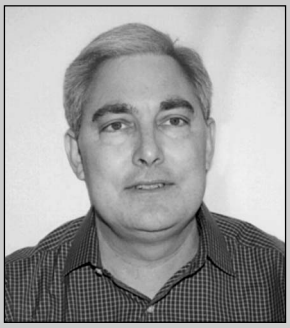
The 15 Most Overlooked Items in Planning for High Availability and Disaster Recovery

Kurt Bishop



Kurt Bishop is a retired member of SAP America's Technical Consulting Team. He has experience in both application and technical support of the R/3 system, where he provided a variety of consulting and support functions for customers and colleagues. Kurt focused on capacity planning, working to standardize and improve the process for both customers and vendors.

(complete bio appears on page 78)

In "Is Your R/3 System Recovery Plan a Disaster? A Three-Step Approach for Designing Recovery and Availability Plans" (September/October 2001), I offered a simplified definition of high availability, or failover, as "a quick recovery" and recommended that, since recovering from a disaster is the logical extension of a normal recovery, your planning process for the two should be the same.

By following the three-step plan laid out in that article[1], you can establish an overarching, high-level recovery plan that lays the basis for a more detailed plan, complete with step-by-step recovery instructions.

Careful planning and design of a reliable and recoverable SAP production environment will take a lot of your time and effort. It's a process that requires quantifying true business needs, documenting all your support and availability options and associated costs, and performing a risk analysis to define appropriate recovery standards for each and every business and/or disaster situation you can think of.

My best advice as you embark on this endeavor? Keep an open mind and don't overlook the obvious. Over the years, I have noticed the firms that are unsuccessful in their SAP disaster recovery and high availability planning tend to ignore the same key business and technical issues. My goal in this article is to set you up for success by sharing with you the 15 most overlooked items and discussing how to incorporate them in your planning:

[1] That three-step plan is summarized in the sidebar on page 61.

1. Know the cost of your downtime.
2. Consider security issues.
3. Avoid self-inflicted wounds.
4. Plan for non-fatal threats requiring system restarts.
5. Size the R/3 recovery system properly.
6. Perform regular and complete testing of backups.
7. Ensure availability of phone lines.
8. Plan for rebuilding network resources.
9. Ensure the disk space needed to recover is available.
10. Document how to deal with corrupt backup files.
11. Ensure temporary support resources are available.
12. Plan for loss of personnel.
13. Consider the loss of business associates' systems.
14. Assign responsibility for verifying results.
15. Review your recovery plan regularly.

Item 1: Know the Cost of Your Downtime

I am constantly amazed at the number of firms that have not documented the cost of their SAP downtime. Everyone agrees SAP downtime is expensive and should be addressed, but few put serious time and effort into identifying the real losses and long-term costs their organization may suffer.

The process of identifying the true effects of SAP downtime is so eye-opening that the firms that undertake it usually end up with full-blown recovery plans. Some firms are so impressed with the numbers they are willing to allocate nearly unlimited budgets for backup and recovery; some even establish offsite bunkers with facilities nicer than many primary environments!

I recommend you start identifying and analyzing your organization's SAP downtime and related losses as soon as possible. Then set a value on your company's information network and decide how much to spend on recovery and availability for these items. Look for tangible and intangible expenses and consequences of impaired systems as you establish the following types of costs related to downtime:

☑ **The cost of employee downtime:** Many employees cannot, and most will not, move on to other productive activities while the SAP system is down. So, for every user normally logged on to your SAP system, figure the cost of that employee's salary, benefits, and so forth to be a complete loss of money during any SAP system failures or serious performance lags. In most situations, you can double the time the system is down to figure the actual time lost. This will account for the time employees must spend to finish whatever they are doing, confirm the system is completely reliable, review their previous work to determine what was lost, and basically return to a productive mindset. To account for recovery efforts that are only partially successful, you must add the time employees need to redo whatever work was lost.

Take downtime and poor performance seriously and attack it with full force to ensure it never occurs again. I remember working on an R/3 installation for one company where a two-hour lunch had become standard fare. When I asked why everyone took so long for lunch, they explained that the system we were going to replace was so poorly designed and maintained that you could set your watch by its daily crash. As the employees began working during a typical day, the demands on the system would grow to a peak workload around 11:00 AM. The system would begin to bog down, eventually working itself into a performance deadlock of one kind or another, and the support staff would completely shut things down to get a fresh start over the lunch period. This happened so often that most people simply went to lunch early and returned late. They knew the system would crash between 11:00 and 11:30 AM and restoration would not be complete until 12:30 or 1:00 PM.

The Three Fundamental Steps of Recovery Planning

My article "Is Your R/3 System Recovery Plan a Disaster? A Three-Step Approach for Designing Recovery and Availability Plans" (published in the September/October 2001 issue) walks you through a three-step approach to establishing the groundwork for a system recovery plan. This refresher highlights the essentials of these three steps.

Step 1: Confirm the Value of Each Business Function

Start your disaster recovery and high availability planning by analyzing the business functions performed with your SAP system and documenting baseline requirements for their support. For each business activity, list the key functions and related data your end users must have in order to perform their day-to-day tasks. Next, add items users need, but are not time-critical, and finish the list with items that are nice to have, but would not stop the users from completing their core business tasks.

For each function supported by the system, list the cost of downtime and hardships incurred when the system is not available or performance is below target levels. Documentation of acceptable levels of performance, system access, and availability for each function allow you to rank every system function in terms of worth to the company and to ascertain the true order of priority for restoration.

This is the planning phase in which you perform your risk analysis — determining the likelihood of various disaster scenarios and the relative degree of loss associated with each one.

Step 2: Document the Technical Recovery Process

Technical recovery planning is fairly simple once the *business* recovery plan is in place.

In this phase, you need to determine the resources needed to satisfy the stated business goals for system access and performance. Develop a budget to supply and maintain the resources required. Add the supporting technical details to afford the users the "best system available for the money they are willing to spend."

Step 3: Invest in the Most Appropriate Plan

Negotiate with management for the most appropriate budget based on risk management and realistic evaluation of applicable recovery strategies.

Design and implement the best system you can with the resources available. Start with duplication of vital equipment. Assume that all systems will fail and need to be restored. Work your way toward the "nice-to-have" options. The business will define what the technician needs to do. Your downtime costs will define how much money you should spend on recovery and availability.

The table on the next page is an example ranking of business functions, listing the minimum information you will need in order to address their importance to the business and relative priority. You will most likely want to include more commentary in your list and also add more details (or change the relative rankings) as you do your technical planning (for example, you can note contact names and phone numbers for responsible users, support personnel and their areas of responsibility, and any special notes that may help the support staff). Details of how systems interface with each other and important procedures to follow in the how-to portion of your recovery plan can be stored here or elsewhere with reference to/from this list.

(continued on next page)

(continued from previous page)

Remember to update this list any time the business requirements change or you feel there is a change in the importance of particular functions in your organization. When using R/3, some business functions with varying degrees of importance may be supported by the same programs or system functions. In situations where there may be several rankings that apply, I suggest you apply the highest rank for any of the business functions involved. If this occurs several times, apply weighted averages or other mathematical techniques to split the business functions and provide a finite ranking.

Rank	Description	Comments
1	Order Entry (VA01, VA02, VA03, VA11, VA12, VA13, VA21, VA22, VA23)	Customers and their orders/revenue are the reason we are in business.
2	Delivery Ticket Generation and Order Processing (VL01, VL02, VL03, VF01, VF02, VF03)	No sense in taking orders if we can't deliver the goods in a timely manner.
3	Accounts Receivable and Credit Processing (FBE1, FBE2, FBE6, FBL4, FD11, FB10, F-22, F-64, F-26, F-27, F-28, F-67)	We have to receive and process checks to have enough money to generate payroll. We can't afford to extend credit to anyone who doesn't have money.
4	Payroll Check Processing (CAT2, CAT3, CAT4, CAPP, CAP5)	Employees are the number one asset and must have money to continue working. They must be able to support their families or they will leave the company.
5	Production Planning (CO01, CO02, CO03, CO05, CO07, CO08, CO10, CO40, CO41, ZZ55, ZZ56, ZZ59)	We have to be able to manage all of our orders and continue production of our products.
6	Inventory Management (MB01, MB02, MB03, MB1A, MB11, MB1B, ME00, ME22, ME23, ZZ81, ZZ82)	Because we are continuing production and distribution of products, we must be able to maintain proper inventories to avoid confusion.
7	Materials Management (MM01, MM02, MM03, IQ01, IQ02, IQ03, IW31, IW32, IW33)	We continually create custom products and must be able to create the bill of lading, materials, etc. to produce them.
8	General Ledger Posting (FBV2, FBV4, F-02, F-03, F-04, F-06, F-07, F-65, FBVO)	As we continue our business we must be able to keep a minimum of books and records. If disaster occurs during, or extends into, the closing period, these reports must be generated to satisfy government requirements or our business will be shut down.
9	Cost Reporting & Accounting (KB21, KB23, KB24, KO88, KE91, KE30, KSV5)	Cost reporting is a vital part of general accounting and we must continue to maintain an efficient and profitable operation.
10	Personnel Processing (PA20, PA30, PA40)	It takes employees to support our customers and we are continually hiring to maintain production levels.

How many customers or orders do you suppose this company lost during these extended lunch breaks? Can you imagine what would happen if your web site was always down for a couple of hours during the busiest part of the day?

Once habits like these develop, a company can simply spiral down until it ultimately fails or is swallowed by its competitors. These are sure signs that employees have lost their spirit and creativity. Instead of finding other things to do, they simply write off the time, and their jobs, as unproductive and a waste of effort. So imagine the "lost-opportunity" costs companies like this incur. I'm not sure how you can calculate that figure in finite terms, but I do know your key defense weapon is a reliable system that affords employees every opportunity to be productive.

☑ **The cost of unreliable central systems:** I have consulted with customers whose habits included extravagant personal backup systems. Some of the most sophisticated PC-based systems I have ever seen were developed by productive and ingenious individuals who worked with some incredibly bad centralized systems. Knowing they couldn't trust the central system to protect their data from loss, these dedicated employees stored complete copies of their important files on their own PCs. Just imagine the time lost in designing, implementing, and using duplicate systems of this nature on a daily basis. The value of customer files and related sensitive data stored on unsecured PC systems like these can easily justify almost any reworking of the central systems.

I have also seen volumes and volumes of disk drives being wasted when technicians have little faith in their systems. Once fear of failure creeps in, end users begin to rat-hole every piece of data ever used, and technicians begin keeping far too many generations of backup files on disk, tape, and related media. The result is a database growing at exponential rates.

☑ **The cost of lost business:** System access and availability make a strong statement about your business. What kind of impression will potential customers get of your business when they access, or attempt to access, your computer system for the first time? Remember how important that first image is. Imagine you are in a strange town, driving down the street looking for a quick meal on your way to a basketball tournament. Would you stop at the place with the dirty parking lot, hard-to-read signs, lights that may or may not work, and refrigerators that work only part time?

When you approach recovery and availability as risk management, then you might as well include all risks. Security is one of your biggest risks, and is so interrelated with recovery and availability, you can't separate the discussions. I won't expand this article into security planning, but the items in the following section will get you off and running in areas that need to be considered.

Item 2: Consider Security Issues

In the area of security, I have seen all of the following items left out of the planning process (some of these may seem like a stretch from recovery planning for now, but they sure won't seem so far-fetched when the system is down):

☑ **Security ABCs:** When you consider the security of information, start at the lowest level. I've seen a lot of firms with the most sophisticated online access systems in the world who had employees leaving reports unsecured in office "piling" systems or floating freely during the process of distribution from the computer center. This sounds innocent enough, but system access from outsiders is a disaster waiting to happen. And I have heard tales of true information espionage where minor disasters were created to divert attention as the crooks accessed the system and files. But the most egregious acts I have witnessed occurred when data was simply downloaded to personal computers and left unattended on hard drives and other devices, and available to anyone on the prowl.

☑ **Audit trails:** I believe nothing can be made

completely safe without a few unwieldy rules and regulations that would disrupt the normal course of business. During a true disaster recovery, you will normally have to compromise some of your precautions. Your key to SAP security, as well as recovery, lies in a well-documented audit trail for all system and file accesses. A solid audit trail will allow you to review the entire process and decide whether someone was simply working in the best interests of system recovery or trying to take advantage of a bad situation. You can't always stop a thief, but you can certainly set up strong deterrents by giving yourself a means to track him down and ensure that he gets his just reward.

☑ **Special IDs:** Depending on the size and complexity of your system, you may need several special IDs with any number of access patterns. I've seen some SAP shops that had a user ID with wide-open access to be used only during emergency situations. The user ID and password were stored in a sealed envelope and kept in a secure place. When the time came, you broke the seal and did what you had to do, similar to breaking the glass to access a fire extinguisher. The main issue with this strategy is controlling physical access to the envelope/user ID.

> ✓ ***Tip***
>
> *Consider using a special logon ID that has appropriate access for whoever is on call. At the end of the coverage period, the person on call changes the password and forwards the information to whoever is on call for the following period. This way, only one or two people ever have access to important information. My team employed this trick with good results. We also limited the access of the IDs so that the on-call person had the ability to create files, but not modify or delete them. Combined with complete recovery procedures that included a few extra copies of any file created, and strict monitoring and follow-up procedures, we had a complete audit trail of what took place.*

☑ **Security policies:** Consider incorporating your entire system access and security protection policies within the policies and procedures for recovery and availability. Because security policies follow the same general rules for determining the relative importance and ranking of systems that must be protected, and because disaster recovery, high availability, and security are topics that are hard to analyze in isolation, many companies will encapsulate their security policies in a chapter or two of their disaster recovery manual. Having a single manual for all key decision processes keeps things simple.

On a related note: Security policies are well and good, but what happens when the technical staff can't create a vital file in the middle of the night because of security standards? A key component of every security policy discussion, for every application, should be what to do during recovery efforts.

☑ **Access to security-related equipment and components:** Don't forget all of the system components that go along with the SAP information being secured. Some items to think about include special check-writing printers or signature generators, checks or printable forms, backup drives, tapes, and any reports or listings with information on system access methods. As soon as you figure out how to lock up your check-generating machine, figure out how to gain access to it during a disaster. And then figure out what you will do if the machine is destroyed in the disaster. Or what if you simply lose the key?

☑ **Virus protection:** In this age of the Internet and personal computers, viruses pose threats to any system's integrity and ability to continue processing. Your SAP high availability/disaster recovery plan will need a special section on how to recover from a virus, including exhaustive diagnostic procedures to determine the amount of damage suffered.

If a virus goes unnoticed long enough, you may be facing a true disaster that requires complete replacement of hardware and software as the safest recovery

alternative. Step-by-step documentation of recovery steps is a prudent defense mechanism.

Item 3: Avoid Self-Inflicted Wounds

Most planners are great at thinking up examples of disasters and security violations, but what about "stupid technical tricks"? We all hate to admit it, but most serious damage done to SAP systems occurs when we attempt to implement changes and upgrades. In fact, I consider internally introduced change to be the most common threat to any and all systems. In today's world of shrink-wrapped software, some of the worst errors are related to upgrades delivered to us by a third-party vendor. Or, how about that simple code change you made that will take place tonight, just prior to the nightly order processing, which runs on a very tight timetable?

Your system is at risk from internal changes 24 hours a day. To protect yourself from these types of self-inflicted wounds, you need to ensure you always know how to back out of changes and return to a known state of operation. I recommend having a standard configuration and business production schedule that is absolutely guaranteed to perform all essential business functions. The base configuration may not have all of the most recently requested bells and whistles, but it should have the most reliable and proven setup available that you can always fall back on, no matter what changes are introduced.

I also recommend dedicating a chapter in your disaster recovery manual to change management policies and procedures. Include extensive testing (Item 6) and signoff protocols with detailed instructions that ensure change and upgrade implementations don't affect SAP production processing. Absolutely no unproven code should be implemented in a productive system. Should something ultimately fail, extensive testing and signoff requirements will ensure an audit trail to assist analysts attempting to rebuild your SAP system. Similar to security, often the best you can hope for is a trail of crumbs to follow after the change has taken place.

Unfortunately, we all know even the best testing procedures will still allow an occasional bad piece of code into the system. The key to system availability will be how quickly you can recognize the problem and back out the offending code. More often than not, your best solution will be to recover the system to a state just prior to the code change. Then you can analyze the problem in a more normal environment with fewer time pressures.

Item 4: Plan for Non-Fatal Threats Requiring System Restarts

Performance problems are a good example of items that usually won't kill the system, but can sure make life painful. In my experience, performance and tuning for R/3 systems is one of the largest black holes for time and money in the technical arena. The replacement or reconfiguration of system components always seems to end with the requirement that the system be stopped and then restarted. That's why I recommend your recovery planning includes how to deal with system restarts in the middle of your busiest day.

> ✓ ***Note!***
>
> *If you take the approach advocated in my previous article — which is to view recovery and availability as risk management — then you might as well include all risks. And having one, and only one, comprehensive plan sure makes things easier when problems arise. You'll find steps for analyzing risks and recommendations on developing appropriate plans in the sidebar on the next page.*

Risk Analysis

How, exactly, do you analyze risks and, once identified, what do you then do about them? Follow these five steps.

Step 1: Identify Pertinent Threats

To identify any and all threats to your system, begin with a brainstorming session, and be sure to follow the most important rule for brainstorming — no idea is too wild or obscure to be included. I suggest you generate two lists, one composed of items that can render your system useless and one that identifies items that can hurt, but will not necessarily kill the system.

As you list the threats, include the source and any factors that may contribute to the problem. For example, if your computer center is located on the fourth floor of a building in an area that is free from rivers, drainage, and other common water-related threats from the outside, you don't have to worry about dealing with floods from external sources and can safely ignore any need for pumps, waterproofing, or flood recovery. But what about internal floods from broken water pipes? Suppose a water pipe in the ceiling breaks and leaks water on your database server. Or suppose someone spills a cold drink on the console. Where are the spare keyboards stored? (You do have a spare, don't you?) Will the operator have to reboot the system when the keyboard is replaced? Although these things may sound strange and trivial, recovery planning requires you to consider absolutely every conceivable threat.

I recommend reviewing the "Availability and Recovery Planning Checklist for R/3" in my previous article (September/October 2001) for a list of important hardware and software items in the R/3 environment, and some tips on the replication recommendations for each. Identifying these items and the risks associated with them will allow you to reduce your overall risk, build better recovery systems, and thus have a more reliable production system.

Step 2: Assess Vulnerable System Components

Go back through your list of potential threats and analyze each item to determine the extent of the threats identified. For example, when considering the threat of internal floods, list all possible components exposed to water, what effect exposure will have, how to limit exposure, and what to do when flooding occurs.

Step 3: Calculate the Probability of a Loss

This may be your toughest job, since many components will have multiple potential threats. Vendors and third-party firms can provide statistics on component failure for almost every item sold. Some of these statistics are a little hard to interpret, but they generally associate a certain number of hours of operation with the likelihood of failure. Unless noted otherwise, assume most vendor testing occurs in pristine laboratory environments … and adjust accordingly. If possible, find out whether their systems run nonstop or have a chance to "cool off" between tests. Any other key items that would make their tests less applicable to your environment are also important to note. Most vendors are not devious; nor are they out to "sneak one by you." Proving the reliability of system components that will be implemented in a variety of environments and asked to perform any number of tasks is hard to do.

Most vendor statistics are quite reliable and should be part of your analysis and planning sessions, but you must ensure you understand the conditions under which they were generated.

For threats from internal or external changes, or any item that requires change to program code or software, you should consider the probability of loss to be 100 percent.

As for tornadoes, hurricanes, and other acts of nature, your guess is as good as mine (though you might try studying weather statistics and charts).

When in doubt, I suggest relying on Murphy's Law ("anything that can go wrong, will").

Step 4: Determine the Effects of Loss

For most items, determining the effects of loss should be easy. List the most probable effects of a loss in simple, generic terms and don't worry too much about the minute technical details. However, you should have lots of information to back up your losses in quantitative terms in separate documents, such as recovery planning manuals. If you prefer to have as few manuals as possible, keep detailed information in this document.

For example, if you lose a database server, your entire R/3 system will be rendered useless. You can list the effect of the loss in these simple terms, along with approximate costs associated with the losses.

The key factor to consider in determination of cost associated with failover and availability planning is how much performance loss end users are willing to withstand, and for what length of time.

Step 5: Document Appropriate Actions

The total risk involved for each threat and component — probability, extent, etc. — ultimately determines the appropriate action. Determining the number of backups you need to overcome the odds of a failure occurring hinges on the odds of failure for the components, the likelihood of destruction during a recovery, the cost of the component, the cost of downtime, and many of the other factors discussed so far.

For example, items such as PCs are very inexpensive when compared to the losses that can occur and the number of threats these components face. For these reasons, you can easily justify keeping plenty of extra PCs available for immediate use. Items such as application servers are much more expensive, but are still exposed to considerable risk, and their loss can easily cost a lot of money. For items of this nature, you should always consider having as many backups available as possible. To reduce costs you can formulate plans to "borrow" servers from training or other environments to temporarily replace production items until the initial disaster has been averted.

The good news in the R/3 environment is that you can combine your backup procedures with performance and tuning and have a win-win situation. For messaging and application servers, you can usually devise a plan to allow switchovers from one application server to another. Sharing resources is a natural part of any R/3 setup, thus providing added performance as well as built-in backup and recovery.

Item 5: Size the R/3 Recovery System Properly

What size and configuration do you require for your backup environment?

Technically, running all modules and users on a single application server is possible. However, we all know that the ability to run a production environment without serious performance bottlenecks requires taking many factors into account.

> ✓ ***Tip***
>
> *A common choice when evaluating application servers is whether to have one very large server or several small servers. Argued strictly in terms of performance and maintenance, it's possible to go back and forth on this question for days, with no clear-cut winner. However, I have found that once I frame the question in terms of recovery and availability options, this decision has always been easy for me.*
>
> *I rarely choose one large server because several smaller servers afford a variety of recovery and failover options. When recovery/failover time arrives, I don't have to listen to a vendor explain why they can't send me one of their large servers. And if problems occur on a day-to-day basis, I can rob parts from one server and move them to another until the vendor finds time to complete their warranty support. Therefore, I prefer several small servers over one very large one.*

For addressing performance and capacity planning issues related to your backup environment, I strongly suggest visiting SAPNet. Within the Performance and Benchmark section, you will gain access to SAP's Quick Sizer tool. This easy-to-use, online tool can be of great assistance to you when planning the capacities needed for a backup or remote environment. You just enter a few transaction volume figures and active user counts to get a list of server and disk options capable of supporting your estimated workloads. A quick call to your end users for some current production figures will provide you all of the information you need to complete a variety of sizing exercises. The Quick Sizer tool will let you develop ballpark system requirements for as many situations as you care to ponder. And hardware vendors can use the Quick Sizer output as the basis for pricing equipment per your specifications.

Combine the results of your Quick Sizer computations with the basic setup and tuning guidelines in **Figure 1**. This should give you a great start on determining the configuration you will require for your backup environment.

Item 6: Perform Regular and Complete Testing of Backups

It seems obvious to say that you need to test how your systems respond to various types of disasters and how well your backup plans work, but it's surprising how often critical tests are simply overlooked. The following true story provides an illustration of this point.

One of the largest corporations in the US incurred considerable losses and weeks of expensive downtime when a natural disaster destroyed their corporate headquarters and information center. To avoid any future loss of the data center, and to keep downtime under 48 hours, they established a remote data center, some 500 miles from the primary site, that incorporated duplicate components of nearly every piece of hardware in the environment.

I was asked to review the entire process and make any suggestions I felt appropriate. After a tour of this most-impressive facility, I was presented with test results showing that 36-40 hours was the average startup and recovery time required for the facility to be 100% ready for full-production processing, leaving (they claimed) more than 8 hours for unexpected problems or delays.

Figure 1 Conventional Guidelines for Baseline Sizing of R/3 Recovery Systems

✓ **Total Work Processes**	Most application servers run best with fewer than 25-30 work processes. However, if the average transaction is a simple update of one or two records, such as you have with Financial (FI) or Payroll Accounting (PA) updates, you may be able to use up to 40-50 work processes on larger, faster processors.
✓ **Update Work Processes**	For transactions with average or light updates to records, such as FI and PA, you will need 1 update work process for each 6-8 dialog work processes. For heavy updating, or to ensure better performance, plan on 1 update process for every 4 dialog work processes. Batch work processes that support heavy updating, such as Point of Sale (POS) or Application Link Enabling (ALE) transactions, may need even more update work processes. Batch processes that support reporting and light update transactions may need fewer.
✓ **Batch Work Processes**	If you have absolutely no production statistics to base estimates on, consider 1 batch process for every 6-8 dialog work processes.
✓ **Dialog Work Processes**	A dialog work process can normally support approximately 10 Medium Active Users (those who execute 10-15 transactions per hour) in FI, PA, and Personnel Development (PD) modules. For all other core modules, expect a dialog work process to support 6-8 Medium Active Users.
✓ **CPU and Memory**	Generally, when purchasing Central Processing Units (CPUs) and memory, the more the merrier. However, at some point you hit a saturation point where you are simply spending money for no reason. I suggest an absolute minimum of two CPUs for all application servers and four CPUs for database servers. Use the Quick Sizer tool to get some good numbers on CPU and memory requirements.

I then inquired, "I understand your hardware will be up and ready in less than 36 hours, but how long does it take to reload and test your applications software so that end users can be pounding keys at full speed?" Complete silence gripped the room as the CFO realized they had the equivalent of a fully functioning airport with no planes or passengers!

This company learned the importance of *complete* testing of backup plans and procedures. If you want to go all out with your testing procedures, you can simulate a complete disaster and recovery process. Time each and every event, especially situations presenting problems you had not previously considered. You may not have the time and resources to run and crash the entire system at one time, but simulating loss and recovery of bits and pieces of your systems in several separate, controlled scenarios may get the results you need.

Here's one method for simulating a random loss of your hardware and software components:

1. Before the day of the crash, decide the percentage of employees you think would be affected by the disaster you intend to emulate. Include employees who will be unaffected, some who will be partially affected, and some who will be completely lost for one reason or another.

2. Assign a color code to each group of employees — green for unaffected, yellow for partially affected, and red for completely lost. Make up the appropriate number of red, yellow, and green cards to pass out at the beginning of your disaster simulation.

3. Distribute the cards randomly to both technical and non-technical personnel (don't forget to

include management personnel so that you can see how hard it may become to get approvals and decisions on some seemingly minor items):

- Anybody receiving a green card can be considered unaffected by the disaster and proceed to their workstation for a full day of trials and tribulations.
- Those with red cards represent people who have been completely disabled by the disaster (including employees who were killed or injured, or those who are completely unavailable as they tend to personal matters). If you prefer, you can employ them as observers or recorders during your simulation, but they cannot participate or offer any advice on business matters.
- People with yellow cards represent employees who are still available to report to work, but only on a part-time basis due to personal matters that will keep them from 100% dedication to their jobs. On each of the yellow cards, include a number that indicates the percentage of time they will be unavailable for work, or the exact time periods when they must leave the office/test simulation. When they are unavailable, they cannot participate in any way, just like the red-carded employees.

You can spend as little or as much time as you wish coming up with additions to this disaster and availability simulation. The more expensive your downtime, the more sophisticated your disaster recovery re-creations and testing should be. If a full-blown test is too expensive or cumbersome, put together a team of experts and lock them in a conference room where they can play out the disaster on paper and discuss how various scenarios would affect the firm. Even a simple, paper-based simulation of disaster can produce some surprising challenges and solutions that are worth loads of time and money saved in the event your test scenarios come to pass.

Seven Key Testing Considerations and Recommendations

The seven testing considerations and recommendations that I believe are essential to any recovery and availability testing plans are as follows:

1. **Document all tests, and their results, in detail.** You will learn something from every test you perform, and the results of tests in one area will usually help you design appropriate tests for other areas.

   Should you test absolutely everything, or can you test one piece and apply the results to others? Some component failures can be simulated with a single test. For instance, in shops with 40 or more application servers, you really don't want to simulate a loss of each and every server. If the hardware and software is identical, and the recovery process is the same, you can simulate failure for one or two servers and apply your knowledge and recovery techniques to the others.

   However, don't assume that testing the recovery of a single server will simulate a situation in which there is the simultaneous loss of several servers or the controlling network. Also, for critical business functions, such as Sales and Distribution (SD), you may need to conduct more detailed testing than you might for less time-critical support systems, such as FI. Your recovery systems for SD might include use of PCs or some other temporary order-taking process, and therefore might require separate testing procedures.

   Consider concentrating your efforts on the often-overlooked technical items covered in Items 3, 4, 8, and 10 of this article.

   No matter what you decide, you will need the results of previous testing, so document everything you

The sidebar below outlines the seven key testing considerations and recommendations that I believe are essential to any recovery and availability testing plans.

Item 7: Ensure Availability of Phone Lines

Far and away, the most overlooked item in any recovery situation is the phone. Stop and think about how many times you pick up the phone and call someone for assistance, even if it's just to have a pizza delivered after spending 12 hours locked up in the conference room.

Even if the majority of your key support personnel have cell phones, you can't assume that cell phones will work when other lines are down. In a disaster, cell phones can be the first to go, jammed from overuse or loss of towers and switching stations. Most cell phones eventually share the same lines as hardwired phones, so be sure to consider what you will do when every phone around you is worthless. How many of your system components are absolutely worthless without those very same phone lines? A majority of today's systems are so network-dependent that a simple phone outage could completely disable a multi-million dollar environment.

In disaster recovery situations where you do have working phones (cell or hardwired), ensure that your recovery supplies include a list of names and phone numbers. I've been stopped dead in my tracks many nights when the phones were working fine, but we didn't have a simple list of phone numbers for contacting the support personnel who were needed to rebuild the system, or the end users needed to verify it once it was rebuilt.

The most overlooked phone numbers were the ones for end users. There is no reason to drag your

do — both before and after the actual test. For R/3, your test documentation should include hardware models and settings, operating system versions and settings, R/3 releases, and all configuration settings (application level and technical/basis).

For recovery, and especially availability planning, time each event and then compare results with the business system requirements. When timing events, time the entire process and each step involved. I suggest using a simple wristwatch and timing results as if you were an end user. Don't rely on technical times for transaction execution, database access, or network processing. Users care only about the total time they must wait.

2. **Use test procedures to uncover failures.** Don't fall into the all-too-common trap of merely proving your system can generate good results under ideal conditions. Prove that your system can stand up to the rigors of day-to-day life. When you design and write application code, you should design your system to address the possibility that the database might fail. Incorporate standard checkpoints and "safe zones" so you can trace complex transactions that did not complete their entire function before the process was interrupted.

   The same is true for designing how each component in the environment interacts with other components. To simulate failure, you usually need to simulate multiple failures. Suppose you lose a network server, and when you recover with a new server you are still having problems. In the real world, the backup server could also be defective, or you could have other problems that initially appear the same. Proper testing will help you devise appropriate recovery strategies for the hardware, as well as give you a good indication of how many levels of backups you will need for data files.

(continued on next page)

entire technical support staff out of bed if you can locate the right end users to verify the results.

> ✓ ***Tip***
>
> *You might be able to avoid calling personnel associated with a particular system by building test files and verification procedures that will allow the support staff to run some simple quality assurance checks. For example, with payroll systems I always recommend a special test file of mock employees covering the most common sets of requirements for calculations and allocations to summary accounts. Once the systems are restored, you simply run the programs using the test files and compare the results to the known answers that you keep locked up in your recovery file cabinet. The more complex the system, the more involved these files and comparison reports will need to be.*

Item 8: Plan for Rebuilding Network Resources

Once again, think about those phone lines. While rebuilding phone lines will help you in your recovery process, it will not complete your networks. Many Local Area Networks (LANs) are hardwired with internal equipment. This is both good and bad. It is certainly a great way to eliminate your biggest risk item — the phone companies. But the more in-house wiring you have, the more you must prepare for loss of equipment. Therefore, I recommend including the following items in your planning:

☑ **Have lots of extra parts in very convenient places.** Base your need for inventory on items that are the most common, the most critical, or the hardest to find.

You may want to stock your supply cabinet with memory boards, modems, software (e.g., backups,

(continued from previous page)

3. **Test and retest vital components.** The more important an item is, the more it should be tested. Let's briefly consider your database server. Because so many other components rely on the success of the database server and the many functions it performs, and because the entire system grinds to a halt without it, you must be assured that your recovery techniques can guarantee the database server will always be available.

   For a vital component like this, you should conduct tests to determine recovery strategies for partial failures, complete failures, and performance slowdowns. Simulating a complete failure is the easiest. While the system is up and running, simply turn off the server and watch what happens. This technique can be applied to just about any hardware item.

   The other easy hardware test is simulation of intermittent power losses and "glitches" within a particular piece of hardware. For these, you turn off the power switch, wait a second or two, and turn it back on. Some hardware components may pick up right where they left off, while others will either be "lost" or cause components connected to them to become completely confused. A few components can easily continue to process as if nothing happened. Some, however, will produce the most deadly result possible — they will continue to process as if nothing happened when things are out of synch. These are the items that require you to have routines in place to trap the error conditions and take control of the system before more harm is done.

   After a few of these tests, technicians will quickly learn how to read error messages and plan their strategies for future failures.

diagnostic sniffers, etc.), wire, connectors, disk drives, keyboards, etc. When you stock items like modems, work/buy at the lowest common denominator. That is, plan your recovery system to work with the simplest and most common items available, even if that compromises performance in some areas. Keep everything simple and routine.

If you have unique components in your platform, you must be able to provide your own equipment and support techniques during recovery.

☑ **Be prepared to improvise.** In a true disaster, there will be limited supplies of everything, and there is no telling where you may need to go to borrow an item or from whom you may need to borrow it. For example, you may prefer to print certain items on bonded paper, but if your supply room has been flooded by a broken water pipe, you can't simply run to the office supply store and pick up your custom paper. However, they will have lots of plain white paper on which you can stamp or print some unique graphic images. That will get you by until the real thing can be replaced.

☑ **Don't forget the tools that go with your equipment!** You don't have time to search the entire floor to find your wire cutters or Phillips-head screwdriver. Designate a secure, safe, and accessible file cabinet or two as your recovery storage area and store an extra set of tools, supplies, and diagrams there. Your emergency kit might also include nuts, bolts, and screws.

☑ **Make sure you have wiring schematics and diagrams.** You will need accurate and up-to-date schematics for every single wire and connection in the building and your entire system environment. And they must be in a format that is easy to read and understand. Most wiring diagrams are printed so small they are worthless. And a few are likely stored in a computer where you can't print them out — certainly not if that computer is in a state of disrepair.

Keep everything clean and simple. All instructions

4. **Analyze effects on dependent systems.** This step is critical for testing interfaces, networks, or servers that support multiple users or functions. Any interface between multiple components must be tested from every angle to ensure recovery can be completed no matter which piece fails. If information travels from point A to point B, then simulate a crash of: point A only; point B only; the path between point A and point B; point A and point B together; point A and the path; point B and the path; and finally, point A, point B, and the path together. If the information travels in both directions, then you must duplicate the tests while information is flowing in both directions.

   Let's consider a Sales and Distribution (SD) system that polls retail outlets for nightly orders and updates, then generates goods issue statements and related billing statements used to load trucks and ship products each night. Your system must be smart enough to recognize any interruptions and recoveries to the processing of orders so that inventory items are not duplicated or lost. So how will your particular system react to such an interruption of processing? For most customers, the answer is "it depends." There are usually a multitude of checkpoints and configuration settings that determine the appropriate action at several key steps of the process. Many are built into R/3 and occur automatically, and many you have full control over.

   If a problem occurs in the middle of transmission, most processes have several options that control whether you will accept the orders you have received and try to restart at the point of interruption, or whether you will throw out the entire batch and start over when you are assured the entire system has been restored. In most cases, you will have options at the hardware level, operating system level, and R/3 level.

(continued on next page)

should start with the most basic of operations and proceed through detailed steps as if the reader were completely ignorant. Remember, the person who normally installs and tests hardware may be unavailable.

Item 9: Ensure the Disk Space Needed to Recover Is Available

Somehow disk space always gets used for development, testing, or "temporary" production work and is never quite ready for immediate use. You need to ensure the rules on disk space allocation are very clear. Block access via security and passwords and/or by just not letting the general public know about available disk space.

Since the cost of disk space is relatively low now, acquiring it is easy compared to the old days. However, even in recent times, there have been countless occasions where we have attempted to recover files to backup disks filled with so-called "temporary" files. Of course, we did eventually get access, but the time lost trying to find out who, or what, was residing in our space was quite costly. And on those occasions when we simply cleaned the disk and took control, we ended up losing a lot of time arguing over whose actions were right and whose were wrong.

Item 10: Document How to Deal with Corrupt Backup Files

What will you do during a recovery if your backup files are corrupt? This is a real, everyday problem that you must be ready to deal with.

Analysis of this subject will lead you to the conclusion that backups must be, but aren't always, trustworthy. The only way to gain faith in your backup systems is to test them on a regular basis. Once you have solved the problem of making sure that you have three, four, or whatever number of backups you need, decide where you must store them to provide some offsite protection. And then, for disaster planning purposes, consider how to deal with

(continued from previous page)

Without a complete review of your particular environment, I can't tell you how to set up your system, but I can tell you that your review should conclude with a series of tests to simulate a variety of failures and recoveries. Simulating failure of the various components will indicate whether you need backup data files stored within each subsystem and, if so, how many backup files you will need.

5. **Consider database effects whenever hardware is replaced.** You need detailed before-and-after snapshots of data and related processes when analyzing tests that involve the processing of data or transactions. Simple dumps of tables and files or routines to compare raw data are often enough to determine whether data files have been affected. For some tests you can simply compare the size and location of data tables to validate test results. In other cases you will need to review database statistics to decide if data tables have been modified. For complex transactions, or when testing critical elements, you may need to print portions of files, trace transactions, or write special programs to verify table contents.

6. **Prove processing capability in addition to availability.** Determining whether a network is available for use is easy: crash it, rebuild it, and send some transactions. To determine if recovery of a particular network path via rerouting of traffic is a viable recovery technique, you may need to load the system with a particular volume and type of transaction. The way in which several small transactions affect the component being tested may be completely different from the way a few very large transactions affect it.

losing your remote site. Maybe you need several remote sites.

But isn't that expensive? Yes, it is — if you need an entire building to store your data. But most people do not need an entire building. If your database is large enough to demand a full-blown remote site, your firm is probably big enough to afford the entire environment.

It's important to remember at this point that the whole idea of backup and recovery is based on risk management and relative cost/savings. For smaller organizations, a normally safe, but overlooked, alternative is a security deposit box or two at the local bank. It takes little effort to run a few tapes or hard-drive backups over to the bank on a regular basis. And the rates are usually extremely cheap in relation to the risks involved.

Many companies overbuild their computer rooms and storage facilities and sign agreements with other companies to provide each other a safe, secure, remote site. Some go as far as having time-share agreements for disk drives, servers, etc. in times of disaster. If your situation demands such detail, remember to test these plans in full before relying on them. Retest them after every update to your system or to the backup site's system. You will be surprised how many quirks in someone else's operating system or network procedures will stop your processes dead, even when you think you are both running the exact same version of R/3 and supporting functions.

Item 11: Ensure Temporary Support Resources Are Available

The items in this section, while not as technical as the resources we usually think about in a disaster recovery/high availability situation, are all things that were real show-stoppers for the recovery efforts I have been involved in over the years. Don't forget, most recovery efforts take place at night and in very isolated conditions. Whether we are talking mishap or disaster recovery, the room always seems to be deserted when you are looking for good help. So,

For assistance in this area, review SAP and third-party vendor software products for stress/volume testing and implementation tools such as CATT (Computer Aided Test Tool). Remember, recovery is not complete until the users say it is. And I promise, they won't be satisfied unless performance is up to previous standards.

7. **Complete testing before implementation!** Before beginning your development project, and before you let any users on the system, allow your Basis personnel to test their basic recovery techniques on the system before any configurations are made or data is input. You certainly can't expect your support team to guarantee results on a system they haven't had time to simulate recoveries on. Sounds obvious, but a lot of companies skip this step. No matter how pressed for time your project may be, you don't have time to skip testing of recovery processes before putting new items into production status. Remember the old saying, "If you can't find time to do the job right the first time, how are you going to find time to do it over?"

   Because hardware prices are so low these days (in relative terms), most customers with complex systems will install a sandbox system that is used strictly by technical support staff for testing recovery techniques and new release installations. If you don't provide separate equipment for testing, the support team's only option is to commandeer development, test, or production systems where the cost of even a minor miscue is easily more than that of duplicate equipment.

don't forget to ask yourself these questions when you are drawing up your plans:

☑ **Where will you work if the office is demolished?** You can go to the depths of emergency rental agreements for remote office complexes or stick to simple handshake agreements with a few of your neighbors in town. If you end up working in remote or unfamiliar offices, you may not have access to the company cafeteria and you will probably be working long hours without time to go to your favorite deli. Something as simple as a list of favorite sandwiches or specific diet requirements in the admin's desk can prove very handy during tough times.

☑ **Where will you get money to buy necessary items, meet payroll and expenses, acquire food, pick up or deliver mail, etc.?** You should always have agreements with the bank to acquire emergency cash, but also consider the use of petty cash for a small stockade of food and money that can provide the bare essentials for a couple of days. I won't go into the details of where I worked when our computer operator produced a six-pack he had stored in the raised floor of the much-colder-than-normal computer room. Yes, this was back in the good old mainframe days, against company policy, and not something I suggest you follow. But it provided enough relief one night following a 36-hour shift that the vice president gladly "threw one back" with us. He never told a soul and was more than understanding when we presented our next budget, including a small break room for technical support staff.

☑ **Where and how will you get office supplies?** In the file cabinet you designate as your emergency storage area, pop in a laptop computer, printer, paper, staples, pencils, coffee, first-aid kits, etc.

Item 12: Plan for Loss of Personnel

When you design your recovery plans, take into account the potential loss of any or all personnel, as well as any or all system components. If you are one of those companies that claim people are your most valuable asset, prove it by listing them first in your system recovery plans.

If you haven't already posted official on-call listings with at least primary, secondary, and tertiary support personnel and associated protocols, I suggest you do so now. List step-by-step instructions for personnel to follow for any system issues, starting with simple help-desk-related problems, working all the way through true disaster recovery. Most shops already have detailed support processes in place, but the thing often left out is the on-call list for end users. Just because the report printed successfully doesn't mean the numbers are correct. Who will verify the business end of your recovery efforts? After all, I did say recovery isn't complete until the users say it is.

What if support staff is unavailable during the recovery process? Your normal recovery plan, as well as your disaster recovery plan, needs to address this question. At any given time, you may have one or more employees who are leaving for new career challenges or other reasons. When a community-wide disaster occurs, you may be faced with the reality that support personnel will need to take care of some personal business before returning to work.

Item 13: Consider the Loss of Business Associates' Systems

Don't forget to consider what you will do when your best customer or biggest supplier's system bites the dust. How will you get purchase orders to, and confirmations from, your vendors? Your own ability to estimate future inventory and product shipments might have significant ties, logical or physical, to systems outside of your control.

Suppose you sell orange juice as a major source of revenue. If a hurricane rolls through Florida and wipes out your vendor's orange farm, how will you

process and sell your product? Or, what if the hurricane leaves the farm untouched but renders their systems useless for a month, killing your fancy new Business-to-Business (B2B) ordering network? How will you confirm your vendor's raw material shipments in order to guarantee customers you will be processing their orders on time?

Additionally, if your best customer loses a system, you certainly don't want to lose their business because they found it easier to purchase Brand X over the phone after their PC, or a few phone lines, went on the fritz and kept them from accessing your exquisite Internet site. So, as you evaluate your own system, follow the trail of all interfaces and consider their effect on your ability to conduct normal business.

Item 14: Assign Responsibility for Verifying Results

Responsibility for verification of reports and restoration results should belong to the end users to avoid any risk of restoring and using a less-than-perfect database and related environment. The end users are the only people with the knowledge to review and approve complex calculations and audit trails, especially if you are forced to piece a system back together after a serious loss of data.

One sure way to turn a seemingly normal recovery into a disaster is to implement a backup procedure that contains a few accounting items that don't balance properly throughout the entire environment. So be sure to assign responsibility for verifying restored databases to the vice presidents and department directors on the user side of the house.

Item 15: Review Your Recovery Plan Regularly

When evaluating system components and their vulnerabilities, consider the availability of replacements. If you are using hardware items that are no longer available from the vendors, you are taking a considerable risk — one that can be easily eliminated by replacing those items now. Of course, it will be tougher to replace some items than others, since they may be tied to software being used, require upgrades to operating systems, etc.

These are areas where you must be vigilant in maintaining your system and performance of upgrades. If you do not stay relatively current on the release versions of hardware and software you employ, a seemingly small crash can snowball into a massive upgrade with days of expensive downtime.

I recommend regular review of recovery plans, change management procedures, maintenance schedules, and testing of all resources. Some of the most impressive firms I have dealt with conduct regular reviews of all hardware and software components, in addition to all manuals associated with recovery and maintenance of their systems.

Conclusion

In recovering from a disaster, you are rebuilding more than a system: you are rebuilding your business and everything it stands for. Consider all relevant details, large and small, when developing your organization's recovery and availability strategies.

In its essence, recovery and availability planning is all about risk management — you identify relevant threats, assess your system's vulnerability, calculate the probability that something will be lost, determine the effects of that loss, and document the most appropriate action to take. That's why one of the most important planning tasks is knowing the cost of your downtime. If you overlook it, as so many do, your planning will not be based on a solid foundation and it will be severely compromised.

When formalizing your system support documents, be sure you don't overlook the planning items most often ignored by others: security, loss of

business associates' systems, self-inflicted wounds, access to personnel, and regular and complete testing of backup plans and procedures. Testing of your plans and key system components will be the most important, as they will provide invaluable feedback on your entire process. Ensure your test procedures and documentation of actual test results are as comprehensive as possible.

Key technical items to bear in mind are your networks, and especially the phone lines that support them, disk space, and any corrupt backup items that may exist within them. Perhaps the most often overlooked items for disaster recovery are the temporary resources and work environments you will need during the actual recovery process. Ensure your recovery staff will have a place to work and the essential resources they need to begin rebuilding your business. Rehearse assignments, responsibilities, and required tasks, and make sure you have alternate plans to reassign tasks and responsibilities for employees unavailable during your recovery efforts.

Disaster recovery planning can take as little or as much time as you choose and is one area where brainstorming is vital to success, because no detail is too small. The only wrong way to do disaster recovery planning is to do nothing. Every effort you make will eventually prove worthy of the time spent.

My list of items to review may or may not be complete. However, I believe if you follow this list, any specific items important to your firm that I may have overlooked will surface during these discussions.

If you are having trouble getting management to understand the importance of recovery planning or they are unwilling to fund the project, complete Item 1 and the rest will fall into place. Once management truly understands the cost of downtime and poor performance, they won't be as resistant to the notion of comprehensive disaster recovery and high availability planning and all that it entails.

Kurt Bishop is a retired member of SAP America's Technical Consulting Team. He has experience in both application and technical support of the R/3 system, where he provided a variety of consulting and support services for customers and colleagues. Kurt started his seven-year career with SAP in 1994 providing project management consultation for customers with a specialization in technical project management. After two years in this position, he began concentrating on performance and tuning. By popular demand from customers and SAP employees throughout North America, he then concentrated his skills on capacity planning services, where he worked to standardize and improve the process for both customers and vendors.

Prior to joining SAP, Kurt spent seven years as a management consultant for one of the "Big Six" consulting firms. In this capacity, he was involved in both management and information systems consulting for a broad range of firms and industries. He spent considerable time developing strategic systems plans for some of the nation's most prestigious firms, as well as assisting many clients in reengineering their business practices and related support systems. He also has eight years of experience "in the trenches" of information systems for the oil and gas industry, including positions from programmer analyst to project manager, where he developed the skills that prepared him for consulting. His skills are also augmented with four years in manufacturing and materials management, where he served as production manager. Kurt can be reached at kurtbishop@aol.com.

