Realtime
publishers
"Leading the Conversation"

The Definitive Guide™ To

Service-Oriented Systems Management

sponsored by

altiris®

Dan Sullivan

Copyright Statement

Chapter 6: Implementing Systems Management Services, Part 2: Managing Service Delivery

Service delivery is a complex mosaic of multiple processes and procedures that are required to introduce, manage, and develop information services. The previous chapter examined how service delivery is deployed with processes such as incident management, configuration management, change management, and release management. This chapter continues with service delivery, but turns your attention to management.

The deployment step focuses primarily on executing procedures to keep IT operations running smoothly and adapting to the changing needs of users. Management is more about planning, monitoring, and adjusting. In particular, this chapter will address:

- Service-level management
- Financial services management
- Capacity management
- Availability and continuity management

These aspects of service delivery have a common characteristic: These activities address the long-term IT needs of an organization. The deployment operations discussed in the previous chapter are performed to ensure the proper day-to-day function of IT systems. If those activities were not practiced, the consequences would be seen rather quickly. Poor management, however, can continue for some time before the full effects are noticed. Nonetheless, proper systems management must address both the short-term and long-term needs of IT services.

Service-Level Management

How much service-level management is necessary? If you had to distill service-level management to its most essential form and present it as a question, that would be it. IT managers must understand how much storage, computing resources, network bandwidth, training, and time from developers, quality control specialists, and a host of other IT services are needed by users. In addition, IT managers must know when these resources are needed. For example, will the data warehouse extraction, transformation, and load process run during normal business hours or at night? If it is during the day, the application hosting the data will need to accommodate its normal workload plus export potentially large amounts of data. The network must also accommodate the additional load. If the data warehouse is loaded in the middle of the night, the demands on both the application and the network would be less. Something as simple as when a process runs can have a major impact on the performance of that process.

Throughout this guide and in best practices and control frameworks, such as those documented in ITIL and COBIT, there is a major emphasis on formalizing processes and procedures. This idea applies to service-level management as well. The mechanism most commonly used in service-level management is the service level agreement. An SLA is essentially a contract between business units and IT service providers, such as in-house IT departments or outsourced service providers. These agreements typically define the scope and levels of service provided.

Service-level requirements define the functional requirements that the business needs in order to carry out its functions. They also entail, although this is not necessarily explicit, the need for communications between business units and service providers. Business unit requirements are rarely static and even in the best situations, requirements may not capture all nuances of a business unit's needs. Success for both business units and services providers require that communications do not stop once requirements are defined.

Requirements will vary according to business objectives, but several topic areas are common to most business applications:

- Application functions
- Training
- Backup and recovery
- Availability
- Access controls
- Service catalog and satisfaction metrics

Each of these areas should be documented in service-level requirements.

Application Functionality

Within the section on application functionality, the project sponsors should define what the system is to do. It is important to avoid becoming mired in implementation details at this point. The goal is to define *what* the system should do—not how it should be done. For example, if an application must be accessible from both traditional Web clients and mobile device clients, state that purpose; there is no need to include design considerations, such as whether to use Handheld Device Markup Language (HDML) or Wireless Markup Language (WML).

It is important to think of application functionality in terms of business tasks, such as:

- Providing customer support
- Verifying inventory
- Reporting on the status of operations
- Confirming customer orders

The specific functions might cover a broad range of options and they should be as inclusive as possible when dealing with service requirement agreements, especially if part or all of the service will be outsourced.

Training

Training should address both service use and service administration. User training is relevant when an application as well as network and hardware infrastructure are included as part of the provided service. For example, if an outsourcing firm is providing a CRM service that has never been used by the customer, end user training should be included in the scope of the requirements.

Administrator training is almost always required, even when most of the systems infrastructure will be managed by the IT department or an outsourcing firm. Application administrators are often responsible for implementing and maintaining users, roles, and access controls as well as organization-specific configurations related to the application's functions.

Backup and Recovery

Backup and recovery is a crucial element of service requirements. Applications and hardware fail. Sometimes an error in an application will corrupt information in a database; in other cases, hardware will fail and a disk becomes unusable and data must be restored from backups. When defining backup requirements, include the following criteria:

- Recovery time objectives
- Recovery point objectives

Recovery Time Objectives

Recovery time objectives define the acceptable length of time that an application or service can be down. For example, mission-critical applications—for example, a CRM—might have very short recovery time objectives, such as a few minutes. In such cases, failover servers or servers with redundant subsystems are typically used. In other cases, such as a data warehouse and reporting application, the system could be down for as long as a day without adversely impacting operations.

A general rule to keep in mind is that the shorter the recovery time objective, the higher the cost. For example, backups can be restored more quickly from online disk arrays than from offline optical disks, but the cost of disk arrays is higher than that of the offline storage media.

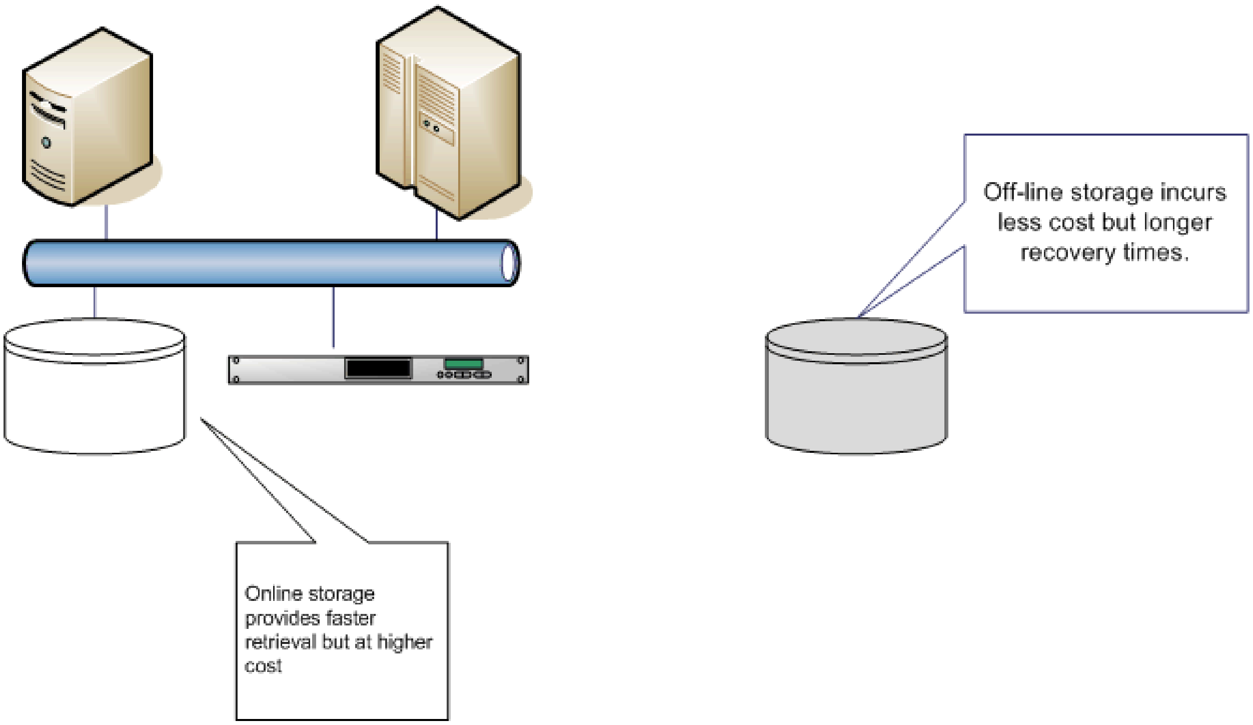
Off-line storage incurs less cost but longer recovery times.
Online storage provides faster retrieval but at higher cost

Figure 6.1: Although implementation details are not part of service requirements, the cost of different options can be a factor.

The time to recover is only one aspect of recovery criteria; another is specifying what it is that will be recovered.

Recovery Point Objectives

If a server fails at 11:00am on a busy business day and backup files and a standby server is available, operations could be back online by early afternoon. But what will be restored? Will all of the changes made up to 10:59am be recovered? Will only changes made before the prior day be recovered? The answer to these questions is determined by the recovery point objective.

Formally, a recovery point objective is the recoverable state of a system at some point in time before a failure. The goal of the recovery time objective is to define the maximum time that a service is unavailable—the recovery point objective defines the maximum amount of data that can be lost due to a failure. Again, like recovery times, the better the recovery point, the more likely it is to increase costs.

For example, consider a CRM application that manages customer account data that uses only daily backups. If there were a failure at 11:00am and a backup had been performed at 3:00am, the recovery point is effectively the previous business day. Any work done before 11:00am on the day of the failure would be lost. Similarly, if the failure occurred at 4:59pm, a full day's work would be lost. This is not suitable for many situations.

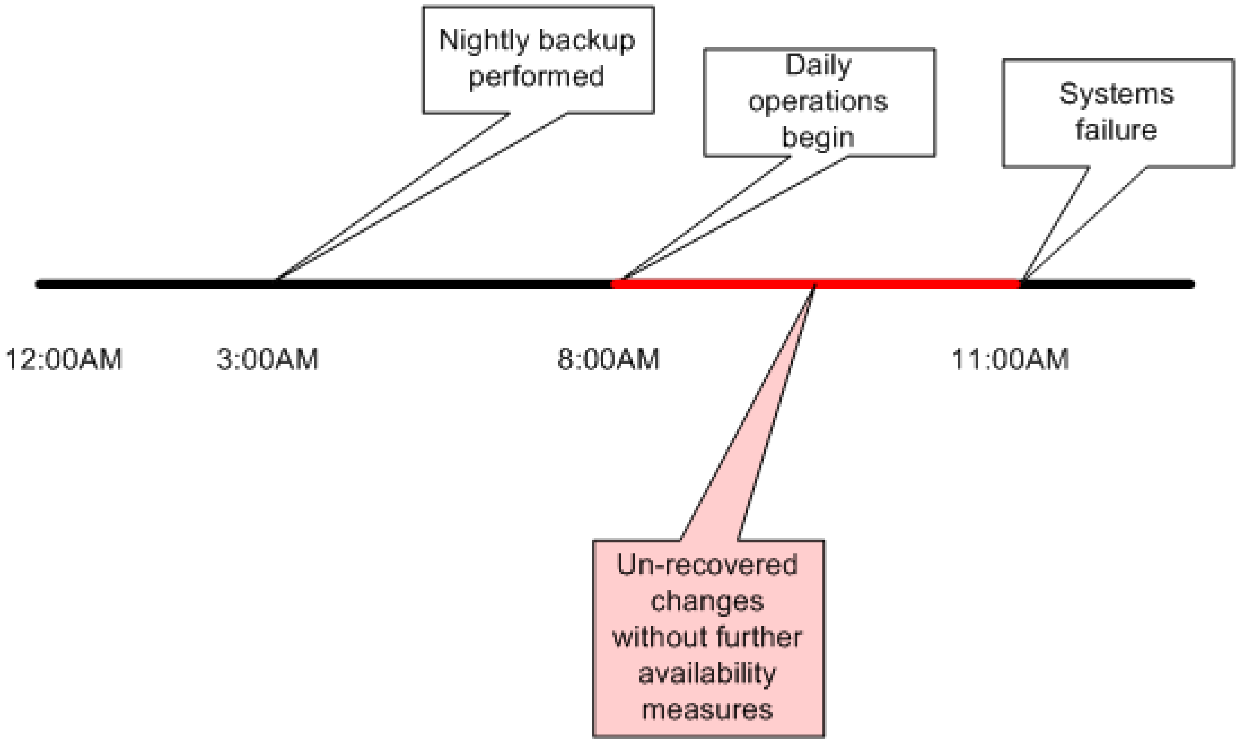

Figure 6.2: Backups without further availability measures can leave work performed since the backup vulnerable to system failures.

Fortunately, there are availability procedures (discussed in more detail later) that can provide recovery up to the point of failure. These tend to require more complex software, but they are often used in applications designed for midsized and large enterprises.

Availability

Availability criteria answer the question “What is the tolerance for downtime with this service?” The answer is obviously closely related to requirements for backup and recovery but also focuses on the tolerance for downtime. Although backup and recovery procedures are designed for particular recovery times and recovery points, availability addresses the question of how frequently the business is willing to tolerate downtime.

For example, a server might go down at 11:00am and be back by 1:00pm the same day and still meet backup and recovery requirements. If the same server goes down every day, it might still meet the recovery objectives, but the business users are not likely to tolerate a system that is down 2 hours of the day. The key questions with regard to availability in service requirements are:

- How long can the system be down?
- How frequently can the system go down?

The length of time a system can be down is expressed in minutes, hours, or days. The amount of disruption in the ideal world is virtually none, but in reality, the cost of countermeasures to prevent downtime must be balanced with the benefits.

The rational choice is to allocate resources to availability measures until the cost of those measures exceeds the expected cost of the corresponding downtime. For example, if a high-availability solution is available for $50,000 and promises to keep downtime to less then 5 minutes, and another solution is available for $5000 but reduces downtime to 1 hour, which solution is better? The answer depends on the lost revenue or cost of being down. If, for example, the business would loose $10,000 if the system were down for 1 hour, the less expensive solution is a better choice.

The frequency is usually expressed as a percentage of total uptime. For example, if a system should be available 24 hours a day, 7 days a week, and the requirement is 99 percent uptime, the system could be down 87.6 hours, or more than 3 days per year. Table 6.1 shows the amount of downtime allowed under several requirements.

Availability Rates

Total Hours per Year: 8760

Availability Requirement	Hours Down per Year
98.00%	175.20
99.00%	87.60
99.50%	43.80
99.90%	8.76
99.95%	4.38
99.99%	0.88

Table 6.1: System availability requirements are often expressed as a percentage of total possible hours a service could be available.

Additional areas that should be addressed in service requirements are security and access controls.

Access Controls

Access controls dictate who can do what with information assets. When developing service agreements, access controls tend to be high level, unlike application-specific access controls, which can be detailed and fine-grained. Access controls are dependent on three mechanisms:

- Identification and identity management
- Authentication
- Authorization

Identification and Identity Management

The purpose of the identification phase is to indicate to an access control system who a user claims to be. A username, a device such a smart card, or a biometric measure can be used for identification. Identification does not necessarily provide evidence for who you are; for that, you depend upon authentication mechanisms.

An identity record is associated with each user of a system. In some cases, these are relatively simple structures with little more than a username. For example, a basic UNIX password file includes the username, encrypted password, user number, group number, home directory, and the name of the shell program to run when the user logs in. In other cases, the structures are much more elaborate.

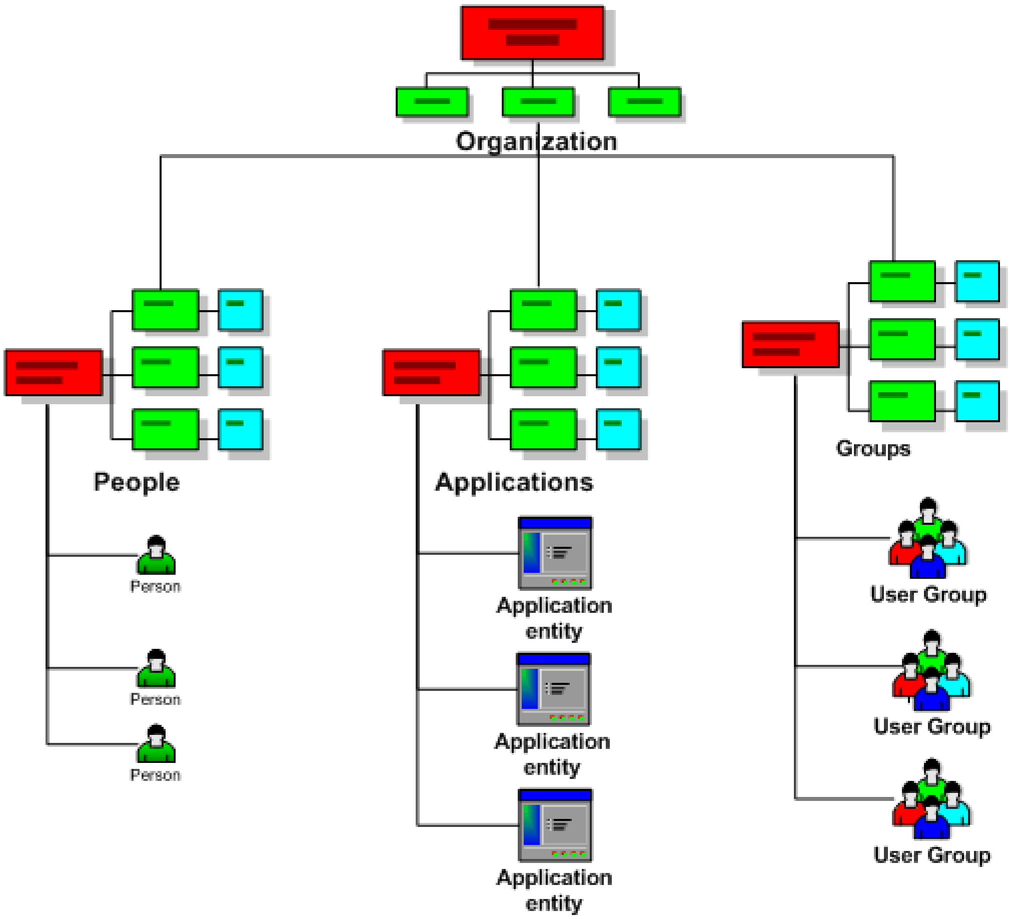

Figure 6.3: LDAP directories maintain identity and organization information that can be leveraged for access control management.

Active Directory (AD) can be used to store detailed information about users, including organizational role, phone numbers, email addresses, public keys (when a public key infrastructure—PKI—is in use) and other identifying information. AD and other types of network directories can store information about other structures and assets on a network:

- Organizations and organizational units (OUs)
- Organizational role
- Groups of users
- Devices
- Applications

An advantage of directory-based identity management is that applications do not need to maintain separate databases of user information. Centrally managing basic user information still allows applications to tailor authentication rules to their particular needs.

Authentication

Authentication is the process of proving one's identity. Passwords are commonly used for this purpose, but with all the well-known limitations of passwords, other techniques have become more popular. Some other methods for authenticating to systems are:

- Smart cards
- Fingerprints
- Palm scans
- Hand geometry
- Retina scan
- Iris scan
- Signature dynamics
- Keyboard dynamics
- Voice print
- Facial scan
- Token devices

The biometric methods also serve as identification methods. The objective of authentication is to grant access to a system only to legitimate users. Because a single method, such as a password, can be compromised, systems with high security requirements may use multi-factor authentication.

With multi-factor authentication, two or more authentication methods are used to verify a user's identity. This method often combines multiple types of mechanisms, relying on, for example, something the user knows (such as a password), something the user has (such as a smart card), and something the user is (such as a unique fingerprint). Once a user has been identified and authenticated, the user is granted access to the system. What the user is able to do with that system is dictated by the authorization rules defined for that user.

Authorization

Authorizations are sets of rules applied to users and resources describing how the user may access and use the resource. For example, users may be able to log into a network and access their own directories as well as directories shared by all users in their department. The following list highlights considerations for defining authorization requirements with regards to SLAs:

- Who are legitimate users of the system or network?
- How is their identity information maintained?
- How are users grouped into roles?
- How are privilege assignments to roles managed?
- Will the auditing capabilities of access control systems meet the audit requirements of the customer?

As a rule, service level requirements should focus on what a service should provide, not how it is provided—but access controls can be an exception to that rule. For example, if an organization has invested in an identity management system, with constituent LDAP or other directories, single sign-on (SSO) services, and authorization services, then an SLA can, and should, dictate the use of that system. Sometimes you cannot avoid having to manage multiple access control systems; in those cases, you should at least try to minimize their number. Another aspect of service level management that spans multiple areas of IT is maintaining a catalog of IT services and service metrics.

Service Catalog and Satisfaction Metrics

The service catalog is a list of all services provided by an IT organization. This list can include both in-house and outsourced services. The catalog should include details about the service, including:

- Applications used to support the service
- Devices used to support the services
- Description of availability of the service
- Costs of the services
- Dependencies or other restrictions on use of the service

In addition to keeping track of what services are provided, service management best practices dictate that you measure how well services are provided. Some common measures include:

- Response time
- Time to resolution
- Number of incidents by category
- Unit costs, such as cost of service per user or number of users supported per device
- Direct customer satisfaction surveys

These metrics, especially when applied to a service desk, should be categorized by priority. A security breach that leaves a database of customer information vulnerable is an urgent incident that must be responded to immediately. A problem that delays or inconveniences without disrupting core business operations might be categorized as normal and addressed on a first-come first-served basis.

Service level management spans the range of IT services. It includes some elements of business continuity planning, security services, and capacity planning. Successful service level management begins with well-defined SLAs that identify user needs in several areas as well as the level of service users can expect in each of those areas. In addition, service managers are expected to measure performance and maintain and improve service. Of course, all this management, along with the rest of IT resources, cost money.

Financial Management for IT Services

Financial management for IT services is challenging. Customers' needs change, technologies are constantly evolving, and there are often multiple ways to solve problems—each with their own advantages and risks. In addition to managing today's operations and projects, managers must plan for future needs. Four common tasks in IT financial management are:

- Cost accounting
- Forecasting
- Capital expenditure analysis
- Operations and project financial management

These are fundamental financial management tasks and not limited to IT operations.

Cost Accounting

Cost accounting is the process of allocating the cost of providing service to the recipients of that service. It sounds like a reasonable method—you pay for the services you use. When you are buying relatively simple products, like a spindle of DVDs for backing up files, you can go to an office supply store, pick out the right product for your needs, and pay the pre-set price. Why can't you do that for all IT services? The answer is, as it often seems to be, that the simplified models of how things work start to break down when you get to real-world scenarios that are more complex than example cases.

Competing Requirements

Consider the following scenario: An IT department provides a backup service. Some departments have relatively simple backup and recovery requirements, while others are more involved. The remote sales departments need their network file servers backed up at night, and their backups should be kept for a week. After that, the backup media can be reused. A full backup on the weekends and incremental backups during the week are sufficient. The total amount of data backed up is in the 100s of gigabytes. Another department has a terabyte-scale customer management database that must be backed up every day. Audit requirements necessitate keeping a month's worth of data. Recovery time requirements are so tight, there can not be too many incremental backups between full backups (recovery from a single full backup is faster than recovery from a full plus several incremental backups). To meet the requirements of the department using the customer management system, the IT department has to buy a high-end backup tape solution with robotic components and high-speed tape devices. How should the costs be allocated?

Cost Allocation

This is where it gets complicated because there are a number of options. The remote office has minimal requirements that could have been fulfilled without the high-end solution needed by the customer management department. The remote office could be charged a rate competitive with what they would pay for an outside service to provide the service. In this model, the remote office does not incur additional cost because of the needs of another department.

Another model allocates the cost based on units of service provided. If the customer management department uses 95 percent of the backup storage and the remote office uses 5 percent, the former is charged 95 percent of the total cost of the service and the latter is charged 5 percent. In this case, the remote office is paying a premium for high-end hardware it does not need.

A third option is to use a graduated schedule of charges so that the customers using the least amount of service pay less than the customer that forces the IT department to use high-end solutions.

Yet another option is to have two backup solutions: one for low-end needs and one for high-end needs. Each department would pay the full cost of its solution. Unfortunately, this could be the most expensive option because two types of systems would have to be purchased and maintained. This is the least rational solution for the organization as a whole.

In practice, the second option, allocating costs based on usage, is the easiest to implement. It avoids the competitive analysis required by the first option, the political battles associated with a graduated scale, and the extra expense of the two-solution option.

Implementing Charge Backs

Once costs have been allocated to users, the IT department can charge those departments for the service. These costs to the customers are known as charge backs. Some IT organizations operate as an internal service that recovers all their costs from customers. There are definitely advantages to this setup. For example, when costs, including IT costs, are allocated properly, managers can determine which products or services are profit makers and which hurt profits. However, when costs are distorted, the true cost of a single product or service cannot be determined.

Care must be taken when using charge backs in profit calculations. If a department is not allowed to seek competitive bids for a service, such as backups, should the department be held responsible for having to pay the higher prices for an internal service? Cost accounting attempts to provide accurate measures of the true costs of services, however, in practice, the complexities of providing a service and the global considerations of the organization as a whole can introduce distortions that are not accommodated by basic accounting methods. Another challenge facing managers is trying to estimate future needs.

Forecasting

Forecasting is as much an art as a science. It is fundamentally about estimating the cost of future services, which include several types of costs, such as:

- Labor, including both employees and contracted labor
- Capital expenditures
- Lease costs
- Service contracts, such as maintenance
- Consulting

Within the areas of forecasting that can be standardized, a few general observations can be made:

Forecasting at the Appropriate Level

First, forecasting should be detailed enough to take into account significant differences in costs without encumbering the process with too many details. For example, when forecasting labor costs, do not simply use a head count and an average cost per person; labor rates vary and the forecasts should reflect that. For example, a service desk technician and network operation technician could be grouped into the same general pay level, and database administrators and project managers are grouped into another. This way, if changes are needed to the forecast, for example, another 10 service desk technicians are added, the forecast can more closely reflect the actual costs.

Differing Patterns of Cost Growth

Keep in mind that some costs tend to grow gradually and continuously and others have jumps in costs. Adding more disk space to a storage array will have a relatively linear growth rate; adding one disk might cost X, adding two disk costs 2X, adding three disks cost 3X, and so on. Other costs, especially labor costs, have growth rates that look more like steps rather than steady inclines.

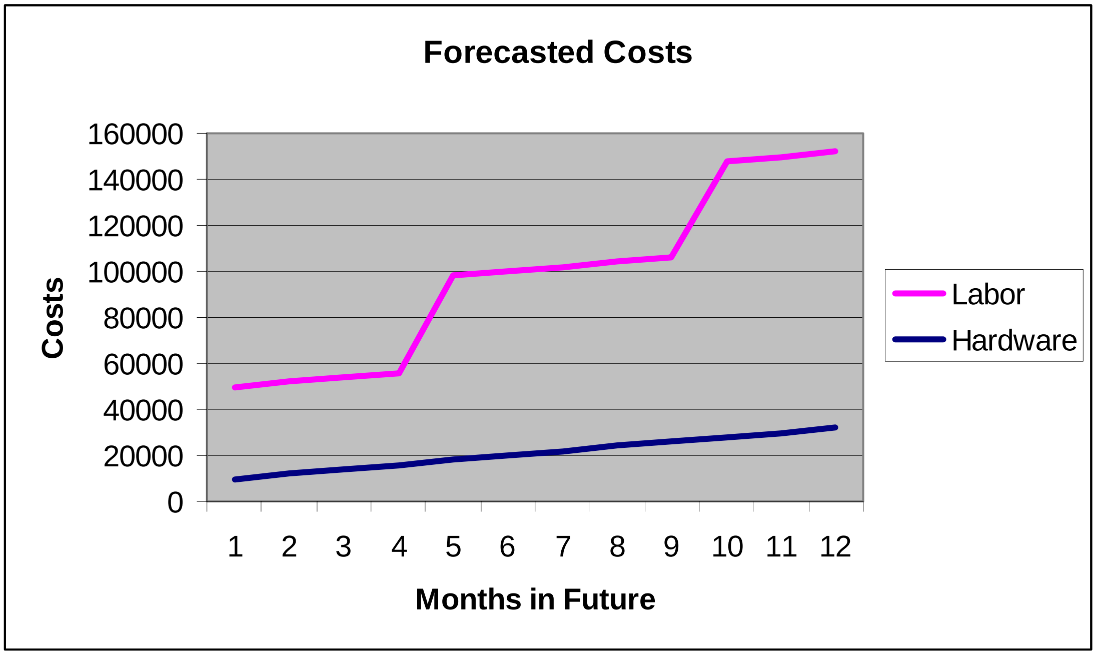

Figure 6.4: Patterns of growth in cost can vary, some are continuous and others are more step-like.

For example, the number of servers can increase for a while before an additional systems administrator will have to be added to the staff. However, the total cost of adding that one server that necessitates hiring another administrator is far higher than the cost of adding the previous server. This interaction among resource types must be accounted for when forecasting.

Accounting for Cash Flow

In some organizations, annual budgets are established and funds are available immediately for expenditures. For example, in a government agency with an approved capital budget, the funds are generally available for use once all the approvals are in place (assuming no further restrictions on the funding). In other organizations, especially businesses, plans may be subject to cash flows.

Cash flow, essentially the money coming into a business minus the funds going out, can vary over time, and expenditures must be timed to occur after sufficient cash is on hand. For example, if the IT department plans to purchase additional servers and hire a new systems administrator, the business needs cash on hand to pay for the hardware and meet payroll. When forecasting, consider the timing of cash flows in the business, especially if your business is subject to seasonal variations.

When forecasting, it helps to distinguish types of costs and how their growth patterns vary. It is especially important to watch for costs that introduce jumps in the total cost of a project or operation as well as the timing of expenditures that should be staged according to expected cash flows in the organization.

It should also be noted that forecasting for operational expenses, such as labor, leases, and small equipment, requires a different type of analysis than major expenditures for equipment with multi-year life spans. Those large expenditures warrant a more investment-oriented approach known as capital expenditure analysis.

Capital Expenditure Analysis

Capital expenditure analysis focuses on the purchase of equipment with relatively long lifetimes (in the IT world, this period seems to be about 3 years, give or take a year). These purchases are essentially investments, and questions arise about these investments, just like any other. How much is this investment worth in today’s dollars? What kind of return on investment (ROI) can you expect? Which piece of equipment is the better investment, A or B? Fortunately, there are well-established methods supporting capital expenditure analysis. Three commonly used calculations are:

- Net present value (NPV)
- ROI
- Internal rate of return (IRR)

These measures can be used separately or together to help formulate a decision about a particular investment.

NPV

The NPV of an investment is a measure, in today’s dollars, of the value of future savings or returns due to an investment made today. To determine the value of future savings or returns, you must take into account the present value of money. For example, if you were given the choice of receiving $1000 today or $1100 dollars one year from now, which choice would maximize your revenue? That would depend on the interest rate of money in the open market. If the interest rate is 5 percent per year, then having $1000 to invest today would yield $1050 in one year; the better investment would be to wait and receive the $1100 in one year.

The interest rate used in this calculation is known as the discount rate. It is used to determine the relative value of an investment. The NPV calculation takes into account the fact that savings or returns accrue over time, and it uses the discount rate to account for changes in the value of money over time.

Let's look at an example to see how this works: The IT department is considering investing in a new database server to replace two existing servers. The cost of the database server is $50,000. The department estimates that it will save $15,000 per year in maintenance, service contracts, and licensing costs. Will the investment in a new server save money in the long run?

To answer that question, you use the formula for calculating NPV. Assuming the useful life of the database server is 3 years, the formula for NPV is:

Amount saved in Year 1 / (1 + Discount Rate) +

Amount saved in Year 2 / (1 + Discount Rate)2 +

Amount saved in Year 3 / (1 + Discount Rate)3

Assuming a 5 percent discount rate, the calculations are shown in Table 6.2.

NPV Calculations

Year	Amount Saved	Discount Rate	Value per Year
1	$15,000	0.05	$14,286
2	$15,000	0.05	$13,605
3	$15,000	0.05	$12,958
		NPV:	$40,849

Table 6.2: Example NPV calculations.

In this example, the NPV of the investment is $40,849, less than the $50,000 investment. Unless there are other reasons to make the investment, the organization would be better off keeping the $50,000 in the bank than spending it on the server. Another commonly used measure in capital expenditure analysis is ROI.

ROI

ROI is a commonly used measure for a number of reasons; ROI

- Takes into account the total cost and benefit of an investment
- Is expressed as a percentage, not a dollar amount, so it is easy to compare ROIs for different investment options
- Is well known, perhaps in large part to the first two reasons

ROI is a calculation that takes into account the present value of future savings (like the NPV calculation), increased income generated by the investment, and the initial costs of the investment.

In the NPV calculation, you started with the amount saved in a given year. With the ROI calculation, you start with the net benefit of an investment for a given year. The formula for net benefit is:

> Net Benefit = Savings due to Investment + Increased Revenue due to Investment – Recurring Costs

The net benefit fits into the ROI formula which is similar to the net present value formula. For a three year period, the ROI formula is:

> [Net Benefit in Year 1 / (1 + Discount Rate) +
>
> Net Benefit in Year 2 / (1 + Discount Rate)2 +
>
> Net Benefit in Year 3 / (1 + Discount Rate)3] / Initial Costs

Let's use the formulas in an example. An organization is considering an investment in a network security appliance. The appliance will allow the IT department to retire or repurpose two servers running content-filtering and antivirus software. The appliance requires less time to administer than the two servers currently running security countermeasures, so there will be some savings on labor. The appliance will also filter traffic faster, allowing for the rollout of new Web-based services expected to generate additional revenue in the future. What is the ROI?

Start by calculating the net benefit for each of the next 3 years as shown in Table 6.3.

Year	Savings	Additional Revenue	Recurring Costs	Net Benefit
1	$10,000	$10,000		$20,000
2	$20,000	$40,000	$5000	$55,000
3	$10,000	$80,000	$5000	$85,000

Table 6.3: Net benefit calculations for security appliance investment.

The savings are due to expenses that would be incurred if the existing servers and applications running on those servers are kept. Year 1 and year 3 consist of software license costs, administration costs, and routine maintenance costs. Year 2 includes those as well as several hardware upgrades or replacements expected based on mean time between failures (MTBF) of several of the server components.

The additional revenue is due to the fact that a new service can be offered because of the higher throughput available from the security appliance. The first year will consist primarily of a small pilot program and initial marketing efforts. Projections for the plan estimate significant growth starting in the second year and continuing into the third year. The recurring costs are the costs associated with maintenance. These include minimal administration charges as well as maintenance fees charged by the appliance vendor. The net benefit is calculated according to the formula shown earlier.

You can now move on to the ROI formula. Assuming a 5 percent discount rate, and an initial cost of $25,000 the ROI is:

$$[\ \$20{,}000 / (1 + 0.05) +$$

$$\$55{,}000 / (1 + 0.05)^{2\ +}$$

$$\$85{,}000 / (1 + 0.05)^{3}\ ] / \$25{,}000 = 569\%$$

This is clearly a good investment option, due largely to the increased revenue enabled by the new appliance and, to a lesser degree, to the savings of the expenditures in the second year to maintain the existing servers. ROI is a broadly understood and easily used calculation. Another capital expenditure calculation that allows for comparison between projects is IRR.

IRR

IRR is a percentage, and is thus similar to the ROI rate; however, IRR does not depend on knowing, or estimating, a discount rate. Rather, IRR calculates the discount rate for which the NPV of an investment is zero. The advantage of this approach is that it is easy to compare two projects to determine which is a better investment regardless of the size of the investment.

IRR is an iterative calculation that starts with the initial cost of an investment and subsequent revenues or savings generated by the investment. Microsoft Excel provides a built-in function IRR as well as a modified version of IRR, called MIRR, that address some of the shortcomings of IRR in some situations.

Some financial analysts have questioned the use of IRR in capital expenditure analysis because, they claim, it makes some poor investments look better than they actually are in some cases. See John C. Kelleher and Justin J. MacCormack's "Internal Rate of Return: A Cautionary Tale" at http://www.cfo.com/article.cfm/3304945/1/c_3348836 for more information.

Capital expenditure analysis is an important part of IT financial management. Like forecasting, it helps formulate plans and budgets for operations and projects, which, in turn, need to be financially managed on a day-to-day basis.

Operations and Project Financial Management

Within the day-to-day operations of IT departments, the main activities are roughly divided into two types: operations and projects. Each has particular needs when it comes to management.

Operational Management Issues

Operations management is focused on the standard operating procedures in place within an organization. These can include:

- Performing backups on servers and client devices
- Monitoring network performance
- Reviewing audit logs
- Granting privileges to users and provisioning user accounts
- Populating the data warehouse and generating management reports
- Performing service desk support
- Responding to incidents such as hardware failures
- Installing patches and other software upgrades

These tasks are part of the relatively stable set of operations that constitute the IT support for running an organization. The issues most relevant to managing operations are:

- Staffing and training
- Meeting service delivery and other performance measures
- Tracking budget expenditures
- Seeking additional funding for unanticipated costs

As these tasks are done repeatedly, they are often, or at least should be, defined within standard operating procedures. There are, however, times when IT has to perform tasks outside of the routines defined for day-to-day operations. When these tasks are substantial and sizeable enough, they are managed as projects.

Project Management

Project management is a well-defined practice for achieving a set of one-time objectives, such as developing an application or migrating a service, such as email, from one platform to another. Unlike operations, specific projects are usually not repeated. However, the nature of projects is sufficiently similar to warrant a set of best practices. The following sections summarize the core activities within project management, especially as they relate to financial management.

Project Management Tasks

The main project management tasks are as follows:

- Planning project work—This task entails determining the exact nature of project deliverables (for example, a functioning program), the delivery schedule, and the resource required to accomplish the task.
- Performing risk analysis—Like risk analysis for IT management in general, the objective of this task is to identify risks and their likelihoods, and mitigate those risks when possible.
- Estimating and allocating resources—Different skills are needed at different times of a project; to operate efficiently, projects should staff only resources as they are needed. Similarly, hardware resources should be allocated as needed but with sufficient time to detect and adjust for unforeseen dependencies and incompatibilities.
- Assigning tasks and directing activities—These are the day-to-day items that project managers tend to focus on. The objective is to keep on schedule and to address problems early.
- Tracking and progress reporting—Another job of the project manager is to report the status of the project to management and seek help when problems cannot be resolved by just the project team; for example, when a business partner fails to meet a contractual agreement, thus putting the project deliverable and schedule in jeopardy.
- Performing post-project analysis—The objective of this task is to learn from the projects, especially from mistakes that may be avoided in future projects.

Over time, a number of project documents have emerged as part of project management best practices that help to control and document the state of projects as they execute.

Project Management Documentation

As projects are one-time activities often requiring skill sets from groups or departments that might not regularly work together, it is important to have a well-defined procedure for communicating expectations about the goals of the project, controlling the execution of the project, and reporting on it status. The following list highlights the core documents used for these purposes:

- Project charter—This charter provides a definition of the scope of the project, its objectives, and its lifespan.
- Business justification—This document provides the argument for pursuing the project. It may include ROI or other investment analysis measures.
- Work breakdown structure—This document is a precursor to a project plan that lists the tasks to be accomplished and the activities that will have to occur to accomplish each task.
- Risk management plan—This plan documents the risks to the project, the likelihood of each risk, and a mitigation strategy for each high-impact and highly likely risk.

> ✎ Not all identified risks must have a mitigation strategy; it is best to focus on those that could be the most disruptive.

- Project plan—This document is a combination of a resource management plan, a project schedule (including Gantt charts), and task assignments.
- Status reports—These are often short (one page) summaries of the state of the project. The reports note what was scheduled to be accomplished in the reporting period (typically one week), what was actually accomplished, planned work for the current reporting period, explanations for variations from the schedule, and a list of issues requiring management attention.

Project management presents challenges not found in operations management. At the same time, a well-developed set of best practices exist for managing projects. Anyone responsible for project management is strongly encouraged to use them.

> 🕮 For more information about best practices in project management, see resources at the Project Management Institute, http://www.pmi.org/.

Financial and service level management are major parts of service delivery management. More narrowly focused but still essential areas of service level management are addressed in the remainder of this chapter.

Capacity Management

Capacity management is the practice of understanding the demands for IT resources, such as computing, memory, storage, and network bandwidth, and ensuring adequate resources are in place when they are needed. This is done in three ways:

- Performance management
- Workload management
- Application sizing and modeling

These are closely related but address the needs of capacity management in slightly different ways.

Performance Management

Performance management entails monitoring systems to ensure resources are used efficiently, to detect trends in the growth or reduction in the needs for particular resources, and to identify performance bottlenecks. When a performance problem occurs, the key to resolving the problem is identifying the point in the process that is causing the slowdown.

Consider a customer management system that generates reports on sales activity. The company has been growing and sales activity is increasing; at the same time, the report-generation process is taking longer and longer, out of proportion with the growth in sales activities. The causes of the problem could be:

- Insufficient memory in the database server
- Insufficient bandwidth on the network
- Poorly coded SQL within the database application that does not scale well
- Insufficient CPU capacity for the number and size of the reports

It is critical to identify the bottleneck. If the problem is poorly written SQL code, adding more processors and memory may reduce the problem; however, this option will incur a significant expense and will probably work only for a short time. If the problem is insufficient memory, the CPU is probably not being used to capacity; adding faster or additional CPUs will not reduce the problem. Still another possible solution is to adjust the overall load on the system.

Workload Management

Workload management entails understanding the full set of processes that must be run, their dependencies, and their resource requirements and scheduling jobs and resources to maximize the use of computing resources. The first step to workload management is identifying the resources needed by each job. Some will require large amounts of bandwidth but little CPU, such as transferring data for a data warehouse load; others will be both disk and memory intensive, such as sorting large data sets for reports.

The second step is to schedule complementary jobs so that the contention for a single resource is minimized. Assuming there are not linear dependencies between the jobs (for example, job A must finish before job B starts), processes with different resource requirements should be scheduled together. Another rule is to schedule jobs early when there is a dependency on them by multiple other jobs. This method maximizes scheduling options of the later jobs. The other core process in workload management is monitoring. This should focus on both current performance and trends in growth or reduction in the need for particular resources. Another area of capacity management entails analyzing the needs of new applications.

Application Sizing and Modeling

The purpose of application sizing and modeling is to understand the capacity demands of an application before it goes online. It can be far more cost effective to understand the computing, storage, and bandwidth requirements before investing in hardware than after. The application modeling process should take into account several factors:

- The types of processes in the applications, such as CPU-intensive calculations or data-intensive database queries
- The relative frequency with which these different types of jobs will be executed
- The number of users and the times they will execute jobs on the system
- Outside constraints, such as time restrictions on when jobs are executed so that dependent jobs can meet their service level schedules

Application sizing and modeling entails elements of forecasting, so it will not yield certain results. When dealing with application sizing, it is best to plan for ranges of uses (for example, 10 to 100 users with moderate reporting demands or 2000 to 3000 users with high reporting demands). This gives the business sponsors the option to invest for the level of capacity while planning for future levels and to mitigate risks if the exact level of demand is not known.

Availability and Continuity Management

The goal of availability and continuity management is to ensure that IT systems are available for use when they are needed. Specific requirements should be detailed in SLAs. The requirements will usually be defined in terms of access to the system and performances level and not necessarily list the types of disruptions that can interrupt service. IT managers, responsible for availability and continuity management, will have to understand and address a variety of potentially disruptive problems.

Availability and SLAs

The objective of availability management is to meet service level requirements; this is done by monitoring and responding to key metrics. These metrics can address several aspects of system availability:

- Server availability—For example, does the server respond to a ping request?
- Acceptable performance—For example, do key database queries return results within a predefined length of time?
- Security—Is data transmitted confidentially, is the server protected by anti-malware programs, and so on?

When metrics indicate that SLAs are not met, a procedure should be in place to respond. For example, if application response times are slowing because of an increase in the number of users, secondary jobs can be shutdown or their priority lowered to free CPU capacity for the critical jobs.

There is also the possibility of a catastrophic disruption that leaves entire systems unavailable. For example, a natural disaster could destroy the primary data center housing servers and network equipment supporting a customer management system. This level of disruption is addressed by continuity management.

Continuity Management

The goal of continuity management is to ensure business operations are able to continue in case of a significant disruption in multiple services. IT continuity planning should be done as part of a broader exercise in business continuity management.

If there is a significant disruption in services, the business should determine which systems are mission critical and the order in which they should be brought back online. There are also financial considerations in continuity management. How much should be spent to ensure the customer management application is available in the event of a disaster at the primary data center? How long can the system be down before the business suffers adverse impacts? These questions are best answered using a formal risk analysis procedure that includes:

- Identifying assets
- Assessing the value of assets
- Identifying potential threats to assets and the likelihood of their occurrence
- Prioritizing the allocation of resources based on asset value and threat level

The purpose of availability and continuity management is to keep systems up and running at a level that meets SLAs. The tasks involved are varied because the problems addressed range from the relatively minor (the system is sluggish for short periods of time) to major (operations need to be relocated to a backup data center).

Summary

Service delivery entails many tasks, both operational and management. The previous chapter examined the operational aspects; this chapter covered the management side of service delivery. Although management issues are broad, they are dominated by service level management and financial management. If these areas are well managed, three other key areas—capacity management, availability management, and continuity management—are already well on their way to being effectively implemented. The next chapter continues to examine elements of systems management with a focus on application, software, and hardware management issues.