Chapter 1

The Case for Adaptive Infrastructure

Everywhere you look today, change is occurring at warp speed. And nowhere is it happening faster than in the business world. Those of us who have ridden out the business cycles of the last two decades understand that change is a given. But, with the rise of the Internet and e-Business, change isn't something that you decide to do anymore: It's something that can be forced on you daily.

You hear a lot of talk about business agility, but what does this really mean? It certainly isn't the ability of the CEO to leap tall buildings in a single bound. Agility means being prepared for change at a moment's notice. It also means you must have the infrastructure in place to support change without throwing away everything and starting over after each change because a completely new start takes too much time and is almost always too expensive.

The reason organizations need an adaptive infrastructure is very simple: More change happens in business than IT or the business can anticipate. To cope with the many unforeseen circumstances and competitive demands, businesses must create and possess a certain flexible, adaptive range.

Developing this flexibility requires an infrastructure planning process that makes it much easier to introduce new business initiatives and to grow initiatives that are already under way.

This chapter explains the fundamental approaches to adaptive infrastructure, so that you can start creating a more focused, organized way of dealing with changes in your organization. An adaptive infrastructure approach requires you to create deliverables in three areas—platforms, patterns, and services—and follow a five-step process for infrastructure planning. Concentrating on the delivery of solutions or "infrastructure products" and reusable components will provide you with the keys to success.

What is IT Infrastructure?

Before getting started, it helps to agree what the term "IT infrastructure" actually means. Generally, infrastructure is a relative term meaning "the structure beneath a structure." This definition implies different layers of structure, which metaphorically provide support or services, as shown in Figure 1.1.

In the physical world, the term infrastructure often refers to public utilities, such as water, electricity, gas, sewage, and telephone services. These utilities are just more layers of a total structure that includes IT infrastructure. Each layer of infrastructure has certain characteristics, including:

- Shared by a larger audience than the structures it supports
- More static and permanent than the structures it supports
- Considered a *service*, including the people and processes involved in support, rather than just a physical structure or device
- Often physically connected to the structure it supports
- Distinct from the structures it supports in terms of its lifecycle (plan, build, run, change, exit)
- Distinct from the structures it supports in terms of its ownership and the people who execute the lifecycle

The notion of a separate ownership and lifecycle (especially the design and run phases) is fundamental to the concept of infrastructure set forth in this book. This concept contradicts an assumption implicit in many object-oriented application development projects—that all reusable components should come from one set of designers, or that all designers should use a common framework such as Java 2 Enterprise Edition (J2EE). Discarding that assumption means you must design with a much looser coupling in mind.

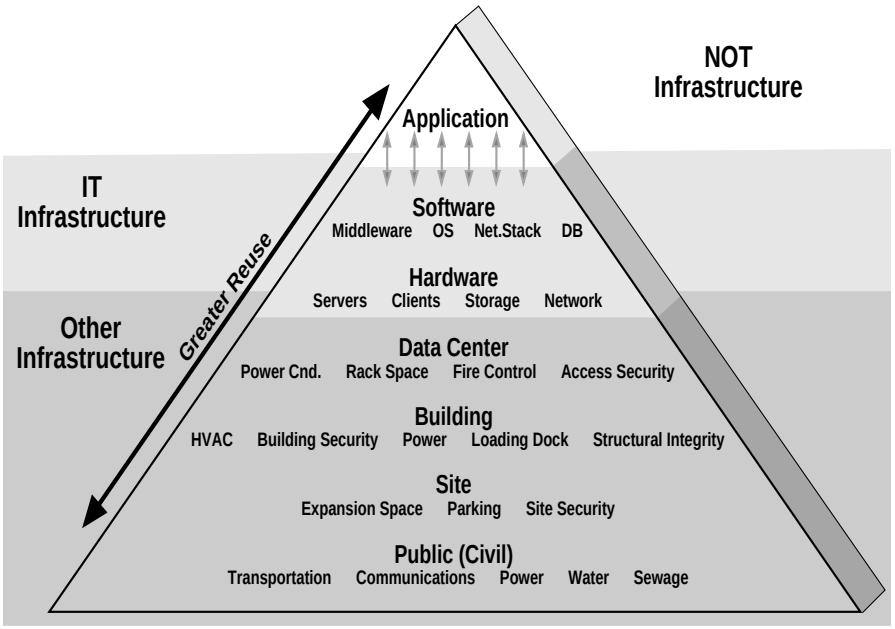


Figure 1.1 Types of Infrastructures

In contrast, those things understood as IT infrastructure (networks, servers, etc.) are already separate in ownership and lifecycle from application development. Expanding the notion of infrastructure is one key goal of an adaptive infrastructure strategy.

The Idea of Shared Infrastructure

When you think collectively of the people, process, and service aspects of a structure, you can more easily refer to it as a *system*. Not all infrastructure is physically connected to the structure it serves. For example, many people consider mail delivery, fire fighting, police patrolling, and garbage collection to be infrastructure services, yet none of these services must be performed while in physical connection with a building.

Infrastructure is easily described when it is clearly shared by many applications. Strictly speaking, however, many physical infrastructures aren't shared by multiple applications. The notion of infrastructure must also include the idea of an *unshared* infrastructure, which means every application has its own infrastructure.

Certainly, in some cases, it's best to consider a separate component still part of an application—even if it is not shared much. Consider an appliance, which is a bundle of hardware and software that is planned, built, and run as a single unit. Now consider a common off-the-shelf software package that is designed to be deployed and run on any common server type such as UNIX or Windows 2000/.NET Server. Even if the server isn't shared by other applications, it seems odd to think that it isn't infrastructure.

Surprisingly, even if the physical component isn't shared, other types of sharing may still be extremely valuable to infrastructure practitioners. The key distinction is that while the server isn't physically shared by multiple applications, it shares its lifecycle with other servers of the same model, in the sense that it is designed, built, changed, and exited by the server vendor. That server might be installed and managed using shared operational processes. Accordingly, the division between structure and infrastructure is relative, as well as porous.

Shared infrastructures certainly offer some benefits, which usually include lower costs due to greater efficiency, higher quality due to better management, and faster implementation times because it's already installed and available. However, a shared infrastructure is not always better, nor is it always practical in real business conditions. The value of shared versus unshared infrastructure is discussed in detail later in this book, but keep in mind that both kinds exist, and both are still infrastructure.

Creating Your Own Definitions

The previous section discusses one way to define infrastructure, but other definitions abound. For instance, some people define infrastructure as "anything that is shared across multiple business units," which includes large corporate applications, such as enterprise resource planning (ERP), customer relationship management (CRM), supply chain management (SCM), or even e-mail applications. Given that much new development treats these packages as infrastructure to be leveraged, this definition also shows that the job of maintaining applications has passed to a different group of people from the designers and developers of the application. Clearly, ownership of the solution is a key condition when defining layers of infrastructure.

Another common definition of infrastructure might be "anything that isn't fun anymore!" Many application developers and business units that enjoy the planning and execution phases of a project have no interest in

actually supporting in-place infrastructure once the operational phase begins. You see this happen also with marketing departments that hastily and gladly give up ownership of the Web site after the first significant infrastructure meltdown. In fact, this kind of thinking is a clear indicator of the need for some separation of ownership and lifecycle.

You can create your own definition of infrastructure, but remember the general rules and cases already discussed here. The next few chapters provide more specifics on the technology side of particular infrastructure components.

Why Infrastructure (Suddenly) Matters

As time goes by, more businesses are buying, commissioning, and even renting applications in one form or another. Meanwhile, fewer companies are building their applications internally from scratch. As a result, an adaptive infrastructure strategy becomes crucial to providing a versatile, flexible, and agile foundation for application deployment.

As business units increasingly make the application selections for their companies, infrastructure becomes a key focus for IT personnel. Therefore, an adaptive infrastructure should exhibit several key traits:

- Efficiency. The ability to provide reusable components that are reasonably priced and can be turned around quickly for application development projects.
- Effectiveness. The easy integration of all components in a way that supports their robust operation.
- Agility. Effective planning and design processes that allow companies to develop new applications quickly and to re-purpose or upgrade their existing infrastructure to support new requirements for existing or new applications.

Infrastructure matters because as your organization turns to third parties for more of its applications work, getting the infrastructure right is what's left for you to do. From a practical standpoint, if you are implementing a major new application package, what differentiates your effort isn't the application itself, but how successfully or quickly you get it running and how well it works. In many cases, these problems aren't application issues; they're infrastructure issues.

Most important, the Internet has made applications and infrastructure increasingly visible to customers and to the general public. Today, much of your organization's reputation and brand identity depends on

the quality of your IT infrastructure and operations, not just on your applications.

With the Internet as a common currency in the business world, any lack of integration, robustness, or agility on your part becomes immediately and embarrassingly obvious to key customers, no matter where they are in the world. When you made mistakes in the past, only your employees knew and it wasn't a big deal. Now, if you have problems, the whole world knows. CNN may broadcast a report on how your Web site hasn't been up for five hours! That's definitely poor advertising for the company's brand.

An infrastructure that frequently fails, or doesn't support the traffic load, or can't provide a single integrated view of your complex organization can cost your company immediately in lost sales and lost goodwill. When you realize what's at stake, you begin to see why savvy organizations are investing more to make sure that their infrastructure doesn't lag far behind their business vision and applications. Unfortunately, once your lack of robustness and agility is exposed, you cannot change things very quickly. Having an adaptive infrastructure will ensure that you don't get caught flat-footed when your time comes to shine on the world's stage.

The Clash of Cultures

So you'd like a better infrastructure but you're a little short on cash? IT people realize all too well how difficult it can be to sell infrastructure improvements to the organization. In companies with less enlightened management, any IT department trying to reach the goal of an adaptive infrastructure will encounter two common yet fundamental reactions.

- Stability is good. People often feel it's good to have an infrastructure that is stable, unchanging, and predictable. Certainly, predictable, systematic behavior must be achieved at some level. However, infrastructure must also be flexible, even breakable, to be fully leveraged by business.
- Infrastructure costs are bad. Second, businesses usually regard IT infrastructure as a cost to be minimized or a necessary evil. However, as business increasingly becomes "informational" in nature, the systems for information capture, management, and delivery become even more central to business success.

A clear misalignment between business and IT organizations dominates infrastructure decision-making. Business is chronically disconnected from what is happening on the infrastructure side. Here's the way it typically works:

1. High-level executives work with high-level consultants to determine strategic direction. This direction might take the form of some broad initiative, such as becoming more customer-focused, creating unique new offerings, integrating with suppliers, or becoming a low-cost provider.
2. Usually, this initiative drives some major application development. However, information about these decisions often reaches infrastructure planners in the form of rumors or hearsay and, usually, after the decision has already been made. Infrastructure planners should be involved in the decision-making process. All too often, they aren't.
3. Meanwhile, the systems and applications are developed and customized behind a wall of consultants and systems integrators. The resulting application code, servers, and other components often show up in the operations center when the application is about to be promoted into production, expecting that IT can support it without any extra costs.
4. Not surprisingly, this disjointed solution often results in applications that do not perform nearly as well as intended. Such applications can even degrade the performance of other applications, because developers don't understand the complexities of a shared network infrastructure within the organization.
5. This approach further increases the complexity of managing the whole infrastructure. Fresh applications often arrive with new or different components, which require an already over-stretched IT organization stretch itself further to support these variations.
6. Unfortunately, the need to support everything often leads to mediocre support at best, or expensive and ineffective outsourcing at worst. For excellence in business as well as infrastructure, you must be able to focus on what's critical to the business. Of course, outsourcing isn't all bad; it's a legitimate business requirement and you must assign skills and effort to this indispensable approach. However, outsourcing infrastructure must be well thought out and not executed only as a reactive or simple coping strategy. Other-

wise, the infrastructure becomes expensive, slows things down, and usually ends up generating quality problems.

7. Finally, this situation places IT in the unenviable situation of developing an ongoing investment strategy with little or no idea of what new applications will look like—or even what type of business they must support. Not surprisingly, this way of doing business can get quite expensive. While technology vendors deliver constant improvements in price and performance, major infrastructure investments are still multimillion-dollar decisions.

The gap between business and IT, as well as between application development and infrastructure within IT, often causes havoc. Some of the issues can be addressed through better communication. Fundamentally, however, infrastructure planners within IT must be prepared to take their own initiatives. To solve the problem, infrastructure planners must introduce standard practices and procedures into the planning, design, and implementation phases. These standards should create and accommodate adaptability.

What's the Problem with Most Infrastructure?

Simply stated, the problems outlined above manifest themselves in many different ways. Do any of these comments sound familiar to you?

- Costs too much. You can't get funding, can't figure out costs, or can't articulate value.
- Too slow. You can't get applications out the door fast enough to be effective. The time to market must be faster.
- Nothing works together. Your systems are incomplete, overly complex, or unpredictable. Data and processes from one application may not be available to another.
- Handoffs don't work. Ownership isn't clearly defined. Handoffs don't work between IT groups such as application developers, infrastructure planners, and operations. People are always dropping the ball. Key assignments aren't being made to get work done.
- Too much theory, not enough practice. People don't know enough about technology, or they are waiting for technology that will take too long to arrive, so they are paralyzed by uncertainty.
- Lack of focus. Planners spend all their time responding to a pager and never have time for key planning activities.

What else causes problems in your infrastructure today? Is it technology, processes, or people? What problems would your IT infrastructure customers identify? You should have a good idea of the exact nature of the problems in your organization before you start suggesting solutions. Then, as you move toward solving problems, you can tie your solutions back to the problems they are meant to solve.

What's the Solution?

Even though your problems may seem unsolvable, in fact you have some very clear and workable solutions to your dilemma. You can stop the vicious cycle of problems described above in various ways:

- Plan your infrastructure end-to-end. When you plan infrastructure, you can't just plan a piece at a time and hope it comes together. To be adaptive, your infrastructure planning efforts must become more extensive. They must consider all layers of the IT model and fit new components into a complete infrastructure solution that can service an entire application. This book provides a number of successful techniques, such as categorizing infrastructure into *patterns* that can help you do a more thorough job of planning.
- Design an adaptive infrastructure. Your infrastructure shouldn't just meet today's requirements; it should be ready to scale, adapt, change, or grow to deal with challenges already looming on the horizon. Once you identify these challenges, you must face them squarely and start designing for them immediately. This book explains the fundamental concepts of adaptive infrastructure and explains how to identify your major challenges. The answer to being more adaptive usually involves focusing more on people and processes, instead of focusing only on the infrastructure components.
- Execute a reuse-centric strategy. A key reason for building an adaptive infrastructure is that many design standards and actual physical components of the infrastructure can be reused. Reinventing the wheel for every application only makes your infrastructure increasingly unmanageable and slows down its delivery. This book explains how to identify key infrastructure patterns within your organization and how to structure them to leverage a set of reusable adaptive infrastructure services.
- Overcome the tech-only focus. Many IT people seem to focus on making product choices or architecture choices, while ignoring the people and processes needed for successful operation. You

can make great technology choices, but if you don't have the right people and processes, your choices will be useless and you won't get the success you need from them.

- Choose the right technology and products. Of course, striking a balance doesn't reduce the need to always select the best technologies and products for your infrastructure and application delivery needs. The latest, best-of-breed solution might not always be the right one for your organization. Quite often, it turns out that the best engines are not made of the best individual parts as much as they are made from parts that work best together. The same holds true for IT infrastructure.
- Balance immediate needs with long-term goals. Few people have the proverbial luxury of stopping the train to redesign the railroad tracks. There simply isn't time to do that in today's fast-paced world, and the costs would be horrendous. To be successful, you must be able to change what you are doing *while you are still doing it*. This theme is repeated throughout this book. You must strike a balance that helps you transform while you are performing. This book shows you how to create this balance, and it gives you a few specific approaches that can work for you.

The Philosophy of Adaptive Infrastructure

Obviously, developing an adaptive infrastructure isn't something that happens overnight. To create major change within your organization, you must start by changing yourself—by adopting a new way of thinking and a philosophy that will guide you toward your goals.

Striking a Balance

Adaptive infrastructure strategy must include the people, processes, and technologies that provide ongoing support for an organization's applications.

To achieve success, you must strike a balance that doesn't neglect or place undue emphasis on any of these three areas:

- Technology. This category includes the hardware, software, and leveraged third-party services that constitute the infrastructure. Too much focus on technology is known as the "silver bullet syndrome," the idea that a particular technology will solve all your problems. In the real world, silver bullets rarely exist.

- People. In this category, you can include the roles, skills, and organizational structures involved in the infrastructure lifecycle processes. Over-reliance on people creates a "hero mentality," which assumes that by working harder and being smarter, you can do everything by working around the clock to get it done. This is one of the most pervasive problems in IT departments today. Now that the euphoria has subsided over "get rich quick" stock options in the Internet Age, employees won't work around the clock, and even if they do, it is at a serious cost to their health and well being. Make sure your behavior doesn't reinforce hero mentality. While heroes should be rewarded, the key is emphasizing how things will be done differently in the future, so that people don't have to be heroes and don't have to drive themselves to the point of burning out.
- Process. This category includes standards and information that define the lifecycle of infrastructure (design, procure/build, customize, configure, deploy/install, manage, change, exit). Actually, too much focus on process is quite rare. Some companies are a little too process-centered, but many more IT organizations lack a systematic approach.

To strike a balance, you must know enough about technologies to pick the right technology. You must understand the people skills required to get the job done, without expecting superheroes to come in and save the day. Process is an increasingly important focus because it's not enough to have a visceral understanding of adaptive infrastructure strategies; you must focus on process as well. As explained in Chapter 6, you need a set of effective processes and methods that will help you bring about change. Don't try to accomplish important things by chance, and don't make people figure it out as they go. Doing so rarely leads to consistent, replicable success. After all, the goal of a great infrastructure program is not that things go well once, but that everything goes well consistently, from project to project.

Key Organizing Principles

One way to look at adaptive infrastructure is to see it as a set of components, patterns, and services, along with the people and processes nec-

essary to tie them together. These key organizing principles drive much of the content in this book:

- Platform is an organizing concept that groups individual component technologies into technical domains (or layers).
- Patterns are organizing concepts that facilitate rapid mapping from business requirements to end-to-end infrastructure designs. Patterns structure component selections from many platform layers.
- Services are "infrastructure applications" that shift responsibility for certain services out of the application domain into the infrastructure domain. Services provide a set of physically shared components, such as a network or a credit card processing service, which multiple applications can leverage.

Figure 1.2 shows that all the elements of adaptive infrastructure work together to support applications in an organized way.

If applications are the physical manifestation of real business processes, then all the elements of infrastructure must work together successfully to ensure their flawless performance.

The art of adaptive infrastructure, however, is not to cater to every application on its own terms. Doing so only creates more "stovepipes" (or applications resistant to adaptive infrastructure) within your organi-

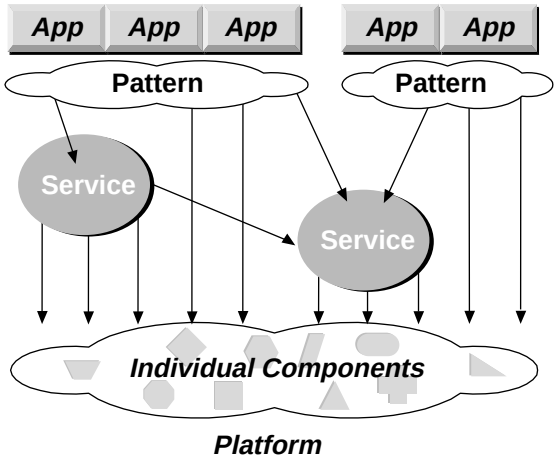


Figure 1.2 Key Organizing Principles for Adaptive Infrastructure

zation. Instead, you can make both the application and infrastructure development processes more manageable by defining infrastructure "patterns" that you can manage more effectively. These patterns are built on a foundation of key services that you have clearly identified as crucial to your business operation. These services, in turn, are based on individual components working together as part of your adaptive infrastructure platform.

On one level, the first step in building an adaptive infrastructure is to identify and catalog all these elements—the patterns, platforms, and services, along with the people, processes, and packaging that will make your IT organization successful. Once you organize your world this way, you can avoid the dilemma of having to start from scratch each time a new application rolls out, asking questions such as, "What exactly do I need for service levels?" or "Which component do I select?" Instead, you will have pre-built solutions that you can apply or adapt at a moment's notice, which will provide your organization with ultimate agility.

Everything you need to create an adaptive infrastructure strategy boils down to the six fundamental concepts discussed on the following pages. These concepts set the tone for your infrastructure planning efforts. They form the core strategies of this book and map directly to the remaining chapters.

1. Identify and Catalogue Technologies

If your decision-making attitude is, "I bought from Vendor X, so now everything is solved," you're probably thinking the wrong way about the problem. The notion of platforms emphasizes organizing hardware, software, and networking components (technologies) into common application runtime targets that maximize component reuse and systems integration while providing a base level of shared services.

To manage your infrastructure well, you must first identify and catalog all the components by their functional categories.

By organizing components into categories, you can assess the complexity of managing hundreds of components. As you begin the organization process, you will start to see the components used to deploy applications fitting into different layers of stacked infrastructure, as shown in Figure 1.3.

The component items in the layers below the line (e.g., a particular server) are all purchased, not physically built. In contrast, those above the line may be internally developed, particularly if they represent areas of potential competitive advantage. And where the application and the infra-

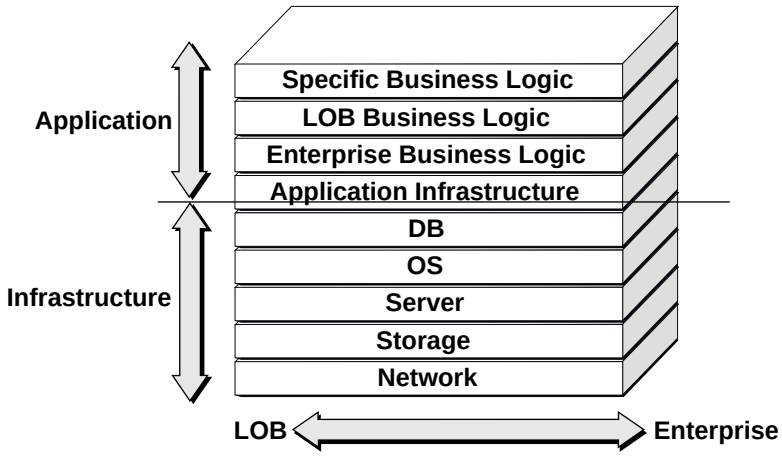


Figure 1.3 The Infrastructure Stack

structure worlds intersect, systems integration becomes a crucial element. This integration typically focuses on various uses of middleware to combine applications, processes, data types, and underlying infrastructure as necessary.

Chapter 2 provides more detail about these layers and the organizing principles behind categorizing and managing component technologies.

2. Develop Reusable Infrastructure Patterns

One important way to resolve some problems is to simplify wherever possible. The best way to simplify is to identify patterns within your infrastructure that can be supported, augmented, nurtured, and reused across applications to ensure success during an endless succession of business initiatives and application rollouts. Using these end-to-end sets of infrastructure components (from many platform layers), you can clarify and unify technology, planning, and operational processes, as well as personnel experiences.

It's a losing proposition when you react to the wide variety of application development requests by trying to maintain expertise in every type of infrastructure. To make things more manageable, select a few key patterns to build your expertise around, and then use these patterns to support business projects in a repeatable way. By starting with a core set of patterns and tweaking them as necessary, this makes things easier and less expensive for everyone. In other words, simplify and prioritize.

The old barriers that prevented you from simplifying are starting to fall away. Standardization of network protocols, for instance, not only simplifies your job, but also helps you deliver a better, more focused, and ultimately more credible result to your business managers. Thinking in terms of patterns can help you handle ongoing plan-build-run work quickly and efficiently, even in places where you cannot set standards easily—and particularly at the product level, where technology innovation is fastest and maturity is a distant goal.

If you create standardized infrastructure patterns that are robust, flexible, and reusable, this will help you streamline the process of providing infrastructure for application developers and business units. The infrastructure pattern matching (IPM) methodology, outlined in Chapter 6, will help you work with your internal customers to set infrastructure investment priorities that reflect real business requirements. Figure 1.4 shows how a pattern-based method unites as it simplifies.

Using pattern-based methods, you can avoid the old problem of creating internal "stovepipes" that are isolated from the rest of the infrastructure and don't integrate well with the rest of the business. Instead, you will have an infrastructure that integrates much more easily, serves multiple applications more efficiently, and actually outlives the applications it supports. Pattern-based infrastructures reduce the incredible variety of

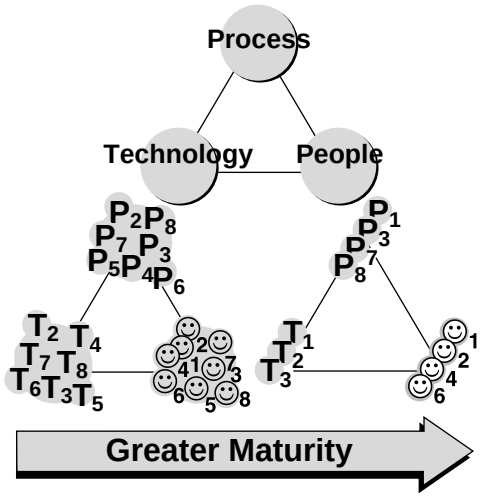


Figure 1.4 Goals of a Pattern-based Infrastructure

technologies, processes, and people (skills, roles, etc.) that are required for accurate application delivery, which will result in more focused and repeatable excellence.

Chapter 3 explains how to define, categorize, and catalog patterns, and it provides a starter kit of nine patterns that you can use to begin building an adaptive infrastructure.

3. Develop Adaptive Infrastructure Services

The next step in organizing your infrastructure thinking is to structure components from platform layers into various classes of services to be used by patterns. A service exists when someone delegates the responsibility for performing a process to a service provider. A service provider can be any person or system that can perform a task repetitively. In the outside world, service providers include people such as bank tellers and plumbers. In the IT world, a service provider might be a storage area network, a database, or an IT help-desk person. Unlike a component, which is focused solely on technology, an adaptive infrastructure service is a shared set of technologies that is implemented once, with a common set of processes and people skills, to be reused by multiple applications. While these services are not the entire end-to-end set of infrastructure services for an application, they can be reused by such infrastructure patterns.

To be truly efficient and reusable, services must be decoupled and become separate processes from the person or system that interacts with them. By defining services in this way, you can start removing the stovepipes from your IT infrastructure.

The network itself is an ideal example of the concept of adaptive services. The network has been removed from the application for so long that people tend to forget the days when each application required its own special type of network. Today, no one thinks of the network as part of an application; it's a service on which the application runs. No one builds a unique network just to host a single application. Not too long ago, however, such an arrangement was painfully common.

In recent years, IT infrastructure has evolved to the point where most people use a single network service, namely TCP/IP, to support all applications. While networks are still dedicated for specialized applications, such as the wireless networks used in the power industry to dispatch repair trucks, this practice is relatively rare.

Integration is another important area where services are being decoupled from applications. Enterprise application integration (EAI) uses message brokers to move data transfer functions out of the applications

themselves and into a reusable service that can be shared by all applications. The trend, obviously, is to turn reusable shared components over to the infrastructure where they can be more easily maintained and shared across applications as services. This is the kind of building block approach that an adaptive infrastructure strategy can help you achieve.

Chapter 4 explains the organizing principles behind adaptive services. Chapter 5 provides a starter kit of services you can use for your own infrastructure, with special emphasis on Transactional Integration and Identity Infrastructure services.

4. Use Good Tools

Once you have identified patterns, platforms, services, organizational issues, and old problems that must be fixed, you should sit down with a robust set of tools and processes and begin the journey toward organization and clarity. Chapter 6 introduces the following processes and methods that you can use to simplify your job.

Infrastructure Pattern Matching (IPM). If you're an infrastructure planner, what the business really wants from you—in addition to credibility and leadership in the IT arena—is the ability to quickly estimate the cost, schedules, and risks associated with new projects. Infrastructure Pattern Matching (IPM) helps by providing systematic answers to three fundamental questions: Who are the users, where are they, and what work is being performed? Answering these questions helps you define service-level commitments, analyze costs, and identify the core technology issues that affect the infrastructure's ability to support business initiatives.

Periodic and Annual Processes. Having structured, repeatable processes with concrete outputs or deliverables will make a difference in terms of the speed, quality, and cost of everything you do. Chapter 6 shows you how to organize two planning processes that you will execute repeatedly. One of them is periodic (or *strategic*) infrastructure planning to review your standard infrastructure patterns and services on a regular annual cycle. The other is per-project (or *tactical*) infrastructure planning, which is done for each application or new technology being introduced into the organization. Figure 1.5 shows the relationship between tactical and strategic processes.

With a robust set of tools and a well-defined set of processes, your team is able to respond to application support requests in a repeatable, structured way in a matter of hours, rather than in weeks or months. In

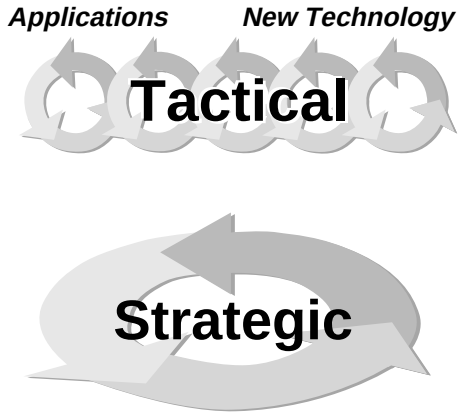


Figure 1.5 Tactical Versus Strategic Processes

the process, you will generate enormous credibility for the adaptive infrastructure concept and for your whole IT organization.

Portfolios. Anyone who uses Quicken or Microsoft Money knows that half the battle in financial management involves keeping your planning portfolios up-to-date. Waiting until tax time to update your portfolios can be extremely painful. The key is to apply discipline and a set of easy-to-use tools to continuously update your portfolios. The portfolio concept is also an important tool of infrastructure planning, as shown in Figure 1.6. Infrastructure portfolios keep you organized as you identify, catalog, and manage your patterns, platforms, and services on an ongoing basis.

Once you develop a set of infrastructure portfolios, people will know where to find the details when they need them. A portfolio can be something as basic as a physical filing cabinet with folders in it or a directory structure with word processing documents, spreadsheets, and diagrams. The best way to make a portfolio sharable is to put it on an intranet site using a database or content management system to deliver the information to the widest audience.

The contents of a portfolio are fairly predictable. For instance, in the Pattern portfolio (see Figure 1.6) you might have a Web Publish folder with use cases, architectural diagrams, and service level metrics showing the service levels that the Web Publish pattern will meet. A reference manifest might show the product names and version numbers to be used for the database component of this reference architecture.

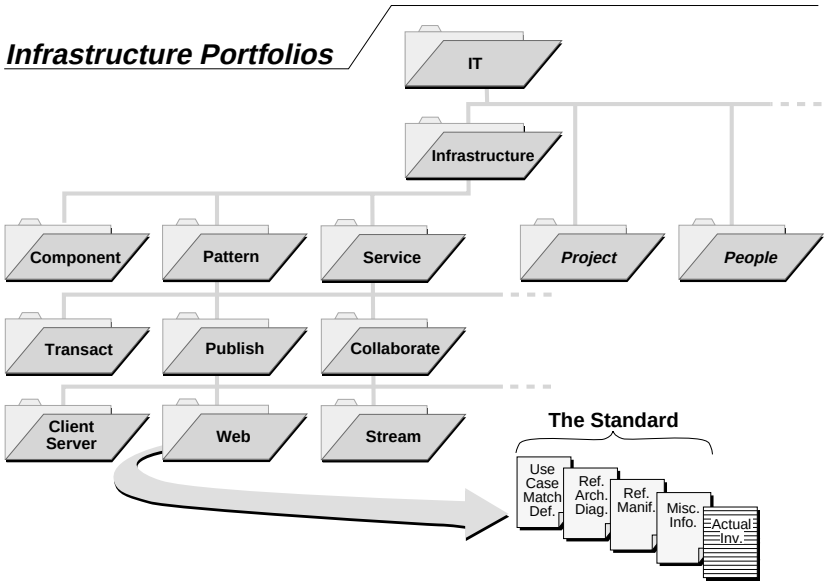


Figure 1.6 The Concept of Portfolios

Techniques for building and using portfolios are discussed later in this book. Chapter 6 provides more detail on proven infrastructure planning methods such as infrastructure pattern matching, predictive cost management, and impact assessment labs.

5. Get Organized

For infrastructure planning to work, it has to be more than just a "good idea." It has to become an essential part of your business.

The only realistic way to incorporate infrastructure planning into your business is to create new roles and responsibilities, job titles, and even new groups or departments where necessary. Someone must own the processes of infrastructure planning and development to ensure that it is completed. Staffing with a full-time equivalent person (FTE) will ensure that at least one person is no longer being distracted by day-to-day operational responsibilities. What's more, infrastructure planning roles must be clearly separated from traditional application development roles. Separating the roles allows each group to focus on its particular strengths, particularly if they offer a shared service with separate lifecycles.

Infrastructure developers can be responsible for designing, implementing, and managing the interfaces between enterprisewide resources and the infrastructure shared by multiple applications.

Application developers provide project-related interface requirements to infrastructure developers who ensure that interfaces are implemented efficiently, securely, and with management controls.

At the group level, having a team of infrastructure planners and developers can create priorities across an array of infrastructure projects. The team can make sure that infrastructure standards, including components, patterns, and services, are available. Perhaps more important, the team can guarantee that the infrastructure standards are reused for particular application development projects. Such a group can also identify potential areas for reuse, not only of technologies, but also of project management methodologies, documentation, and some of the processes and people involved.

Within this more refined focus, infrastructure planning professionals can recognize when unique components are required and determine what they will cost. Planners can maintain a longer-term view of infrastructure requirements, and they understand the cost/return dynamics of infrastructure building. They can also shift the focus from an emphasis on particular technologies to continuous improvement of the delivery process.

By focusing on reuse, product or solution delivery, and the other concepts presented here, infrastructure planning becomes a more organized and tangible process. Presenting a range of potential infrastructure choices to internal customers, including application developers and business users, places the ultimate decision with those who are responsible for creating value within the enterprise. Business users can actually start comparing the benefits of one project to the other, ultimately making the determination from a business standpoint. Infrastructure planners indicate the cost options, but business users must determine the benefits.

Obviously, creating this type of organizational structure requires a clear definition of the relationship between infrastructure planners and application developers, as described in Chapter 6. In addition, planners must establish credibility and maintain constructive relationships with the people who manage project teams within the different business units. Chapter 7 gives further detail on how to organize people and processes into distinct roles and responsibilities, as well as how to encourage compliance with adaptive infrastructure standards and policies.

Chapter 7 also provides more detail on the organizational issues that you will encounter and explores alternatives for staffing and organizing your infrastructure team.

6. Describe Value Through Packaging

You can learn all the engineering principles, design methodologies, and pattern approaches that you want. You can create your own infrastructure development group and achieve perfection in all your processes. But if you can't sell your approach to the business and continually show the value of what you're doing, then all is for naught.

Selling is the main idea here. Business unit managers who hold the purse strings must understand the *value* of what you are proposing. Only when they see value will they be willing to loosen the purse strings and give you the investment dollars and management support that you need.

One of the most important techniques you can use to sell the value of adaptive infrastructure to upper management is the concept of an "infrastructure product." This product is an ongoing, reproducible, and repeatable set of services that your IT organization can deliver to the business.

For example, in a retail store environment, line executives will sign on much quicker for a world-class system that sustains a particular retail function, such as point-of-sale transaction processes, than they would for a world-class systems administration function. Retail executives will always be more interested in the in-store process, which they see as more valuable. When dealing with these types of concerns, your emphasis should be on packaging and pricing infrastructure products that support those efforts. Don't just solve your own infrastructure problems; solve your customer's problems too. At the very least, make a connection that shows how the work you must do to address your own issues will also end up solving their problems.

In addition, business leaders often have specific applications that they will pay extra to see delivered well, such as e-Business systems, enterprise resource planning applications, customer relationship management solutions, collaborative engineering capabilities, or accounting applications. Once you master the quality-of-service issues, you can create premium subscription services for applications, while ensuring that these services are handled in a premium fashion. Such applications can then justify additional expenses such as online backup and around-the-clock support, because of the higher value perceived by the business.

Once your organization accepts the concept of infrastructure as a set of packaged products and services, infrastructure planning becomes an ongoing process of refinement. It focuses more on increasing the service levels offered to the business. As you add more infrastructure and applications, the entire conglomeration starts behaving in an almost organic fashion. The objective is to optimize ongoing investments, while maintaining a balance between what the infrastructure delivers and what applications require.

As with any other business, deliverable products should be packaged in a coherent manner and supplied in a consistent, repeatable fashion. They should be organized to match what the business values. If infrastructure costs go up, you must tie these increases to real business needs and value. With a product-delivery mindset, the idea of *reusability* becomes more important in your approach.

Chapter 7 explains how to develop packaged infrastructure products and communicate their benefits to managers in a way that will earn levels of funding for future infrastructure development projects commensurate with their business value.

Benefits of Adaptive Infrastructure

So far, most of the benefits of adaptive infrastructure have been fairly obvious. Other benefits may not be so apparent. Some of the less understood benefits are detailed below.

Developing an Adaptive Range

Understanding exactly what is "adaptive" about adaptive infrastructure can be a complex task. A number of dimensions are involved in quantifying the flexibility or adaptive range of infrastructure, as shown in Figure 1.7.

One obvious measurement is cost. Everyone wants a fixed, low cost, but how you deliver it is the real problem. Speed is the next measurement that attracts the most attention. Speed refers to the timeliness with which IT can respond to users and implement the business processes they want. However, speed is a derivative measurement. Certainly, buying more powerful hardware can increase the transaction rate, but infrastructure isn't equipped with a speed knob that can be turned up or down as needs dictate. To increase speed, other dimensions must be addressed.

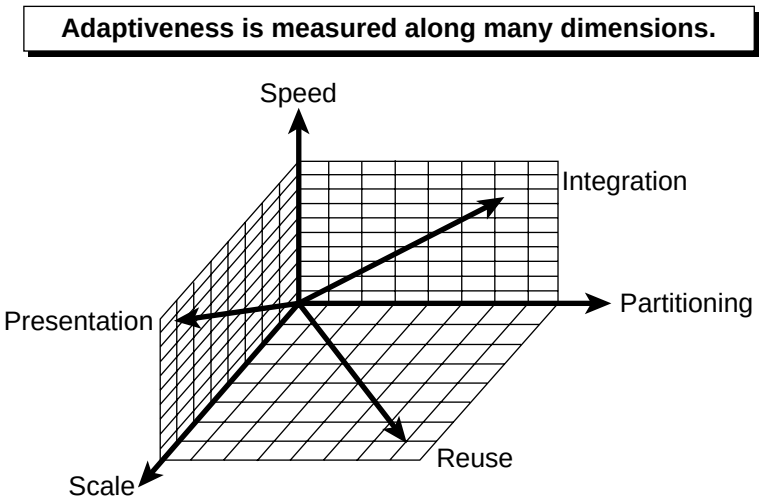


Figure 1.7 Measuring Adaptive Range

Scalability. You can build in some headroom (adaptive range), so that boxes and boards don't have to be swapped when increasing the number of users for an application. That is a relatively easy (though expensive) way to be adaptive, but it isn't sufficient to accommodate the rapid changes in business processes that users are creating.

Presentation. Another dimension that affects virtually every IT organization is presentation: the way information is presented to users or business partners. Historically, the presentation layer has shown little adaptiveness. Organizations now spend a great amount of time and effort converting more traditional 2-tier, client/server applications into Web-based applications. Unfortunately, designing Web-only presentation solutions is equally limiting. Other presentation methods, such as personal digital assistants (PDAs) and interactive voice response (IVR), can't support a full-page Web display and will require completely new development efforts, which are expensive and time-consuming.

Presentation independence alone doesn't guarantee sufficient adaptability, however. Infrastructure planners should be very focused on the front end, but if the application-to-application integration issues aren't addressed simultaneously, the result is taller "stovepipes."

Partitioned complexity. The ability to partition functionality and complexity within the infrastructure is another measure of adaptability. If

the infrastructure cannot be effectively partitioned, the resulting complexity will spread throughout the organization and eventually become unmanageable. You must effectively manage the interfaces between applications and between infrastructure components, both for the enterprise and for applications used by external partners.

Integration/reuse. Infrastructure integration and reuse are also measures of adaptability. The typical organization requires a dramatic increase in the reuse of infrastructure code, other technology components, and skills to increase adaptability and speed of deployment. Reusable code is the opposite of legacy code. Whereas legacy code can be difficult to maintain, enhance, and integrate, the most adaptive code can be adjusted as the business evolves.

These factors and more contribute to the adaptability of an organization's infrastructure. This book discusses many of these issues in greater depth.

Increasingly, you must describe the value proposition for infrastructure directly to the business users in terms of the discrete service levels that they want delivered. By turning to a discussion of service levels, you can influence business users to consider more than just the immediate impact of a single application. You must convince the business to consider the value of adaptability, because it will take more money, effort, and time to deliver than single implementations.

The Importance of Reusability

Of all the concepts discussed so far, reusability is the most crucial. The patterns, platforms, and models used to create an adaptive infrastructure all depend on reusability. Without an effective approach to this issue, it will be difficult to apply the additional rigors of specifying pattern components and creating predictive cost models.

Reuse isn't simply a matter of being thrifty or making do with whatever hardware happens to be lying around the shop. In fact, reuse often involves just the opposite—weaning business units away from archaic legacy systems that are no longer cost-effective and replacing flawed applications or integrating them with other applications within the enterprise.

When implementing a reuse policy, the biggest challenge is getting people used to the new policy and getting them to fund it appropriately. It's difficult to tell business executives that their infrastructure requirements are physically impossible, or that they can't have an application because it's too difficult to support.

Many business units have the financial and human resources to subsidize non-standard implementations, but they don't. Instead, the IT department ends up absorbing the extra expense to support the deviation, or devoting extra resources and effort to making sure it works properly. This will continue until you present errant business units with a reasonable alternative or a bill for the extra time and effort. If the business units only encounter half-hearted resistance from IT, you will probably spend most of your time doing ad-hoc support, and you can forget about creating an adaptive infrastructure.

Reuse policies are both a logical precursor to and an integral part of implementing adaptive infrastructure techniques. Designing reusability well requires that you leverage all the concepts introduced here, particularly patterns and services. Once your organization is ready for reusability, or actively practicing it, you are ready to implement an adaptive infrastructure.

Reusable Components

The following table is a list of the infrastructure components that should be included in any reuse efforts.

Reuse and Adaptive Infrastructure

The concept of reuse has been around since the arrival of object-oriented programming (OOP) languages in the 1980s. OOP promised that developers could write applications, then reuse large blocks of code, called "objects," in other new applications.

OOP failed for a number of reasons. The concept was too complex and fine-grained for the average programmer to grasp. In many cases, software developers created objects that were so specific to a particular application that it was difficult to reuse them without a major rewrite.

As a result, developers were forced to start from scratch each time. Still, the OOP concept was compelling enough that others have since attempted to introduce reuse at higher levels of abstraction, such as security, middleware, and networking.

Scalability was another problem. An organization with a 100-percent object-oriented approach had to support millions of objects. This proliferation resulted in a separate object for every customer in a particular database. The history of some of object-oriented database wars shows this simply doesn't work, as shown in Figure 1.8.

Table 1.1 Reusable Components

Basic Components	**Metacomponents (Define standards for architecture, interoperability, etc.)**
Network - Circuits - Routers, hubs, etc. - Management software	Business rules
Hardware - Servers - Storage - Workstations	Process models
Software - Applications - Subsystems - User interface designs - Utilities (backup/recovery, security, audit, error handling, etc.) - Code fragments, macros, object classes, etc.	Documentation templates - Technical architectures - User manual outlines - Operations methods
	Action diagrams
	Data models - Types - Definitions - Structures
	Algorithms
	Plans - Project - Deployment - Testing

Note: A more extensive infrastructure component list can be found in the component catalog listing detailed in Chapter 2.

In fact, reuse is best achieved at the infrastructure level, not *within* applications but *between* applications. For this reason, you should focus reuse efforts on broader external components.

A banking and financial services firm, for example, might have multiple credit authorization processes to support various business units. To promote reuse, you could consolidate these various applications into a single credit authorization module. It doesn't matter whether this common module is object-oriented or not. Reusability stems from the fact that different applications share it.

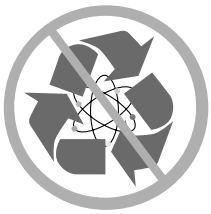
Third-party software vendors are leading the way in this area. SAP, for example, includes integration components in its architecture in the form of Business APIs (BAPIs). PeopleSoft leverages middleware from BEA Systems to integrate various modules, processes, and data types within its application suite. These types of components and middleware are crucial for organizations that are integrating existing applications into e-Commerce and supply chain solutions.

Applications that present IT services to external business partners depend heavily on component-oriented solutions. Each part of the supply chain becomes a component with interfaces exposed to its neighboring component, permitting the development of some fairly complex supply chain configurations.

The key to success in infrastructure planning is to change the way things are done so that applications become easier, cheaper, and quicker to integrate. Success also means being able to run these services long-term with high quality.

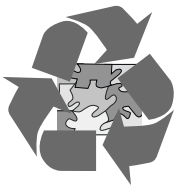
Another form of reuse that emerged in the 1990s involves placing a "wrapper" around legacy applications using interface definition language (IDL) models such as CORBA or DCOM. This practice is a perfect example of internal applications not needing to be object oriented. Typical mainframe applications can be migrated into a component framework by wrapping them inside an IDL.

Today, approaches such as Java 2 Enterprise Edition (J2EE) and Microsoft .NET are moving further down that path. These approaches provide not only "wrapper" capabilities, but also the beginnings of a set of Web services that can be leveraged across applications, processes, data types, and businesses. Web services represent the latest opportunities for leveraging and reuse at the intersection of the application and infrastructure worlds. These approaches take advantage of standards such as Extensible Markup Language (XML); Universal Description, Discovery, and Integration (UDDI); Simple Object Access Protocol (SOAP); and Web Services Description Language (WSDL).



▲ **Internal Components (Fine-Grained), NOT!**

- Object-oriented programming constructs
- Programming patterns
- 1 million customers = 1 million customer objects



▲ **External Components (Coarse-Grained)**

- Merging 12-credit authorization process into one
- Accessing SAP business process via BAPIs
- Integrating existing applications into e-Commerce and supply chain solutions
- Wrappering legacy applications with IDL

Figure 1.8 What Kind of Reuse Is Achievable?

In summary, successful reuse efforts don't get deeply involved with fine-grained componentization. Rather, these broad efforts emphasize integration components and focus on the infrastructure level.

Is Your Organization Ready?

Now that you understand the concept of adaptive infrastructure, it's time to determine whether you are ready to use the techniques and concepts described in this book.

The following questions can be revealing:

- Has your organization ever not purchased an IT product because it did not comply with some specified level of interoperability or failed to fit current or planned infrastructure practice?
- Does your organization have a technical architecture group? What is it responsible for?
- Has your organization ever funded a reusability study? Who paid for it?
- Exactly what are your infrastructure successes and failures? What would your customers, including application developers, business managers, and partners, say are your strengths and weaknesses?

The answers to these questions will provide a fairly clear picture of whether your organization is pursuing an infrastructure reuse strategy, or if it is ready to do so. If your organization actually has managed to veto a product purchase, you might want to analyze exactly how that was accomplished.

Summary

To summarize what's been discussed in this introductory chapter:

Success depends on the appropriate IT focus. Your focus should be on solution or IT product delivery, not technological prowess. Any IT organization will find it difficult to differentiate itself positively based on how effectively it handles its systems administration. Delivering basic technology services is like having the lights come on when you flip a switch. The real value proposition for IT is not making the lights work, because this is expected of you anyway. Instead, the real value proposition is delivering sharable support services for applications in a way that promotes reuse, cost savings, and agility.

An "infrastructure product" mentality helps simplify options and drive competencies. Thinking in terms of infrastructure products leads to a delivery mentality. Creating tangible infrastructure products reduces the amount of uncertainty for your IT department and, ultimately, your organization. You should focus on core infrastructure patterns and services, looking for ways to reuse pattern and service assets and expertise while emphasizing consistent delivery with people and process improvements.

Developing adaptive infrastructure requires a change in culture and relationships. Successful implementation of adaptive infrastructure will change the relationship between business users and the IT group, but it should also change a key piece of the IT culture: the application developer community. Your efforts will create a new class of workers—infrastructure planners and developers, who will have application developers as their customers.

The ultimate goal is for IT and business to develop strategic investment priorities together. The typical IT department can develop a suitable infrastructure plan at a given point in time. Developing a plan that reflects and accommodates developments within the business,

then updating and managing that plan, will require close coordination with business managers.

Reuse is the linchpin of adaptive infrastructure. Substantive reuse is impossible at the application level. Instead, reuse must be fostered at the interface level, among application components. A number of adaptive infrastructure concepts are designed to facilitate reuse, including adaptive infrastructure services, infrastructure pattern matching, predictive infrastructure cost models, and the role of the infrastructure planner.