Outsourcing

Although outsourcing is always on my mind as I write, we haven't written about outsourcing in detail in almost twenty years. Because our readers include both outsourcing companies and outsourced businesses, we receive questions from both sides of the situation. The latest technology changes provided by z/OS and System z have increased the complexity of outsourcing agreements. If you are in an outsourcing contract or considering outsourcing, this should be of value to you.

Introduction

Outsourcing - letting someone else handle your data processing - is often compared to a marriage (or at minimum, a partnership). If marriage is on your mind and if you want it to be a happy one, then you'll want to put in place certain rules of disclosure and fair play to avoid unpleasant surprises and upsets later on. This article discusses many considerations, precautions, pitfalls and procedures that you need to be aware of in order to construct a good contract and on-going relationship. Service level agreements, baseline measurement, benchmarks, and reporting techniques are a few of the topics covered. This article should be of interest to auditors as well as managers either considering or already involved in such an endeavor.

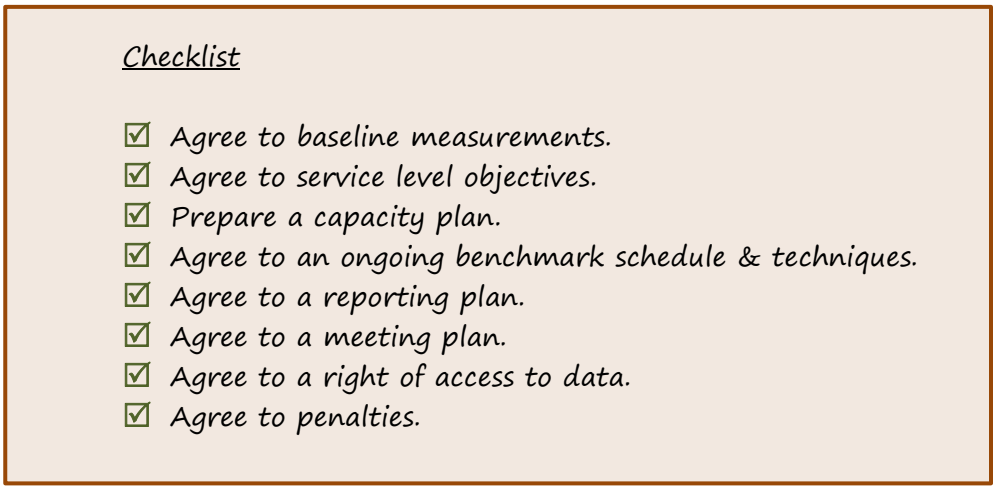
These tips could save you hundreds of thousands of dollars!

A good outsourcing business relationship can resemble a good marriage. But like a good marriage (or any contract), it works best when things are spelled out in advance, assumptions are minimized, and on-going disclosure and communications are open and continuous.

For twenty-five years I was involved in both the customer and vendor sides of these contracts. In one case, I was able to save a customer over $200,000 a month in charges over (expected) baseline by pointing out errors made during the original measurement; by proving that CPU increases and, therefore, CPU charges, were due to the outsourcer's changes; and by resolving a conflict about service levels. All of the problems could have been avoided with a better outsourcing contract.

Figure 9 shows some requirements for any outsourcing contract that will help you build a successful, and non-antagonizing, relationship.

Figure 9 – Agreement Checklist



Ensuring that these items are incorporated into any agreement prior to conversion to a vendor-run installation can save you hundreds of thousands of dollars by avoiding misunderstandings later in the contract. Let's look at each of these in more detail.

Baseline Measurements

Start of the Contract

Almost all contracts begin with the vendor running "baseline" measurements of the existing system. After all, they need to understand the workload that exists today in order to determine the cost of running the contract. Too often, this baseline measurement is accepted without question by the installation. The problem occurs much later when costs are higher than expected, primarily because the baseline was incorrect. Without exception, the customer should run their own baseline measurements and compare them to the vendor's. Typically, they will both use the same SMF data, so their results should be similar.

Unfortunately, the fact of life is that SMF data is subject to interpretation and each analysis will produce slightly different results. Here are just a few examples:

- The date and timestamps might be slightly different on two machines, thus causing negative run times or response times. Some programs or products might set these times to zero, while others will drop the records as invalid. Hence a different job count. Multiple job termination records are written for re-started jobs. Similarly, some programs will drop them while others count them.

- Much of the data center work is run as started tasks that run for several hours or days. If interval recording is not turned on and the system crashes, all SMF data for the started tasks is lost, thus underestimating the amount of work on the system. The same applies to TSO users who are logged on for several hours.

- The SMF period that was analyzed might have several IPLs included in the data, and any unscheduled IPL (sometimes even scheduled IPLs if "halt EOD" isn't entered) will result in lost SMF data. If the SMF and RMF data (most of RMF data is not lost due to an IPL) are not then compared and reconciled, the baseline measurements might be as far as 20% off.

- CPU measurements vary for several reasons, such as running under PR/SM, running on a uni-processor or multi-processor (amounts vary due to the number of logical CPs), running on a lightly-loaded or heavily-loaded machine, the speed of the processors, whether a coupling facility is installed, what type of workload is running, and many other reasons.

- The SMF period chosen might not be representative of the actual workload in the installation. Analyzing a department store or credit card company volume on other than the November/December time frame will give extremely underestimated values for resource consumption since much of their workload occurs during those two months.

This has been only a short list of why SMF measurements are subject to interpretation and should be analyzed from a knowledgeable point of view.

As noted above, the time frame for the baseline is perhaps one of the most critical decisions to be made. Which months (weeks are actually better) are most representative of your workload? To pick your baseline, a two-year profile is almost a necessity. This profile should contain as a minimum: CPU usage (actual hours, not busy percent), CPU usage during peak interval (20 minutes or less), CPU usage by application (even if it's only broken out by TSO, test batch, production batch, online and MVS/utilities), number of jobs, number of steps, number of online transactions, number of online users, number of TSO users, number of I/Os, XCF traffic, network traffic, online response times, and batch turn-around times. Any trends in this history should be reflected somehow on the baseline.

This breakout of the baseline measurements is also a critical step. If you simply add up the amount of CPU time without determining the users of that CPU time, you'll never be able to understand differences that might occur later in the measurements. For example, if the TSO usage amounts to 4 hours of CPU time per month during the baseline period, then suddenly jumps to 8 hours of CPU time per month after the baseline, you'll know where the problem lies. If you simply see an increase in CPU usage after the baseline, you won't know whether the increase is from batch, TSO, your online system, or external changes such as running under PR/SM or a new level of the operating system.

When dealing with CPU times, you need both CPU times and CPU service units. See our *Tuning Letter 2004 No 3* (pages 20-29) to help you understand the difference between the two measurements. You'll need CPU times from SMF type 30 records, as well as CPU times from RMF type 72 service class records. You'll need zIIP and zAAP measurements as well.

Service Level Objectives

Without a doubt, the most important step prior to outsourcing is the determination of current service level objectives (SLOs) and service level agreements (SLAs). Any outsourcing contract must define the service to be provided in terms of SLAs. These are normally specified by application or group of users, but might be specified for the installation as a whole. I've seen too many contracts that somehow bypass the issue with phrasing such as: "service levels will be provided at the same level as at the start of the contract." If you haven't agreed on what the service is at the start of the contract, you'll be battling and feuding for the life of the contract. You can depend on many sleepless nights!

Beware of daily averages – use only peak intervals

For much more detail on SLOs than presented in this article, please see our five part series on Service Levels in previous *Tuning Letters*. (They're on the annual DVD we send to all subscribers which contains all past issues.)

Service Levels - Part 1 - z/OS 101 (SLOs versus SLAs), Tuning Letter 2011 No. 6
Service Levels - Part 2 - TSO, Tuning Letter 2012 No. 1
Service Levels - Part 3 - Batch, Tuning Letter 2012 No. 2
Service Levels - Part 4 - CICS and IMS, Tuning Letter 2012 No. 3
Service Levels - Part 5 - Other Service Levels, Tuning Letter 2012 No. 4

When response times are measured is one of the most important decisions to agree on. I saw one site that had agreed to the outsourcer's statement that average TSO response time was 0.4 seconds, as averaged over a 24-hour period. In reality, that ***was*** the average. A year into the contract, TSO users were complaining of horrible response times between 10 am. and 11 am (and also 2-3 pm). When measured, the peak period response times were over 6 seconds, but the daily average was still 0.4. Looking at historical data, the user found that their peak interval response times never exceeded 1.0 seconds before outsourcing. Unfortunately, the customer lost the battle because the vendor was meeting the contract commitments. The lesson here? Agree to commitments only when they are based on peak interval measurements.

A contract can specify the actual service levels or can define a step whereby both parties agree to a procedure for setting service levels. As an example of the differences in these two approaches, let's talk about TSO response time. Assume that prior to the contract, first period TSO response time was .3 seconds during the peak interval and the users were currently satisfied with the service. One contract might state that first period TSO response time will not exceed .3 seconds for the life of the contract. Another contract might specify that TSO response will not exceed .3 seconds for the first year and a joint review of response time will be held at the end of the first year to negotiate for the next year. In both examples, of course, the contract would need to specify that 99% (or so) of all transactions should complete in period 1.

Why would SLAs need to be changed? Using our TSO example, we've seen that as the industry has changed, so has our perception of "acceptable" response time. I remember when 15 second CICS response time was acceptable (back in the old days), and then 1 to 3 seconds was considered minimal, but now anything over .2 seconds might be unacceptable. If the customer wants better response time, there should be a forum for obtaining that service, even though they may have to pay for it.

Additionally, the type of work changes. More and more applications are using Java, such as WebSphere. Java, while easier to find programmers for, takes a great deal more resources than COBOL. WebSphere response times might be longer than traditional CICS applications.

So what are the SLAs that need to be defined? There are normally four separate areas that are defined: online response time, batch turn-around time, availability, and volumes. Let's look at each one.

SLAs: Online

Online response time is probably the most important of the SLAs because it has such an immediate impact on the users and their productivity (not to mention their morale!). The easiest, but most misleading, technique is to use "average" response time. Averages can be very misleading, since you can't see how bad some people have it, and a single looping transaction could invalidate the average for an entire day. One CICS application that I reviewed had an "average" response time of 4 seconds. On further inspection, half of the users had .2 seconds and the other half had 7.8 second response time. I'd hate to be in the second group! As you can see, the average of 4 seconds wasn't representative of either group.

A better technique is to report response times by percentages. For example, an SLA might state that 90% of all users will receive 1.0 second response time. The other 10% might be dying, but at least you've addressed the majority of the users. Some installations only care about peak hour response times while others might be concerned with prime time (day shift). You'll probably have different SLAs for each online system and for each user group.

The battle between measurement of internal versus external response time is a continuing saga. The users only care about (and experience) external response times, but the software only measures internal response times. Sometimes the difference between the two can be several seconds. So what'ya gonna do? You can either pay for the hardware to measure external response time or you can find a way to estimate the external response time. Hardware solutions might include installing intelligent PC applications at the terminal to collect and record the actual response times. Estimation techniques vary from use of a network monitor to estimate host and network response times to a stopwatch approach periodically (weekly or monthly stopwatch tests at various terminals). Whichever technique you use, the user should be in agreement with the measurement techniques.

SLAs: Batch

Batch turn-around time for test jobs can be tricky. It's not extremely difficult if you have standard job classes that restrict resource usage (e.g. Class T jobs have no tape mounts and take less than 1 second of CPU time). The turn-around time can be defined from the time of job submission to the job end (not including print time, since as soon as the job ends, most users are able to see the output online). This is an easy measurement to obtain from SMF. Some installations, however, have chosen to use only input queue time for their turn-around objective, stating that the data center isn't responsible for how long the job runs - that's up to the program. In a perfect world, that statement might be reasonable, but it's not. If the data center chooses to open up 50 batch initiators, the input queue time would be zero, but the job turn-around time will be excessive due to very large swap out times or long delays waiting for the dispatcher. In almost all cases, test batch jobs should have different SLAs for each job class based on the time from job submission to job end.

Availability SLAs are critical but very complex to determine

Batch production is another story. In most cases you don't really care how long the job takes, but you do care that it's completed by 6:00am in the morning. This is often called deadline scheduling and can be a little hard to implement. Often, you need a list of specific job names and their required completion times. Too often, the completion time alone is not sufficient. If the reports aren't delivered by 8:00am, the users complain. Report delivery would then have to be tracked using a manual system. While I'm not suggesting this level of tracking, it might be necessary in some installations.

For more information on batch service levels, please see our 19-page article in *Service Levels - Part 3.*

SLAs: Availability

Having the system up and available is even more important than having good response time. But the measurement of availability is a tad difficult. While it's fairly easy to determine if MVS is alive and running and even to determine if CICS (or IMS or DB2 or ...) is up, it's very difficult to determine if the user can get to an application.

All of the pieces have to be working for the user to see results. This includes: MVS, VTAM, the online application such as CICS or TSO, channels, communication controllers, cluster controllers, terminals, applications and data bases. If any of these has a problem, so do some of the users.

How do you report the fact the CICS was available for 80% of the users, but wasn't available to 10% of the users because of a controller failure, and wasn't available to another 10% because a database was still loading? Most availability SLAs ignore the partial downs, but will lose credibility with the users who see the true story.

Some installations choose to solve the problem in the following way: They set an availability guideline for the applications (i.e. CICS is available for 98% of prime shift)

and also set a complaint threshold (i.e. no more than 2 complaints per day for unavailable resources by user). Different availability guidelines might be set for MVS and each online application. These are typically set up for prime shifts only. (There's a major difference between a 2-hour down at 3 am versus one at 11 am!)

Service level availability is discussed in *Service Levels - Part 5* of our multi-part series.

SLAs: Volume of Work

In a non-outsourcing environment, the response time and turn-around time objectives are often based on some volume of work. For example, CICS response times of .3 seconds can be guaranteed only if there are less than 1,000 transactions per second. If the user increases the load, then they can expect to see poorer response time. In an outsourcing environment you might have a different scenario. The outsource vendor might have built into the contract a higher price for higher volumes. If so, there would be no limit on the volume of transactions or jobs and the vendor is committed to provide the agreed upon response time. If not, the vendor might have limits set on the user to restrict the amount of work and to invalidate the response agreements when the volume has been exceeded. Either way, someone will end up caring about the volumes as they relate to response times.

Capacity Plan

Many outsourcing agreements have neglected to take planned growth into account. They "assume" that the workload will remain constant or will slightly increase and they base the charges on the anticipated slight increase. Anything above the "baseline" of this anticipated increase is charged at maximum rates. The vendor does this to protect themselves from large unplanned increases in volumes. Prior to commitment on an outsourcing contract, a capacity plan should be prepared by the installation to show the anticipated increase in workload. This plan would include expected growth of current applications as well as new applications.

Most organizations are hesitant to do this saying that the reason for going to an outside vendor is to eliminate this level of detail and the technical expertise to perform such a function. That's understandable, but the cost of not performing this step can result in several hundreds of thousands of dollars extra during the life of the contract. If the technical expertise is not available, then bringing in an outside consultant to prepare a plan will pay for itself many times in the years to come.

Benchmarks

Benchmarks are a critical part of any outsourcing plan. The simple fact of life is that hardware and software changes result in changes in the CPU seconds consumed. For example, running a workload under a fully loaded multi-LPAR environment can result in CPU increases of from 5 to 30% over running the same job in a stand-alone environment. Since the use of LPARs is for the benefit of the vendor and may result in de-

creased service to the customer, it's unfair for the customer to pay for the increased CPU time. Likewise, the use of certain MVS functions will result in increased CPU times to eliminate I/O and improve response times. If the contract is heavily weighted to charges for CPU time consumed, the customer will be paying extra for the improved response times whether they want it or not.

A good benchmark can identify any changes due to hardware or software changes and can be used to adjust factors for charging. The basics of a "good" benchmark include the following:

1. Jobs that are representative of the actual workload of the installation (e.g. assembler jobs aren't good benchmarks for COBOL workloads since COBOL programs tend to use more storage-to-storage instructions).

2. Jobs that are consistent in any environment (i.e. not dependent on the placement of datasets) - these are generally the CPU-bound jobs.

3. Online transactions that are representative of the major transaction volume of the installation.

4. Actual production jobs that have the same input.

5. Multiple runs of the same jobs (10 runs is normally acceptable to provide statistical averages).

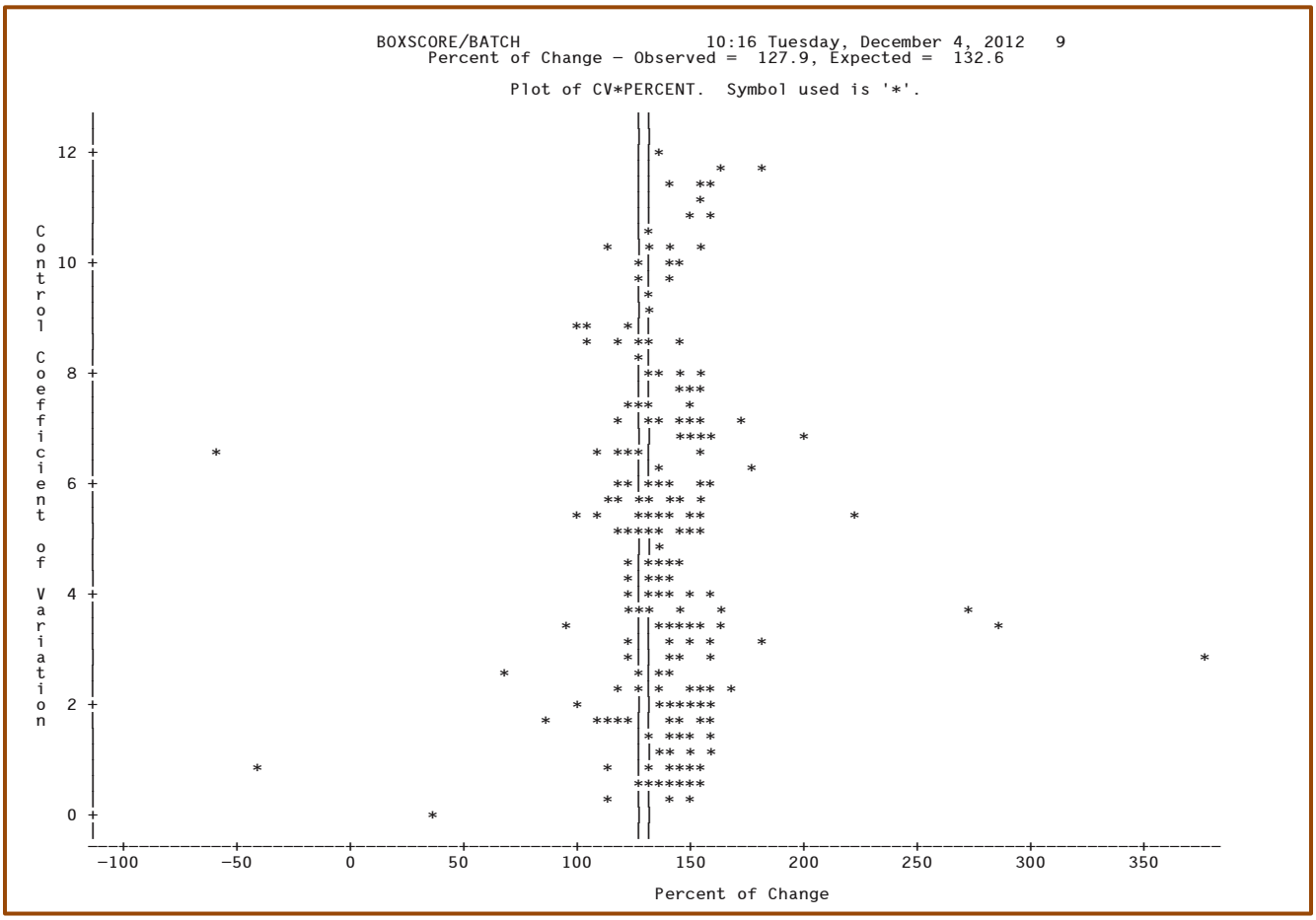
Periodic benchmarks are a must!

Benchmarks should be run prior to the start of the contract, after any major upgrade of hardware or software, and at some pre-determined time period (monthly or quarterly) to determine if other things changed. For example, if you run a benchmark immediately after converting to an LPAR environment, you won't see much of a change. It's only after other workloads (LPARs) start processing that you'll see a CPU increase in your own workload. Periodic benchmarks are a must!

This is the reason we created our software product, BoxScore (www.watsonwalker.com/boxscore.html) - to help people, especially outsourcers and their customers, determine the effect of any change, whether it be hardware or software. BoxScore identifies all of the job steps or transactions that are stable before the change, confirms that they're still stable after the change, then determines the average amount of change in CPU time for each set of jobs or transactions. One of the BoxScore reports is shown in Figure 10. Each point in the graph is one or more stable job (or transaction) sets. The % of change is shown on the X-axis, while the variability (coefficient of variation) is shown on the Y-axis. The two vertical lines show the observed (actual) and expected CPU usage. In this case of a processor change, the installation was expecting a 132.6% improvement, but saw only a 127.9% improvement. But the important part of this chart is the amount of variation in the job steps. Each job step behaves slightly differently when moved from one machine to another. If you're the outsourcing supplier, you'll point to the jobs on the right side of the graph

to show how much CPU time is saved (one step ran 400X faster!). The outsourced users who had jobs at the left side of the graph will be complaining that their jobs took more CPU time than expected and cost more than expected (two steps actually took *more CPU time* on the slower machine). In an outsourced environment, you'll need BoxScore or something like it to keep an eye on changes in the installation.

Figure 10 – BoxScore Chart of CPU Changes



Reporting Plan

Here's the place to save the marriage! It's called "communication," both verbal and written. This follows all of the same guidelines as found in marriage manuals. Communication should be open, honest and continuous.

Daily reporting (weekly at a minimum) is essential. As part of the contractual commitment, the vendor should define the reports that will be provided on a daily, weekly and monthly basis.

Reports should include a minimum of:

- Service level agreement reports, both agreed and actual delivered service.

- Volumes in terms of jobs, steps, transactions, and resources (CPU, I/O and storage).

- Trending reports to show baseline and current, and chargeback amounts (unless determined at month end).

During the contract, derivation for the reports is needed as soon as they're produced. A separate documentation manual describing the reports and how the data is obtained is needed to reduce any misunderstandings. For example, if CPU time is reported, you need to understand whether the CPU time was obtained from SMF data or RMF data, whether all jobs were included or some excluded, whether it was factored for any reason (such as conversion to a different CPU model, adjusted for PR/SM, adjusted due to contract agreement, etc.).

If service levels are reported, you need to understand how they were obtained (what the source is), what time period the responses represent, the application and user group represented, and the definition of how external response time is determined.

Meeting Plan

Meetings are the mechanism to eliminate misunderstandings. The more you talk, the less confusion and irritation you'll have. Here are some recommendations for running effective, productive meetings:

1. Publish an agenda prior to the meeting to define the attendees, topics for discussion, time, location and length of meeting. One person is responsible for distributing the agenda and is contacted by others if they want to add something to the agenda.

2. Hold the meeting using the topics defined in the agenda (there should be no surprises with new topics - if someone wants to introduce a new topic, it can be mentioned and scheduled for the next meeting).

3. Send minutes out to attendees and non-attendees within a day of the meeting including the following:

    - Items resolved
    - New due dates
    - Problems that were defined
    - Responsibilities for follow-up
    - New topics for the next meeting

It really doesn't matter whether the vendor or customer writes the minutes, since rebuttals of the minutes can be documented in the following meeting and noted again in the minutes. The more that's documented, the less confusion and misunderstandings you'll have.

What about "trust"? I continually run into environments where the attitude of the vendor or customer management is: "why do we have to document everything, we should trust one another - after all, this is a partnership (or marriage)". My response to that is very simple: it's not a partnership - it's a business relationship between customer and vendor. Both parties benefit from a smooth-running relationship with good documentation and no assumptions.

Right of Access and Disclosure

Most customers have either an audit department or a small technical staff to monitor the vendor operations. This is normally acknowledged in the contract where the scope and responsibilities of the customer audit function is defined. Part of the scope of this audit function demands access to all datasets and raw data that relate to the customer account. This includes access to raw SMF data (including RMF data), SYS1.PARMLIB members, and source code for programs that produce the status reports described earlier.

Outsourced customers must retain key technical experts to protect their interests

The audit function is crippled without access to these facilities. Additionally, some type of online monitor should be given to the customer to monitor the system. RMF Monitor II or RMF Monitor III provides that access under TSO, but any online monitor will serve the purpose. It's not the responsibility of the audit function to "tune" the system or locate performance problems, but to see if the customer priorities are being implemented (i.e. CICS is a higher priority than TSO). Independent consultants or auditors should be usable by either party at their discretion.

You will almost never hear of a bad outsourcing experience. The reason for that is that the majority of outsourcing vendors include an agreement that ALL of the contacts, agreements, disagreements, missing service levels, cancellation of contracts, etc. must be kept confidential. Penalties are enormous if a customer discloses to the press or any other public forum problems with the contract or the agreement. An outsourcer who intends to meet the service agreements has no need of such a restriction, and the presence of such a restriction will severely limit the options of the customer to obtain satisfaction in the case of poor service. Avoid this type of clause at all costs.

Penalties

Too many contracts forget about penalties. They might document that a certain service level is to be maintained, but omit the penalty for not meeting that service. In such contracts, there is no incentive to meet the service levels.

Penalties, such as reduction in charges, are imperative when talking about missed service levels. These should be documented in the contract along with the service levels. Some means of arbitration should also be defined that allows an outside arbitrator in cases that can't be resolved.

One very poor contract stated that if the customer specified in writing that the contract commitments weren't being met, the outsourcer could correct the problem within seven days with no penalty. It was the customer's responsibility to notice that the contract wasn't met. The result, as you might expect, was that good service was only provided one day a week.

On the other hand, one outsourcer I know of accepts a penalty of the loss of an entire day's revenue for any day with missed service. When I asked the outsourcer how he could afford to do that, he laughed and said "we never miss our service objectives!"

Insist upon strict penalties for missed service that the vendor will work hard to avoid.

A Final Note

Some of the contracts I've seen have all of the above items included in the contract but still cause untold frustration on the part of both companies. The problem lies with the lack of concern about contract compliance. The contract is designed to provide a good working relationship.

If you choose an attitude of "let's not rock the boat" by not insisting on contract compliance, you are essentially giving away the store. Responsible management should insist on both parties being aware of contractual commitments and meeting those commitments.

Obviously, the first step is obtaining a contract that is protective of both parties and protective of what is often a very fragile relationship. The considerations mentioned in this paper should help in that process. ■