholder perspective, an OS/DB migration is risky and complex. *Replatforming*, as it is called by some, requires a certain amount of downtime, for instance, and therefore affects the system's end-user community. The SAP IT shop tasked with replatforming may also be faced with multiple learning curves as it seeks to understand and optimize a new platform for production. Operational procedures and methods employed for daily tape backups, solutions used to provide high availability, disaster recovery processes and practices, and so on can easily be affected as well.

Clearly, lower cost of ownership isn't free, but with these potential migration headaches comes the potential for equally great benefits (no pain, no gain, right?). Additional scalability and performance can mean the difference between poor business decisions and good ones, slow time-to-market reactions and greater responsiveness to customer demands. Lower costs can free up capital that can be used to expand the business or go after new markets. We'll take a closer look at these benefits and more as we examine the perspective of company executives, business leaders, senior IT management, and other key stakeholders when it comes to a migration. Understanding stakeholder needs will help determine migration requirements (and feasibility, for that matter) as well as help build stakeholder buy-in, which is critical to the success of any migration.

Executives

In the eyes of a company's executives, an SAP migration represents a strategic long-term investment geared toward better hosting of mission-critical business processes thereby maximizing shareholder wealth.

The best approach to achieving executive buy-in lies in obtaining a solid understanding of executive needs. Consider the following as part of your migration evaluation, keeping in mind that the change inherent to a migration is almost never initially welcomed by the executive ranks (change means disruption, and disruption not only costs money but invites risk):

- Is there the potential to bring on a new business unit, or integrate an acquisition into the company's fold in the next few years? Such matters can be more easily addressed with a platform that scales well — more money spent today can reap significant cost savings in terms of meeting SLAs while enjoying increased availability, timely reporting, and reduced planned downtime (to conduct a technology refresh, for example) in the future.

- Is there a strategic business or IT relationship that needs to be considered? For instance, has the company decided to "bet the bank on Microsoft" or sign a favorable site-wide licensing arrangement with Oracle? Is a particular server or storage vendor the new standard of choice? In these cases, such decisions inherently limit technology options in some way, therefore impacting cost, availability, scalability, performance, etc.

- Is there the possibility of an outsourcing deal looming in the near future? Are there other sourcing alternatives or potential relationships with key partners and vendors that need to be considered? If so, a premature decision to migrate platforms may be nothing more than a waste of time and money; outsourcing can mean deploying a completely new technology stack replete with new staffing, after all.

- Will an OS/DB migration provide a competitive advantage with regard to business, financial, customer, or other insights? Will it enable seamless integration across the supply chain, or position the

company to extend its reach, and therefore decision-making abilities, into suppliers and vendors? Will it improve customer relationship management, or product "attach" rates? How exactly will an OS/DB migration increase shareholder value? You'll need to convince executives that the OS/DB migration in question will indeed add dollars to the bottom line, either through greater cost reductions or a combination of increased revenue and improved profitability.

With executives soundly in your corner, you'll be amazed at how well many of the remaining decisions and issues work themselves out. On the other hand, if you've failed to build a compelling business case, perhaps the fact of the matter is that a compelling business case does not exist, and a migration is therefore not warranted.

Business Leaders

Regardless of how you look at it, in the eyes of business leaders an OS/DB migration often spells "inconvenience." Think about it — if the current system is subject to greater than expected unplanned downtime, performance issues, security headaches, and so on, business leaders are already less than happy. A change represents yet another unknown, and at this juncture business leaders equate unknowns with even worse performance, even worse scalability, and so on. Convincing business leaders that a migration to a new technology platform will make matters better could amount to a lot of work.

If the system has been performing well, you might face even bigger challenges, though. Why? Because who in their right mind wants to put a well-performing system at risk by introducing new underlying technologies and processes? This is especially true for environments where uptime and performance have been better than expected, scalability has been a non-issue, the platform has enabled smooth integration between systems, and the migration is being pursued for TCO reasons more than anything else. The bar is set pretty high; a migration is likely perceived as an unnecessary risk that will jostle the bar at best, or knock it completely down worst case. Tasked with managing business processes well, business leaders could be your hardest sell yet.

The best approach in this situation is to work with business leaders in a preparatory kind of way, couched within an old-fashioned "sales job." If the end-user experiences have been poor of late, sell the migration in terms of enhanced online response-time performance, increased scalability (for faster month-end reports), better manageability and therefore improved availability, tighter and easier-to-manage security, and so on. If the current system has been performing well, position the new platform's greater flexibility and improved agility, or position its ability to meet new or changing business needs faster, better, and cheaper than business leaders ever thought possible. Do all of this well upfront, once you have the executive team on board.

Regardless of how you specifically choose to "sell" the migration, be sure you are prepared with a clear understanding of business needs and how those needs will be better met through an OS/DB migration. For example:

- Discuss what business leaders intend to pursue in the next three years, in the hopes of tying a proposed OS/DB migration into the steps necessary to fulfill the long-term vision for the business in some manner. Will new SAP functionality within existing systems or a complementary bolt-on be required? Will a completely new SAP NetWeaver or mySAP Business Suite component be implemented, which in turn might invite drafting new IT standards or revising existing ones? All of this speaks to more tightly integrated and therefore complicated business processes — challenges that can be alleviated by deploying the right application platform.

- Assuming a poorly running system, discuss the delta between current and desired response-time metrics for key online transactions. Talk about batch report requirements, too, including limitations users have become accustomed to (such as the ability to only run certain reports during

off-peak hours because "that's the only time the system can handle it," thus complicating and slowing down decision making), and how a new platform might overcome these limitations and enable the business to reach new heights.

- Even though the current system might meet existing SLAs, discuss opportunities to provide still better performance, greater batch execution windows, or more seamless availability, thus making the scheduled downtime necessary to complete the migration an easier pill to swallow. In these cases, you might sell the migration in the light that a bit of investment (near-term end-user downtime) will pay off handsomely in the form of increased responsiveness and the ability to better address unknown future business peaks. Or you might focus further on the downtime aspect, citing that a little downtime in the short term spent implementing a more available solution makes it possible to reduce both unplanned and planned downtime in the long term.

Don't forget to include business leaders in the migration process itself as well as the decision-making process; they will play a key "validation" role during and after the migration (by testing key SAP transactions to verify data integrity and continued functionality, for instance, before turning the new system back over to its larger audience). Such a role is paramount to data integrity as well as buy-in, and cannot be taken lightly. Alienating business leaders early on can spell disaster later. Bring them in early, make them a central part of the team (as should be the case, after all, given their central stakeholder role), listen to their needs, and do your best to balance these needs against your budget, your headcount constraints, and other limiting factors and stakeholder requirements.

Senior IT Management

With executives and business leaders on your side, you next need to look at ways to build buy-in from the IT management team ultimately tasked with supporting the new platform. In my experience, the greatest barriers to success can be found in old-school IT shops; technology biases, "we don't do that here" attitudes, and an unwillingness to change the status quo will without a doubt affect not only the success of an OS/DB migration, but whether such an option is fairly considered in the first place.

Like other stakeholders, the best approach to achieving buy-in from senior IT management requires obtaining a solid understanding of their needs. Consider the following as part of your migration evaluation, keeping in mind the perspective that senior IT management shares with business leaders and executives — to provide an accessible service at a reasonable cost and low risk:

- To what extent will the infrastructure support teams, SAP Basis/technology teams, database administrators, SAP data center and operations staff, SAP help desk, and others be affected?

- Relative to new technologies, how will the need for new or updated skills, the need to master new operational processes, or the need to learn new tools impact the IT team's day-to-day responsibilities?

- Who in the company's current IT team has the ability to get up to speed quickly, to mentor and help train longtime colleagues and new hires, or in some other way make a case for retaining most if not all of the current IT staff?

- To what extent will a new platform affect the IT team's goal of standardization and simplification? How will the new platform aid or hinder consolidation efforts? Are new technologies involved?

Using these questions as a starting point, I suggest you work to understand current technology limitations and how these limitations affect the company's ability to conduct day-to-day business as well as to meet seasonal and quarter-end processing requirements. Get everyone in the same room and hash out strategic goals, current competencies and skill sets, and a vision for the future of IT. Some additional questions to consider include: Is the business heading down the path of Enterprise Services Architecture (ESA) and Web

services, yet saddled with aged technology platforms? Will new or updated business processes dictate bringing in SAP XI or investing in other enabling technologies like Microsoft BizTalk or IBM WebSphere? Questions like these naturally open the door to conversations geared toward identifying and addressing business needs rather than introducing cool technologies; with business needs clarified, the application of an appropriate technology platform should more easily fall into place.

A migration that then carefully considers the linkage between the business and IT, and maps this to measurable IT deployment plans, is well positioned to gain the support of senior IT management. Be prepared to play the outsourcing card if need be, though, especially if there's a reasonable chance of it being seriously considered in the course of evaluating the next wave of necessary technology refreshes. When faced with the decision to consider adopting Linux and mySQL MaxDB, for example, or stand in an unemployment line while a system is outsourced elsewhere, most teams will line up pretty quickly behind your migration plan that not only reduces costs but saves many jobs in the process.

IT management therefore plays nearly as critical a role as other stakeholders. Why? Because business leaders and the executive team alike will be less apt to push for a migration if their own IT department is ill-equipped or unable to support it. If IT management does not sign up to address training and knowledge gaps, the project will be much less likely to get off on the right foot.

Once the idea of an OS/DB project begins to gain traction throughout the organization, it's time to get serious and think about the vendors, partners, and migration team members you will ultimately consider bringing in, and the associated requirements. We'll look at these considerations next.

Vendors and Partners

I suggest that you build a "short list" of vendors and partners that can provide the following:

- Server and OS expertise, and associated SAP skills
- Disk subsystem hardware, including skills in optimizing what traditionally amounts to the number-one bottleneck of most any enterprise application — the hardware underneath your SAP database
- Basic infrastructure support services in an SAP environment, from data center power and cooling facilities up through network operations, cabling, hardware rack-n-stack, server deployment, and more
- Experienced database administrators familiar with the proposed system
- SAP Basis/technology expertise relative to both the source and proposed target systems
- Certified SAP migration consultants (required by SAP for all heterogeneous migrations)
- Access to other SAP experts, especially tuning and similar performance optimization skills

Using this list, you can start to zero in on the vendors and partners best-suited to support the migration itself and the resulting new platform.

The Migration Team

Be sure to consider the project plan and the technical project management expertise you'll require to bring an OS/DB migration full circle. And all other matters being equal, use the "more skills in fewer full-time employees" mantra to drive staffing decisions. Such an approach to staffing an OS/DB migration team helps simplify the work of coordinating and completing the migration. For best balance, keep in mind that the level of expertise needs to be deep enough to get the job done properly; after this is achieved, seek complementary skill set "broadness." For example, in the world of SAP migrations to Microsoft Windows/SQL Server, it's quite advisable to seek out SAP Basis experts knowledgeable in both the server/OS combination as well as the database, and

to a lesser extent even the disk subsystem. In pursuing such a staffing strategy, 3 or 4 folks might fit the bill for what otherwise could have been a 10-person migration team! I'll provide some additional tips on building a lean and effective migration team later in the article.

Once you understand the considerations involved with your key stakeholders, and in turn the requirements that will drive the migration, you need to start evaluating some hard numbers associated with the technology platforms under consideration for the migration. I'll show you how to do this next.

Analyzing the Costs of a Proposed Migration

Although it would be nice to completely map out and quantify all the costs associated with an SAP technology platform under consideration, and compare it to one's current platform relative to its cost structure, such an exercise is complex and time consuming. Yet before a company invests hundreds of thousands, if not millions, of dollars in a new technology platform, at least *some* kind of financial analysis is necessary. Some of my own customers have quantified costs and savings via mature financial modeling like that afforded by earned value added (EVA) analysis, return on equity (ROE) exercises, and other established methods of measuring the value of tangible assets. These methods naturally consume a whole lot of time and resources, but are highly credible. A faster method along the lines of something I long ago coined a *delta TCO analysis* or simply *delta analysis* may be a better fit for your particular budgets and timelines, though. Before we can discuss this method, a quick recap on TCO is in order.

Technology, People, and Processes

What exactly *is* TCO? To understand a company's total cost of ownership relative to migrating from one platform to another, we need to look at three general subject matter areas. The first of these is *technology*, including all of the infrastructure and any hardware and software that must be leased or purchased. Next, the cost of all the *people* necessary to evaluate the current environment, migrate to the new environment, and subsequently manage ongoing operations. Finally, a TCO assessment must take into account the costs of the various systems management, administration, operations, migration, recovery, and other *processes* germane to a particular solution.

Further, TCO is not about a single point in time — it's about a *range* of time, such as three years. After all, if only acquisition costs were taken into account, everyone would be running SAP on mySQL MaxDB and low-cost blade servers (not a bad way to go for some, actually). So beyond acquisition costs, like one-time product and product installation costs, *recurring costs*, including the monthly cost of SAP operations, systems management, maintenance, and other day-after-day costs must be factored in as well. This mix of factors is illustrated in **Figure 1**. Note how the costs associated with people tend to rise over time, whereas technology costs rise and fall. It's also important to note that unless a process-oriented approach like the IT Infrastructure Library (ITIL) or IT Service Management (ITSM) is employed, process costs tend to remain flat over an SAP system's lifecycle.[4] The process costs between particular solutions may vary, though.

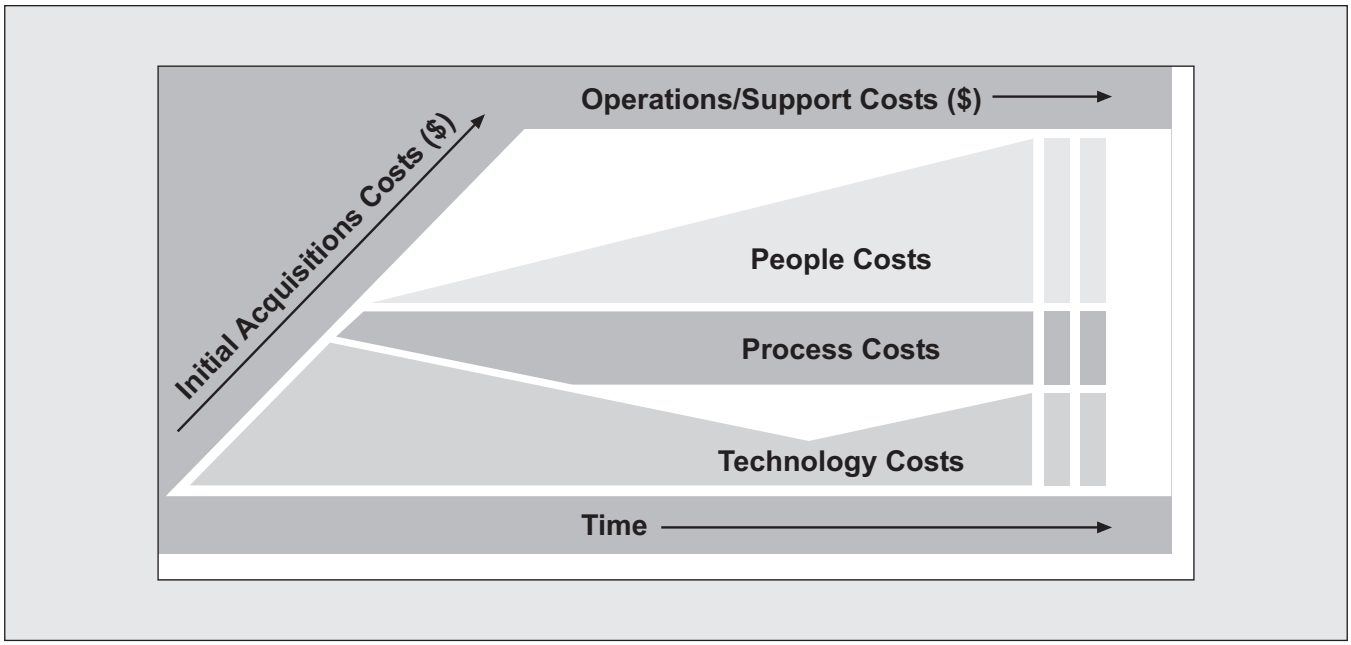
Let's now take a look at how to execute a delta TCO analysis to evaluate a potential migration.

Executing a Delta TCO Analysis

A delta TCO analysis is nothing more than a commonsense approach to evaluating two or more alternatives — in this case, SAP technology platforms. Instead of trying to quantify total costs across the entire spectrum of technologies, people, and processes, a delta analysis instead quantifies only the *differences*. For example, if Option A's server acquisition costs are $500,000 and Option B's server acquisition costs are

[4] For more on ITIL and ITSM, visit www.itil.org.

Figure 1 TCO Includes Initial and Ongoing Costs, and Spans Technology, People, and Process Areas



$200,000, then the delta is recorded as a $300,000 cost against Option A. That is, Option A costs $300,000 more than Option B. In the same way, all other technology costs may be evaluated and factored in — costs of the OS, database, platform-specific tools and utilities, specialized software overlays, maintenance and support, high availability and disaster recovery options, and so on. In the end, one alternative will reveal itself as less costly than the other. The key is to ensure that the alternatives are indeed *like-for-like* — that the same level of performance, availability, scalability, manageability, and other capabilities are reflected equitably within each alternative. Comparing a system that can host half a million SAPS (a common measurement of workload in the SAP world) against a system that can only host 100,000 SAPS makes no sense; "true up" the alternatives, so that an apples-to-apples comparison is indeed possible.

By way of example, consider one of my recent clients that found themselves in the midst of a mandatory technology refresh. Their plans to upgrade SAP R/3 to SAP ERP Central Component (SAP ECC) 5.0 necessitated quite a bit more system horsepower to service the same number of logged-in users than they currently had available. Thus they had the perfect opportunity to evaluate upgrading their current SAP technology platform or explore alternatives. They were a long-time Unix shop, running a popular Unix variant on a popular platform. Their IT team knew the environment well, but with only pockets of expertise in competing technology platforms, they were limited in breadth of support capabilities. A quick analysis of their current workloads extrapolated for SAP ECC 5.0 indicated the need for a system that supported 40,000 SAPS at a peak CPU utilization of 70% (implying an additional 30% CPU bandwidth to address future unknowns). The idea was to size a system with a lifespan of three years.

By working with the SAP Competency Centers for various hardware vendors (or in the absence of such in one case, by comparing published tpcc numbers between different vendors and finding something that looked suitable), my client assembled four different configurations from three different vendors. Each system should have been designed to host 40,000 SAPS. In one case, though, the system was greatly oversized to address unknown peak workloads never

imagined by the client. And in another case, the system was designed to support 40,000 SAPS but at 95% CPU utilization — again, not what was specified. One of the vendors factored in an expensive disaster recovery clustering solution, whereas the other two did not. One assumed the client owned a disk subsystem with the necessary disk space to house three years worth of growth; the others factored disk requirements into their configurations. Two of the vendors did not include operating system costs, and one included the costs of a bolt-on enterprise systems management package. Server and disk infrastructure setup costs were included in two. One even added incremental network gear into the mix.

By iteratively working with each vendor to tweak its configurations, we finally arrived at what appeared to be four apples-to-apples SAP system sizings. That's where the real fun began. Backing out the costs of all the "extras" revealed a wide discrepancy between core server, OS, and disk subsystem acquisition costs. Maintenance costs over the life of the systems varied even more. In the end, given a particular level of desired availability and performance, it was fairly easy to weigh the alternatives against one another by applying the delta TCO approach. In the same way, comparing the possible OS and RDBMS options supported atop the various hardware platforms provided visibility into the delta between each system's high-level three-year TCO relative to technology.

That was just the beginning. The delta analysis approach gets a bit trickier when it comes to people costs. Sure, there's plenty of data published by Gartner/META Group, IDC, Forrester, Aberdeen Group, and others that quantifies the average costs or differences between platform systems administrators, database administrators, data center environments, and so on. The real challenge falls back on what is likely to happen in your own situation, though. Very few IT shops consider dismissing all of their SAP support team after an OS/DB migration, for example — despite the fact that the experts tell you such a move makes financial sense. Instead, existing teams are augmented with subject matter experts (SMEs) while a small set of existing headcount are redeployed, or the existing headcount is given training in the hopes that they will get up to speed quickly on the new platform. So while certain platforms may appear to boast very low people costs, much of this very real cost savings is typically realized in *new* SAP deployments, not migration scenarios.

With all this in mind, be sure to capture the real costs for your unique situation. Factor in the likely scenarios that will play out, and do the math. In my own experience, moving to a Microsoft platform can offer significant savings in database administration, infrastructure specialists, and so on. The fact that many diverse skills normally found throughout an SAP IT organization chart are readily found in *fewer* full-time employees bodes well for a transition to Microsoft, too. But at the end of the day, if you're not prepared to make the hard decisions surrounding redeploying or retooling your Unix server, legacy database, disk subsystem, and other SAP technical support personnel, those savings will be elusive at best.

The last area for tracking the cost deltas between two or more different platforms lies in process-oriented costs. A nebulous term, typical processes evaluated for a delta analysis include operational processes like those regarding tape backup/restore, high availability and disaster recovery day-to-day operational and failover processes, costs associated with systems management processes unique to a particular platform or technology stack, and so on. Be careful to exclude the cost of people or technology here, though — it's tempting to start recording technology costs again, for example, when analyzing the cost of a particular high availability approach. Double-counting adds up quickly!

Process costs can vary dramatically between platforms, or be amazingly similar. In my experience, an organization's approach to disaster recovery (including the tools used, the tolerance for downtime in the wake of a disaster, and the need to fail back quickly to the original data center) can drive significant process deltas between different OS/DB target systems. And to what extent a company is willing either to ignore certain in-place IT standards or *enlarge* its standards base to embrace new standards affects process costs.

Think about it — the ability to leverage a company's current technology stack, tool, or management utility expertise makes doing so inherently less expensive than learning how to do something new, at least from a process perspective. New tools and therefore new approaches to managing and maintaining SAP mean more costs in the short term.

Once the technology, people, and process differences are outlined and quantified, it's tempting to simply add up all the numbers and select the least expensive alternative — not always the right answer, as you would certainly suspect. Other matters come into play, some easily quantifiable and some not so much. I recommend pausing at this point not only to get a second pair of eyes involved in reviewing the results of the delta analysis, but also to reflect on the hard dollar costs vs. the soft dollar savings that may be realized by a particular direction. This is detailed further next.

Direct vs. Indirect Costs

Although TCO analysis is all about determining how to obtain the best IT/business value for the least money, it's much more complex than simply determining the relationship between hardware costs and budgets. Specifically, there's great value to be had in evaluating the direct costs associated with a particular platform against the direct costs of another platform — the hard dollars that need to be spent. To an even greater degree, though, there's value in understanding the indirect or soft costs inherent to a particular platform and the support that must go into maintaining it.

In my own analysis over the years, I view the following as direct or "hard" costs:

- Server, disk subsystem, and other hardware-oriented acquisition costs
- Operating system and RDBMS license costs
- Software overlays associated with high availability, disk management, etc.
- Multiyear support and maintenance contracts on all hardware and software
- Employee and contractor salaries
- Systems management applications, tools, and utility costs
- Formal training costs
- Costs associated with *additional* time (outside of costs already captured via normal compensation) that is measured hourly or annually and is spent administering, managing, and maintaining the system; for example, overtime, incremental contracted help, occasional consulting expertise, and so on
- Easily measured *incremental* heating/cooling, power, or other utility costs that vary between different OS/DB platforms
- Any other resources that must be paid for — if you write a check for it once, or every month, it's a hard cost

These costs are easily quantifiable. Indirect or "soft" costs, on the other hand, are much more difficult to quantify and include items like the following:

- An estimate of the total cost of annual unplanned downtime for a particular platform relative to the impact felt by the end-user community (that is, the value of unproductive end-user time)
- The cost of downtime or disruption relative to the SAP IT team tasked with supporting the platform (the value of wasted IT time)
- Additional costs (relative to lost productivity) associated with on-the-job IT training
- The incremental costs of inefficient or redundant processes
- The incremental costs of ineffective tools (typically rectified by procuring, at additional expense, specialty tools that offer a certain amount of redundancy while filling in the existing gaps for which they were purchased)
- Incremental application development and integration costs inherent to a particular platform (where a lack of platform agility or lack of support for

open standards equates to incremental developer time, and therefore greater costs)

- Costs associated with deviations from existing IT or business standards
- Costs of *shared resources* that are not easily quantified (part-time or ad-hoc support related to network infrastructure, data center services, platform security resources, etc.)

As you can imagine, it is monumentally simpler to compare two platforms relative to their hard-cost differences — the dollars are naturally easier to identify and quantify. There is much value in analyzing only the hard-dollar differences — that is, with clearly defined costs and an understanding of their origin, a hard-dollar analysis is difficult to dispute. Yet common sense and a number of industry sources tell us that indirect costs will actually *exceed* direct costs over a period of three to five years. So, for better or worse, a delta analysis that takes both hard and soft costs into account is really the best way to go, assuming you have the time to pursue this path, and access to (or consensus around) specific data points. In this way, you can support the decision to continue evaluating a proposed migration relative to potential risks, which we look at next.

Weighing the Risks of a Proposed Migration

Risks are a part of life, and risks surrounding a fundamental change to your mission-critical SAP systems are no exception. Now is the time to evaluate whether or not a migration is truly worth everyone's time and trouble — well ahead of any investments being made.

Risk vs. Reward

Before you spend too much time uncovering costs and pushing hardware vendors and consulting firms to further sharpen their pencils, determine a target cost savings that makes an OS/DB project worthwhile. Will $2 million in direct cost savings over three years make everyone happy? Does it need to be $5 million to get the CFO's attention and subsequently the end-user community's default buy-in? In one case, one of my clients simply didn't believe that an expected $500,000 a year in hard savings was worth their time to retool, retrain, and learn to support a completely new technology stack. For others, though, this number might be more than enough to take the migration plunge. If you have an early idea as to your own organization's risk-vs.-reward threshold, you'll waste less time making a decision to pursue an OS/DB migration (or *any* changes with the intention of reducing costs, for that matter).

Consolidation and Simplification Considerations

A great number of migration projects involve customers attempting to reduce the number of OS and hardware platforms supported by IT. Many others tend to reflect strategic transitions to lower-cost platforms like those made possible by Microsoft Windows and RedHat or SuSe Linux — the same platforms used in other areas of IT. It's also commonplace to see IT shops seeking to escape expensive RDBMS licensing by moving to less expensive and often just as capable database platforms. The fact of the matter is a slew of SAP migrations today are driven by the need to support consolidation efforts while *simultaneously* increasing service levels and reducing costs.

But is the drive to consolidate information technology resources *outside* of the specialized team that supports SAP worth the trouble and expense? It's relatively easy to consolidate file and print servers, standalone database servers, Web servers, and so on, but does it make sense for the SAP IT team to try to do the same thing within their smaller circle of mission-critical systems? While rhetorical in nature, these questions bear answering sooner rather than later. You might expand your delta analysis, for example, to include the cost efficiencies made possible through leveraging IT consolidation lessons learned elsewhere in your organization. By the same token, you might seek to identify a high-level risk vs.

reward number in the process — if the savings don't exceed a particular number, or a proposed change negatively impacts standardization or otherwise invites complexity, these facts can quickly help you drive a financial stake in the ground.

Other Mitigating Factors

In the past, IT shops flocked from old releases of Microsoft SQL Server and expensive IBM DB2 variants to an assortment of Oracle and Informix releases in response to the need for better performance, greater scalability, and lower costs. The playing field is different today, but the pain points faced by IT have changed very little. This explains why there continues to be so much talk of migration. And this explains why these migrations tend to be complex, as companies seek to revisit both their OS and database choices as they simultaneously pursue simplification, consolidation, and other IT strategies. Does the benefit of SAP consolidation outweigh the risks, though?

To answer this question, take a closer look at the SLAs for which IT has signed itself up. What kind of availability has been promised to the business team? Are there not-to-exceed performance metrics in place (like average online transaction times or the time necessary to execute critical reports)? What kind of scalability does the business team expect? More important, what is IT being held responsible for? And what about security and ease of integration — are things pretty good in this regard today, or is it well past time for a change?

If your SAP systems are meeting all of your business and IT needs, you'll have that much more work in front of you to justify a migration. The risks will be greatest, so the rewards had better be similarly spectacular. On the other hand, if you've been inundated with problems, the good news is that the risk side of this equation will carry much less weight. Therefore the reward side will require less weight to tip the balance in favor of an OS/DB migration. Take a critical look at your system from these perspectives, weigh them against one another, and as I've said before, do the math.

Should an OS/DB migration appear warranted after working through the numbers and weighing the risks, there are still a few remaining key factors that must be considered — namely, the considerations involved in different migration methods and approaches. We'll look at these next.

Choosing a Method to Perform a Proposed Migration

There's more than one way to migrate a system, along with a corresponding number of potential challenges and benefits that represent key inflection points — points comprised of multiple related decision points. Analyzing and weighing these decision points directly affects the amount of downtime necessary to execute a migration as well as its inherent complexity; the method proposed to execute a migration is therefore important indeed. Although SAP publicly endorses its SAP OS/DB Migration Kit as the single best migration method, other approaches are possible. Let's turn our attention now to some of these different methods. This will help us build the groundwork for a better understanding of how to address important SAP OS/DB migration challenges or limitations that might lie ahead.

Using SAP Standard Migration Tools

In a nutshell, an SAP OS/DB migration boils down to dumping or *unloading* the contents of a customer's *source* SAP database into a platform-independent format, and then turning around and loading this data into a new target system. The SAP OS/DB Migration Kit is a set of SAP migration tools (essentially software utilities provided by SAP for a price) that takes care of this critical albeit tedious process.[5] R3LDCTL reads the ABAP Dictionary to extract database-

[5] For a detailed description of using the SAP OS/DB Migration Kit, see the *SAP Professional Journal* articles "Performing a Successful and Cost-Effective Migration of Your SAP System: Understanding the Export Process" (March/April 2004) and "Performing a Successful and Cost-Effective Migration of Your SAP System: Understanding the Import Process" (September/October 2004).

independent table and index structures, and then unloads these structures. R3SZCHK is used to compute the sizes of tables and indexes that must be set up in the target database, while R3LOAD is used to perform the actual unload and load of data between the old and new SAP systems.

Given the great number of database tables found within SAP, it's fortunate that multiple R3LOAD processes may be executed in parallel, which can significantly speed up an SAP OS/DB migration. Yet much work is normally required to make this happen (for example, the SAP migration tools require an SAP-generated migration key and an SAP-certified OS/DB migration consultant), representing one of the primary differentiators of other tools and approaches — speed of execution. The other primary differentiators are cost and complexity.

Using SAP System Copies

As mentioned previously, the most common type of data migration also happens to be the simplest — a *homogeneous system copy*, or simply *system copy*. A homogeneous system copy is routinely used for many purposes, probably central of which is to use a copy of the production database to refresh test or quality assurance and other non-production systems. You might also execute a homogeneous system copy to change the system ID of an instance or to move in or out of an MCOD (multiple components, one database) implementation. Given that neither an SAP migration key nor a certified SAP OS/DB migration consultant is required to execute a system copy, this kind of migration can be performed relatively quickly.[6]

Interestingly, with even a seemingly straightforward system copy, you might still run into a few quirky issues. If you're running a very old release of SAP R/3, for example, you might actually find yourself in the position of having to *upgrade* your source SAP R/3 system first, just to support the move to a new OS or database release. Most often, this is because a particular recently released database version might only be supported on the latest version of a particular OS. This could even necessitate a functional upgrade of SAP R/3 prior to doing the migration — a complicated project indeed to support an otherwise simple homogeneous system copy!

System copies can be quite fast. In the case of very large databases (VLDBs), though, like those exceeding 500GB or so, a system copy might not fit the bill. Similarly, if your downtime window for executing a migration is limited, another migration process is warranted. Some of these approaches are covered next.

Migrating Very Large Databases (VLDBs)

For quickly migrating the largest of production databases, hardware-based disk cloning (enabled by very specific disk subsystems offered from most top-tier storage vendors) can save the day — a clone of your database can be "snapped" and presented for migration activities in a matter of minutes. But this technology is expensive; if it's not already in place, it is unlikely to be deployed simply for a one-time migration. Thus a less-expensive approach is warranted. To this end, SAP offers the Incremental Migration (IMIG) tool. The IMIG tool splits the migration into online and offline components and makes it possible to migrate the largest databases tables while the source SAP system is still online and available to its end users. Afterward, the remaining smaller database tables are migrated in the standard way using the SAP OS/DB Migration Kit — with the source SAP system down and unavailable. Overall, this process can significantly reduce the total time necessary to migrate large databases between platforms.

Not surprisingly, there's a downside to using the IMIG tool. First, the migration project from start to finish will take longer to complete. By virtue of this

[6] For more on performing system copies, see the *SAP Professional Journal* articles "A Homogeneous System Copy in 60 Minutes? It Can Be Done!" (January/February 2002) and "Database System Copies Made Easy — A Guide for Copying an Entire R/3 System to a New Windows Platform" (May/June 2003).

extended timeline, it will naturally cost more as well. Next, the IMIG tool is not for public consumption — SAP themselves must install and execute this tool. This further adds to the cost of the project in that SAP will naturally charge a fee for these high-demand services.

Performing an SAP Migration with Third-Party Tools

Another method of migrating large SAP databases is to use third-party tools like those from Quest, Oracle, and other software and RDBMS vendors. It must be noted that although SAP will not officially sanction or support the use of third-party tools, they will support the migrated system. The key is to ensure that you communicate your plans to SAP and that the migrated system indeed works as expected, both functionally and technically. Without the safety net that SAP provides, you'll need to lean heavily on the third party for any support necessary during the migration process. To address and mitigate the risk of using a non-SAP approved migration method, you will also need to build additional time and possibly specialized resources (that is, expense) into your migration project plan. This will provide for any extra time necessary in the name of "problem or issue resolution."

Using SAP Partner Migration Tools

A number of SAP's hardware partners have developed proprietary migration tools useful in speeding up the migration of VLDBs or transitioning from one particular platform to another. HP, for example, last year developed the HP Smooth Migration Process for migrating VLDBs up to three times faster than an IMIG-driven or standard SAP migration. Developed originally for quickly migrating Tru64 Unix/Oracle systems over to HP-UX/Oracle, the HP Smooth Migration Process has been expanded to include other platforms like Sun Solaris, and products outside of SAP R/3 like SAP BW.

The primary benefit of using the HP Smooth Migration Process lies in its shorter migration time — not the time spent in evaluating and planning, but the time spent actually unloading and loading the database. This capability comes with a price, of course. Disk space for the unloaded database tends to run 20% of the total size of the source database, rather than the default 10%. And there are additional costs to using SAP partner migration tools. In the case of the HP tool, be sure to include the cost of an HP/SAP consulting resource trained in the process. And remember that such a resource does not replace the need for an SAP-certified migration consultant. Finally, keep in mind that unknown issues will need to be escalated to and addressed by the Global HP SAP Competency Center out of Walldorf, or (in the case of similar vendor-provided tools) the SAP Competency Center for the particular migration tool you are using. To be safe, also factor in extra testing and validation time into your migration project plan.

Choosing the Right Approach

It's very important to select the right migration approach for your particular needs. The key lies in understanding the specific downtime windows you have to execute the migration, coupled with your own budgetary and resource constraints and a clear understanding of all stakeholder needs. In the end, if all things are equal, lean toward employing the SAP OS/DB Migration Kit. Why? It works! And with plenty of success stories spanning countless technology platforms and nearly all of SAP's products, the risk of migration will be mitigated to a great extent.

Armed with the information you've gathered up to this point, and comfortable in the knowledge that you know where you are heading and that you understand the hard- and soft-dollar costs of the decision at hand, you are finally in a position to make an informed decision about whether to give the go-ahead on the migration or walk away.

Making the Go/No-Go Decision

With the direct and indirect numbers, risks, migration methods, and other factors in hand, preferably side-by-side in something akin to a Microsoft Excel spreadsheet, making a go/no-go decision should be pretty straightforward. After all, one option will invariably cost less than another. Still, the right answer at the end of the day should not be driven purely by finances. Even at a financial premium, a company may rightly choose to go with a more expensive platform for any of the non-cost-oriented reasons previously mentioned. That is, even though a delta analysis will highlight the relationship between the expected costs of various platforms (illuminating how quickly a particular solution might claim a return on investment), there's more to an SAP OS/DB migration than dollars.

As I have mentioned throughout this article, the technical migration process has been covered in detail in previously published *SAP Professional Journal* articles. There is no reason to belabor these details again here. Instead, I will leave you with some hard-won lessons learned on how to mitigate some of the risks involved in going live on a newly migrated platform.

Mitigating Go-Live Risks: Lessons Learned

Though going live on a newly migrated technology platform should be a "non-event" in the best of cases, such is not always the case. Matters sometimes conspire to create less-than-perfect transitions. Issues crop up from a variety of sources, running the gamut from inconsequential to inconvenient to all-night support marathons. The following are shared in the hopes that you'll benefit from my own lessons learned, along with those of my colleagues, customers, and acquaintances, smoothing out your own migration ruffles in the process.

Checklists, Timelines, and Easily Overlooked Tasks

I have always been a huge fan of simple paper-based checklists — they help me stay the course when things get complicated, or when I'd rather not struggle to remember something I need not remember long. In my experience, the following tasks are representative of such items. Add them to your own checklist of things to do sooner rather than later, keeping in mind that this list is by no means comprehensive but rather a reflection of matters that others have put off too long or completely overlooked until the last minute:

- ☑ Register the migration with SAP, a simple process necessary to obtain the standard SAP migration tools and thereby begin the technical process of migration, assuming you are going this route and employing the standard SAP OS/DB Migration Kit. If you are going with a third-party migration process, register with them as soon as possible, too. Once the migration is registered, waste no time ordering the requisite migration kit. Note that these tools are constantly updated and often improved upon. Be sure to include time in the project plan to acquire, install, review, analyze, and possibly update the migration kit. And finally, I recommend testing the migration tools in a technical sandbox or similar environment rather than using them for the first time in a development or other quasi-production environment.

- ☑ Engage your hardware vendor's SAP Competency Center early on, to begin the process of sizing/resizing as well as to help you confirm that the SAP application you are currently running is indeed supported in the proposed target environment. You might be surprised by what is not actually supported in production environments — better to learn this now!

- ☑ Begin the work of identifying all of your migration team members as soon as possible. Include the time it takes to find, bring in, and inculcate the requisite SAP migration partner and their certified migration consultants as well. Finding a person

with experience in both your source and target SAP systems is preferable, of course, but not always practical. (I'll go into more detail on building a migration team in the next section.)

- ☑ Assuming you are employing the SAP OS/DB Migration Kit, obtain a migration key for each system to be migrated. A preliminary discussion with SAP surrounding license keys is a good idea at this point in time. This is especially valid for large SAP system landscapes consisting of multiple SAP components — licensing can get more complicated than you expect.

- ☑ Early on, create the beginnings of a migration checklist consisting of the specific step-by-step instructions necessary to conduct a migration in your unique environment. Document how long each step takes to execute. And be prepared to update this document throughout the migration process (give yourself time to update the migration checklist after each migration; create a separate task in your project plan to hold the team accountable for maintaining this critical "how-to" document).

- ☑ Give yourself enough time to acquire and build out new development, test or quality assurance, staging, training, and other server and storage environments that reflect the target OS/DB platform. Beyond acquisition time, such work can literally consume weeks — don't hold off too long. While you're at it, set up the specific disk space necessary to stage the data dumps from each source system. To be safe, you'll want 20% to 30% of the disk space consumed by each database being migrated (10% is SAP's minimum; 30% gives you room for multiple iterations or multiple databases). And don't forget to request an adequate amount of downtime from each of the source system's end-user communities and responsible IT liaisons!

- ☑ Establish a "performance planning" task to verify the basic performance of the OS, DB, and SAP layers in the *source* system (optimized for reads, to speed up the migration); then use an identical task to verify the performance of the *target* system (optimized for writes initially, and later for the unique mix of reads/writes hosted by the system during normal operation). Systems optimized in this manner will speed the migration process considerably.

- ☑ Similar to the performance planning task, identify the largest database tables, those that are growing fastest, and any non-SAP-owned objects in the production database — all of these will complicate or present challenges relative to quickly or smoothly completing a migration; the sooner they are identified, the better chance you have of working through issues.

- ☑ To save significant time, consider skipping the formal OS/DB migration process in the case of most non-production environments, and instead perform a homogeneous system copy of the production system once it's migrated.

- ☑ Conduct a formal functional check of the migrated system prior to turning it over to its end user community.

- ☑ Do *not* touch the disaster recovery system until your final production migration is complete. After verifying the production system is operational and performs well, then go ahead and upgrade the disaster recovery system to reflect your new technology stack.

- ☑ Along the same lines as the performance planning tasks, establish a thorough "load testing" task to verify the performance of the OS, DB, and SAP layers in the target system *under load*. This represents a project in its own right, and it will need to include the acquisition and configuration of a load testing tool, resources familiar with using the tool, and the time necessary to script all of your representative selected business processes (more on load testing in an upcoming section).

- ☑ If a functional upgrade to SAP is planned, ensure at least six weeks passes between the final migration of the SAP system to a new OS/DB platform and any functional upgrades (such as upgrading SAP R/3 4.6C to SAP R/3 Enterprise or SAP ECC 5.0, for instance). Don't be tempted to conduct both technology and application-layer refreshes at the same time. Aside from the obvious issue of risk, there's the matter of resource constraints — your people can only do so much.

With the high-level tasks behind us, let's turn our attention to the tasks that make or break an OS/DB migration. These include building the migration team, drafting a detailed pre-migration checklist, and looking into the specific functional and performance-oriented challenges inherent to your new SAP system.

Tips for Building the Migration Team

An SAP migration project team does not need to be huge. Many small shops get away with assigning only a project lead and a technical lead, augmented with part-time specialists as the project plan dictates. But at the end of the day, there's great value in formally assigning team members to cover key project tasks as well as each layer in the source and target technology stacks. Such an approach naturally mitigates risk while encouraging ownership of the project. Thus expertise in all of the following disciplines should be reflected in your migration resource plan, keeping in mind that carving out only a portion of time from your full-time employees' workday fits the bill in many cases:

- ☑ Project leadership/management
- ☑ SAP OS/DB migration process (i.e., the SAP-certified migration consultant, or a third-party consultant experienced in the particular migration process you've selected)
- ☑ SAP Basis/SAP Web Application Server technology platform
- ☑ Source server platform
- ☑ Source operating system
- ☑ Source RDBMS
- ☑ Source disk subsystem/storage
- ☑ Target server platform (if different from source)
- ☑ Target operating system (if different from source)
- ☑ Target RDBMS (if different from source)
- ☑ Target disk subsystem/storage (if different from source)
- ☑ Network and other data center infrastructure
- ☑ High availability and disaster recovery

Project leadership in particular should be addressed by a seasoned project manager, if at all possible one knowledgeable in SAP technology projects. This person must be comfortable juggling and prioritizing work and issues that may span various technologies, business groups, and even IT organizations. And the project manager should preferably be an "insider" as opposed to an outside consultant or other contracted resource.

Expertise in the OS/DB migration process is provided via an SAP-certified migration consultant (or a similarly experienced/certified third-party consultant, if you are going that route). This is perhaps the key role within the core team; he or she is responsible for technical leadership throughout the migration process. Meanwhile, expertise in both the source and target systems is provided via in-house and perhaps contracted resources (especially in the case of target system support). Access to an SAP Basis consultant knowledgeable in both source and target systems will prove critical to the migration, too. Depending on the size, complexity, and mission-critical nature of the database being migrated, the core team should include

high availability, disaster recovery, and disk subsystem experts as well. Inclusion of other technology infrastructure expertise can prove important if not just plain convenient, especially in cases where new data center and network infrastructure is deployed in support of the target system. Finally, arrange for post-migration support from experts in the new SAP technology platform — something along the lines of one to four weeks is common.

Pre-Migration Tasks

While the tasks mentioned earlier apply to the entire migration process, there are a number of tasks that apply specifically to the production system (though testing in a non-production environment is prudent as always). Prior to beginning a production migration, ensure you've considered, included, and tested each of the following — doing otherwise will only cost you even *more* time when you least have such time at your disposal:

- ☑ Specifically for the production environment, set aside the time necessary to test *how* the data in the database will be transferred between the source and target systems (via tape, a high-speed LAN connection via FTP, configuration of a temporary gigabit segment between the source and target systems, use of removable storage system drives or laptop drives, etc.). Test the process!

- ☑ Search for non-standard ABAP code (code developed in-house or by a third party, like that used by many enterprise systems management tools) and database entries. It's especially important to identify ABAP code that executes direct operating system or database calls, as these programs must be noted, changed, and specially tested during the migration.

- ☑ In the same way, run SAP transaction DB02 (Analyze Tables and Indexes) and check for missing tables or indexes; rectify any issues prior to even *attempting* a migration. And seek out those largest database tables early in the migration process. Given that the time needed to complete an OS/DB migration is dependent on the size of the largest database table, look for ways to reduce this table size; for example, shorten the migration time by archiving data from within the largest database tables.

- ☑ Pay particular attention to systems interfaces. Identify the owner of each, determine the transport method and any problematic hard-coded database or OS calls, and then thoroughly test each interface in the new system.

- ☑ Delete existing database backup jobs, all "test" clients, and other useless items occupying space in the source system to speed up the export process. And so they do not attempt to execute in the new system when it's started for the first time, be sure to "de-schedule" all SAP jobs in the source system as well.

- ☑ Thoroughly test operational tools such as backup and recovery systems, RDBMS utilities, systems management applications, and so on. I sometimes refer to this process as an "Operational Proof of Concept" (or Operational POC); include testing of all non-SAP components such as bolt-ons and archiving solutions as well.

In addition to these tasks, it's prudent to *consider* upgrading the OS or the RDBMS as part of an OS/DB migration, even if the RDBMS is not otherwise being changed. For example, taking this opportunity to upgrade from Oracle 9i to 10G or Microsoft SQL Server 2000 to 2005 can save a lot of incremental planned downtime in the near future. True, there's more risk involved in doing so. Weigh the risks, assign a dollar amount to upgrading now vs. waiting until later (evaluate the cost of downtime, for example), and make your decision.

Tips for Executing the Migration

Because time is of the essence, anything that can be done to speed up either the unload or load processes

adds instant value to an SAP migration. HP Consulting and Integration Services have used a number of processes and practices to return precious time back to our customer's IT organizations. I recommend executing time-consuming unload steps *before* actually starting a migration, for example, and speeding up the process through the following:

- ☑ Save time and execute "update statistics" early, or skip it altogether if the database's growth is predictable and stable.
- ☑ Compute all table and index sizes early in the migration process.
- ☑ Once the migration has commenced, copy the completed export files as they finish, instead of waiting until the very end of the unload process to copy everything over to the target's staging areas.
- ☑ If a network is to be used to move the exports between source and target systems, set up a dedicated gigabit LAN segment instead of using your public network. This is especially useful when migrating large databases, because most public LANs are quickly overrun when transferring large dump files.
- ☑ If they are used in the copy/move process, turn off hardware compression on any tape drives (given that SAP's native compression is both better and faster).
- ☑ Instead of special networks or tape processes, consider using high-capacity laptop disk drives to move data quickly between source and target systems.

Conducting a Final Functional Check

Walking through key transactions to ensure they work as expected is the goal of the final functional check. This ensures that the migrated system can functionally support the business as well as the source system. For the best results, functional testing should be conducted using an automated functional/regression test tool like those from the likes of AutoTester, Compuware, Mercury, or Segue. Ensure the regression testing experts within each functional area decide what to test, how to test, and so on — hold them accountable, rather than allowing this testing to be conducted by another party with less at stake. And remember to not only document each test case, but obtain sign-off that key transactions deliver as expected. Finally, by all means, if inconsistencies are encountered, halt the SAP migration process until any issues are resolved. Too much is at stake to do otherwise.

Addressing High Availability and Disaster Recovery

An OS/DB migration is not complete without verifying that any specially configured target high availability and disaster recovery offerings indeed operate as expected. Thus, prior to turning the system back over to its end-user community, a comprehensive failover test is in order. For the most risk-averse of customers, such a test conducted within the umbrella of load testing is appropriate. After all, if a failover test works as expected under your emulated real-world load — and your users and their business processes fail over and continue to work well after an emergency — much of the risk surrounding the availability of your new "unknown" target platform vanishes.

Housekeeping and Other Post-Migration Tasks

Once the SAP OS/DB migration has technically completed successfully, additional steps must be performed before the system is finally turned over to its users. Be sure to create a comprehensive checklist of these housekeeping matters, the sum execution of which may constitute a significant amount of time or cause a significant amount or trouble in their absence. For example, the SAP Transport Management System (TMS) will need to be reinitialized and reconfigured, as will all remote function call (RFC) connections and

other interfaces to external systems. And all batch jobs will need to be re-created, old SAP system statistics deleted, and more.

Finally, in the case of new OS or database platforms, remember that administrative, or "utility," scripts may no longer work as they once did, OS administration utilities will likely differ, and so on. Backup/restore tools and systems management packages may differ considerably as well. These issues should be uncovered and addressed during the test phase of each migration, of course, but it's prudent to verify one last time that the migrated production system is operational and indeed seems to meet key stakeholder expectations.

Optimizing SAP Performance

You will also want to conduct a reasonable amount of your own performance testing prior to turning the system back over. This should include running a standard suite of reports, daily transactions, batch jobs, and other jobs that validate external systems communication, overall system availability, and end-to-end business process performance.[7]

It is important to optimize not only the target system's SAP application layer, but also the rest of the technology stack underneath SAP. A poorly configured server/OS or disk subsystem, for instance, can make troubleshooting performance issues quite difficult, especially when the target system represents a newly supported platform. I recommend obtaining guidance from the various hardware and software vendors that make up your target technology stack — leverage their implementation and configuration best practices, or better yet their consulting services, to mitigate much of the risk of performance issues.

[7] For more on performance testing, see the *SAP Professional Journal* articles "Performance Testing Your SAP System — Best Practices for Preparing Your Test Environment" (January/February 2004) and "Performance Testing Your SAP System — Strategies for Ensuring a Successful Test Execution" (March/April 2004).

Load Testing — The Final Stamp of Approval

Beyond performance optimization, as indicated earlier, quite a bit of work must be completed prior to turning over the newly migrated system to its users. Each one of the technology stack changes represented by the new target system conspires together to create a great unknown. Thus, to truly mitigate the risk associated with all of this change, it is highly recommended to conduct a workload test that reflects your daily user load, along with a peak-load stress test that represents your worst expected hour. For example, a peak-load stress test might seek to emulate the 1,000 users that are typically connected to the system doing work, along with perhaps the incremental 100 users that tend to also work during quarter-end close. And to really simulate your worst hour, you might consider tossing in a number of key long-running reports or batch processes and other workloads that separately pose little risk, but combined can bring a poorly tuned system to its knees. The idea here is simple — give yourself a chance to identify performance or scalability issues, and then iteratively address these issues through tuning and tweaking *before* your end users ever even have a chance to complain of performance problems.

Celebrate!

By any measure, simply determining whether an SAP OS/DB migration is warranted in your particular case can be a lot of work — nearly as much work as the actual migration itself! With this work comes a corresponding amount of risk buried in complexities and challenges stemming from many areas. It is the complexities and challenges behind what can amount to a very involved project that drove me to draft this article in the first place. Twists and turns abound that can make for some very long days for those without a clear understanding of the intricacies of, and task dependencies within, an OS/DB migration. The technical goal of a migration is simple — to complete the migration of data and business processes within the

allotted downtime window, without sacrificing data and business process integrity. But the business goals may be quite a bit more complex. Starting off on the right foot is therefore essential to an on-time, on-budget project that meets the needs of all the company's stakeholders. Staying on track through the employment of a well-thought-out and comprehensive migration project plan is also essential. After the dust has settled, don't forget the most important thing of all — to celebrate your success with colleagues, friends, and family alike. Congratulations!

George W. Anderson is a senior SAP Basis Consultant & Project Manager for HP's Consulting & Integration Services organization, where he specializes in implementing and optimizing HP and other technology vendor platforms for SAP customers. An eight-year veteran of numerous successful implementations, platform and functional upgrades, and performance-oriented SAP engagements, George is a certified SAP Technical Consultant and PMI PMP. A frequent guest speaker or participant at conferences and customer-focused marketing events, George can often be found hosting Webinars, technical roundtables, and partner-driven events for the likes of SAP, Microsoft, Citrix, Segue, Compuware, HP, and other industry giants and thought-leaders. George is the author of SAP Planning: Best Practices in Implementation (2003) and most recently mySAP Toolbag for Performance Tuning and Stress Testing (2004), along with a host of magazine articles and HP internal knowledge briefs. He enjoys learning as much as he does sharing what he has learned; in between responsibilities at home, church, and work, George is currently pursuing a doctorate in Business Administration. You can reach George at george.anderson@hp.com.

