Managing User Service Level Expectations

By Harris Kern's Enterprise Computing Institute

It is the objective of every IT organization to be the best service provider to their end-users. However, this is not possible without first understanding what service level or system availability is desired by end-users. When asked, the most likely response of users will be that the system is available *all* the time. We in IT know that this is not *economically* possible. This is why it is important that you learn to manage user expectations and ensure that it is at a realistic and practical level.

You need to explain to them that the cost for providing system availability gets higher and higher as more availability is needed. They have to realize that they have to be conservative in their specified availability hours, since these costs will be passed on to them somehow — either *directly*, as IT chargeback for services, or (as in most small-to-mid-sized companies) *indirectly*, as the IT organization takes a larger share of the corporate budget.

The Service Level Agreement

These consultations with users form the basis of what is formally called a Service Level Agreement between you as provider of IT services and the end-users as consumers of these services. You can choose to limit yourselves to a simple agreement that covers just system availability hours, or you can expand the agreement to include response time, help desk availability, new feature request turnaround time, and many other performance and quality issues. If you are starting from scratch, we recommend including just the system availability hours portion. Then, as the system becomes more stable and your IT organization matures, you can and should expand on that agreement.

This gradual approach to establishing a Service Level Agreement has many benefits, namely:

- **The users do not expect too much too soon** — The final judges of the IT organization's performance are the users, so it is crucial to manage their expectations. Satisfaction is directly related to expectation levels: set it too high and you will rarely get satisfied users. The lower you are able to set it, the easier it is for you to get very satisfied users. However, bear in mind that many users now are more IT aware and it is not uncommon for them to compare what your IT organization can deliver versus what other similar IT organizations provide their users.
- **It gives the IT organization time to improve on services** — This is an opportunity for the IT organization to be one step ahead of user requirements. It gives the organization a better feel for the resource demands associated with meeting availability requirements, and allows for better planning. Expenses-wise, this approach also spreads-out the costs to a more manageable scenario. So don't be a boastful IT organization and commit everything all at once.

- **It allows for a less demanding agreement** — Since users know that the agreement will be improved later, they will be more willing to settle for a realistic and easily achievable short-term target. The critical factor here is that users are made to realize that the Service Level Agreement is a work-in-progress that will be revisited from time to time.

Never commit what you know you cannot achieve. Agree on a target you can truly achieve in the short term, and establish a timetable for achieving higher system availability in the future. If possible first pilot the system availability target internally within the IT organization or with one small user department. Once you've demonstrated that you can meet your target, roll out the new service level standards throughout the rest of the organization. *If possible, commit only what you have already achieved before.*

Helping Users Identify Their Availability Requirements

It is normal for users to think that they know everything, so you would have to be diplomatic in helping them correctly understand their requirements. Explain to them that there is a scientific methodology for putting a numerical value to what they really need.

- **The first questions to ask users are: What are your scheduled operations? What times of the day and days of the week do you expect to be using the system or application?**

  Answers to this question help you identify the times your system or application must be available or accessible to end-users. Normally, the responses will coincide with the users' regular working hours. For example, users may primarily work with an application from 8:00 a.m. to 5:00 p.m. from Mondays to Fridays. However, some users may want or even need to be able to access the system for overtime work. Depending on the number of users who access the system during off hours, you can choose to include those times as your normal system operating hours. Alternatively, you can set up a procedure for users to request off-hours system availability at least three days in advance.

  When external users or customers access a system, its operating hours are often extended well beyond the normal business hours. This is especially true with online banking, Internet services, e-commerce systems and other essential utilities such as electricity, water, and communications. Users of these systems demand availability 24 hours a day, 7 days a week so that they can use it anytime and every time they need it.

  The critical success factor here is being able to poll representatives from the entire spectrum of possible users of your system or application.

- **The second set of questions to ask users is: How often can you tolerate system outages during the times that you are using the system or application? How about scheduled outages?**

  Your goal is to understand the impact on users if the system becomes unavailable when it is scheduled to be available, no matter how long the outage is. For example, a user may say that he can only afford two outages a month.

This answer also tells you if you can ever schedule an outage during times when the system is committed to be available. You may wish to do so for maintenance, upgrades, or other housekeeping purposes. For instance, a system that should be on line 24 hours a day, 7 days a week may still require a scheduled downtime for upgrading.

- **The final question to ask users is: How long can an outage last if one does occur? Will it make a big difference if we announce that the system is going down prior to it actually happening? What is the cost to you when the system is not available?**

  This question helps identify how long the user is willing to wait for the restoration of the system during an outage, or to what extent the outages can be tolerated without severely impacting the business. For example, a user may say that any outage can only last for up to a maximum of three hours.

  Often, a user will be able to tolerate outages longer if they are scheduled or announced beforehand, so try to find out also what is the ideal amount of advance warning desired.

  The objective here is to try to quantify the business losses if an outage occurs, so that corresponding investments needed to prevent such outages can be justified.

Availability Levels and Measurements

Based on the answers to the questions discussed in the previous section, we can specify which category of availability your users require:

- **High Availability**: System or application is available during specified operating hours with *no unplanned outages*.

  For example, the high availability criteria could be no outages between 8am to 5pm, Monday to Friday. In this example, scheduled maintenance is possible outside these availability hours (e.g. 12am system backups).

  When is an outage considered as pre-announced or not? Remember whose perspective matters — *the user's*. If you announce an outage an hour in advance, you might consider it planned, but your users may consider it unplanned, since they don't have enough time to adjust their work to cope with it. When the outage will occur and when the users are informed about it are also both important. For example, telling the users at 8:00 a.m. that a downtime will occur in eight hours is more acceptable than telling them at 5 p.m. that an outage will happen at 8:00 a.m. the following day, since the latter might give users no time to prepare unless they work overtime.

  High availability is the easiest availability level to achieve, since it still gives you room to schedule system downtimes, as long as you schedule them outside the committed availability period. For example, you can deliver high availability while retaining the ability to schedule nightly backups. You must, however, ensure that the system operates reliably during committed periods of availability. The challenge here is to eliminate problems, or at least make them transparent to users or less likely to affect system availability at the end-user's point of view.

- **Continuous Operations:** System or application is available 24 hours a day, 7 days a week with no scheduled outages.

  In this availability level, users want the system to be always available, but *will tolerate unplanned outages* due to problems. To achieve this level, you must implement techniques that make the system more reliable and eliminate dependence on scheduled maintenance work that would require system downtime.

- **Continuous Availability:** System or application is available 24 hours a day, 7 days a week with *no planned or unplanned outages*. Period of availability is the same as that of continuous operations (which is all the time), but no form of outage will be tolerated by the users.

  This level of availability is the most difficult to achieve and is normally demanded in critical systems that provide essential services to the general public, such as electricity, communication systems, and banking services such as Automated Teller Machines (ATMs). Internet service providers and e-commerce systems also need to provide Continuous Availability. Obviously, this level of availability is also the most costly to achieve. Users must be aware of this expenditure, and be willing to pay for it. One hundred percent continuous availability is almost impossible to achieve over a long period of time as we shall show below.

Quantifying Availability Targets

To quantify the amount of availability achieved, we calculate:

- **Committed hours of availability (A) —** The times during which the system is to be accessible to end-users, usually measured in terms of number of hours per month or any other period suitable to your organization.

  Example: 24 hours a day, 7 days a week = 24 hours per day × 7 days = 720 hours per month

- **Outage hours (B)** — Total number of hours of outage during the committed hours of availability. If the availability level committed is high availability, then count only unplanned outages. For continuous operations, count only planned or scheduled outages. But for continuous availability, total all outages, scheduled or unscheduled.

  Example: Nine hours of unscheduled outage due to server hard disk crash, 15 hours of planned outage for preventive maintenance

We calculate the amount of availability achieved as follows:

- **Achieved availability = ((A – B)/A) × 100 percent)**

  Example:
  - High Availability scenario: ((720 – 9 hours unplanned outage)/720) × 100 percent = 98.75 percent (remember that planned outages are tolerated)
  - Continuous Operations scenario: ((720 – 15 hours planned)/720) × 100 percent = 97.91 percent (here, unplanned outages are tolerated)
  - Continuous Availability scenario: ((720 – 24)/720) × 100 percent = 96.67 percent (no outages are tolerated)

When negotiating an availability target with users, make them aware of its implications in terms of how little allowance it would have with respect to outages in the system. Here is a table of availability targets versus hours of outage allowed for a Continuous Availability requirement.

Continuous Availability Target	**Hours of Outage Per Month Allowed**
99.99 percent	0.07 hours or 4.2 minutes
99.9 percent	0.7 hours or 42 minutes
99.5 percent	3.6 hours
99.0 percent	7.2 hours
98.6 percent	10.0 hours
98.0 percent	14.4 hours

It is important to realize that the cost to achieve higher availability targets rise exponentially. An increase of a percentage point in availability hours does not equate to the same percentage cost increase in IT spending.

When an outage occurs, time is needed to recover from outages. The length of recovery time is dependent on many factors such as:

- **Complexity of the system** — The more complicated your system is, the longer it takes to restart it. Hence, outages that require system shutdown and restart can dramatically impact your ability to meet a challenging availability target. For example, applications running on a large server can take up to half an hour just to restart when the system was shut down normally, longer still if the system had been abnormally terminated and data files have to be recovered.

- **Severity of the problem** — Usually, the greater the severity of the problem, the more time is needed to fully identify and resolve the problem, including restoring lost data or work done.

- **Availability of support personnel** — Let’s say the outage occurs after office hours. A support person called in after hours could easily take an hour or two simply to be onsite to diagnose the problem. Even worse is if the system is remotely located, where travel to that location may further prolong the downtime. You must allow for this possibility to occur.

- **Other factors** — Many other factors prevent the immediate resolution of an outage. Sometimes an application may have an extended outage simply because the system cannot be put offline since other applications are running on the same resources. Other cases involve the lack of replacement hardware by the system supplier, or even lack of support staff. We have seen many availability targets missed simply because a system supplier could not give due attention to the problem, and no backup system supplier existed.

Availability: A User Metric

We cannot over-emphasize that availability should be measured from *the user's point of view.* A system is available if the user can use the application he needs. Otherwise, it is unavailable. Accordingly, availability must be measured *end-to-end,* from computing resource all the way to the end-user interface.

Many IT organizations mistakenly believe that availability is simply equal to main server or network availability. Some may only measure the availability of critical system components. These are grave mistakes. A user may equally be prevented from using an application because his PC is broken, or his data is unavailable, or his PC is infected with a computer virus.

IT organizations that subscribe to this narrow availability mindset go through several stages of alienation from their users:

- **User unhappiness** is the first and least severe stage. Users simply express unhappiness with poor system availability. The IT organization may either recognize a problem or deny it, citing their host or network availability statistics as proof. Those who deny the problem's existence bring their organization to the next stage of user alienation.
- **User distrust** is characterized by user disbelief in much of what the IT organization says. Users may begin to view IT's action plans as insufficient, or view the IT organization as incapable of implementing its plans. They gradually lose interest in helping IT with end user surveys and consultations. IT organizations that can deliver on promises and provide better availability *from the user's point of view* can prevent users from moving to the next stage of user alienation.
- **User opposition** is the third stage of alienation. Here, users do not merely ignore IT plans. They begin to actively oppose them, suggesting alternatives that may not align with IT's overall plans. Users start to take matters into their own hands, researching alternatives that might help solve their problems.

  The challenge for the IT organization at this point is to convince users that the IT plan is better. The best way to meet this challenge is to conduct a pilot test of the user's suggested alternative, then evaluate the results hand-in-hand with users.

Unfortunately, we have seen many IT organizations react arrogantly, telling users to "do what you want, but don't come crying to us for help." These organizations find themselves facing the last stage of user alienation.

- **User outsourcing** is the final stage of user alienation. Users convince management that the best solution lies outside the IT organization. Outsourcing can take the form of hiring an outside consultant to design their system, going directly to an outside system supplier, or even setting up their own IT organization. At this stage, users have completely broken off from the IT organization, and reduced — if not totally eliminated — the need to fund it.

Other than user alienation, other serious side effects of insisting on narrow-minded availability measurements is as follows:

- **Failure to identify root causes of availability problems** — If only a few components are considered when system availability is evaluated, the root cause of the outage might never be found as it might be due to components whose availability is not monitored.

  As an example, a large banking IT organization kept on denying the existence of Automated Teller Machine (ATM or cash machines) problems by pointing out that their mainframe computer, network routers, and network connections are always online. They failed to observe that the ATM machines themselves were the problems.

- **Conflicts between IT divisions** — Many IT organizations usually delegate critical elements of their systems to individual groups within IT. Each then measures the availability of its assigned area, without correlating it with the availability of other areas. This leads to territorial disputes where one group blames others for poor system availability. "Don't blame my group! Our network was up 100 percent of the time..."

- **Expensive and ineffective remedial measures** — If you do not know what the root cause of a problem is, you'll probably spend money on the wrong solution. Or, you'll concentrate on improving only *your* assigned system component, without regard to overall system availability.

- **Inability to determine true system health** — Availability measurements of each component cannot easily be "added up" to reveal true system availability. Ninety-nine percent host availability + 99 percent network availability + 99 percent database availability is not equal to 99 percent system availability. Outages in each area usually occur at different times, and each outage in any component brings the entire system down. In this example, actual system availability can be anywhere from 97 percent to 99 percent.

Why do many IT organizations fall into the trap of measuring only a few system components and not actual end-to-end availability? There are two reasons.

- First, it is easier to measure a few system components. Few tools are available for analyzing and monitoring end-to-end system availability. Many tools measure network or host availability, but few actually check for application outages from the perspective of the user.

- Second, it is easier to achieve higher availability on a per component basis since outages rarely occur repeatedly on the same component. Outages for different components usually occur at different times but may all affect the availability of the system to the user, resulting in far worse availability statistics.

- Third, many IT organizations do not have an end-to-end view of systems, much less an assignment of responsibilities based on this view. Many responsibilities in an IT organization are distributed according to system components and not on the basis of user applications.

Measuring End-To-End Availability

To accurately estimate end-to-end application availability as experienced by end users, you must first thoroughly understand the system's configuration; all the components and resources used by the application, both local and remote; and the hardware and software components required to access those resources. Here is an example:

Sales Personnel Call Management System

Local resources	Sales personnel data; Call reports
Remote resources	Contact management data at each sales rep's computer
Hardware components	Personal computer, LAN adapter, LAN cabling, network switch, router, print server, network printer
Software components	Windows XP, Microsoft Access database, contact management software, call management application

The next step is to monitor all these components for outages. If outages are detected on multiple components at the same time, treat the outage duration as just one instance. To calculate end-to-end availability, add all the outages of each component. Then, apply the formula presented earlier in this chapter.

This approach sounds easy in principle, but taxing in practice? Definitely. That's why you need to automate measurement as much as possible. The simplest way is to use a tool that monitors availability of local and remote resources from a user's PC. This tool regularly attempts to get a response from the resources in question, and records times that critical resources are unavailable. More advanced tools can query an application for problems or execute certain tasks on the application. If the application fails, an outage is recorded. This approach does not identify the source of the problem, but the error condition may help support staff identify the cause.

There is a great demand for automated end user system availability monitoring tools — utilities that can be installed in user workstations that would periodically test the applications for availability. In the absence of such tools, you would have to resort to random sampling of users' availability experiences.

You won't get precise measurements of every user's availability experience. That's unrealistic. Do, however, recognize that users have an availability requirement you must pay attention to. Don't get too dependent on technical measurements for rating your performance. Ultimately, what matters most is that users are happy with the service that the IT organization provides.

Remember that the discussion in this section focuses on how availability is affected by hardware or software outages. Again, this is not the only factor by which a user judges system availability. The system may not be experiencing an outage, but if it is running too slowly, a user may give up waiting and consider an application unavailable. Hardware and software outages, though, make up the majority of the reasons for unavailability.

Summary

The IT organization must understand the level of availability users require, and users must understand the costs of achieving these targets. Often, IT loses users' trust and confidence by dictating availability targets without proper end user consultation. Conversely, users often make unrealistic demands on IT, failing to recognize the cost implications of such a requirement.

Of all the availability levels discussed, Continuous Availability is the most challenging and expensive to provide. More often than not, users are willing to settle for High Availability but with committed hours of operations as close as possible to 24 hours a day, 7 days a week.

Lastly, availability is a user metric, which means that we must measure it from the point of view of the user's experience. Most IT organizations that lose the support of their users have failed to recognize this, focusing instead on the availability of only a few critical components.