Chapter 1

Understanding the SharePoint Hype

In This Chapter

- Understanding what the big deal is about SharePoint
- Exploring the benefits of SharePoint development
- Discovering the pieces that go into the SharePoint stack

> *I had one simple idea about telling friends about arts and technology events. People in the community suggested everything else to us, and that's our theme. We're really run by the people who use the site. We just run the infrastructure, and help out with problems.*
>
> — Craig Newmark (founder, Craigslist.org)

Even though SharePoint implementations are way more common now than they used to be, I'm constantly amazed at how many nontechnical folks have heard about or used SharePoint. As a consultant and author, I'm constantly meeting new people from different walks of life. When the conversation gets around to what it is we do for a living — and I mention that I am a SharePoint consultant — the typical response is usually along the lines of, "Oh SharePoint! We use that at my work." Or "Oh yes, I use SharePoint at work. Can you help me with this thing it's been doing?" Or "I have heard of SharePoint. A friend of mine uses it." On the one hand, it's reassuring to know that SharePoint is so pervasive that even people who are as far as possible from a techie mindset still use it in some fashion. On the other hand, I'm always amazed about what people think SharePoint *is.* I've heard it described as a Web site, a way to send documents, and almost everything else high-tech (except maybe a death ray).

In this chapter, I give a brief history of SharePoint and explain exactly what all the hype is about. Because SharePoint is as much a development platform as it is a software product with many different pieces, I stitch those pieces together for you so you're armed with a big-picture view of what to expect as you begin developing SharePoint solutions. Finally, I introduce SharePoint development to you and fill you in on a secret that SharePoint developers would rather not discuss: You don't have to be a programmer to develop solutions on the SharePoint platform. If you can use a Web browser, then you can develop a SharePoint solution. If you're ready to add "SharePoint Developer" to your résumé, all you have to do to get started is read on.

Everyone Can Be a SharePoint Developer

When most people think of a "developer," they think of a computer-science geek sitting in a basement with a can of Mountain Dew banging out complex computer code. In the past, this was often not far wrong; getting a computer to do something specific to solve your problem required writing a computer application using special programming languages that would then be compiled down into the 1s and 0s that a computer could understand. These languages were, and still are, complex and mysterious and require an extraordinary amount of dedication and knowledge.

If you're a hard-core programmer, no sweat — your skills will always be useful. Until computers take on thoughts of their own, we need programmers just as we need surgeons for our (very human) bodies. The good news, however, is that software applications are evolving in a very real and exciting way. Tools are becoming available that give users and analysts outside the IT department more effective ways to command their computers to create solutions to real-world business problems. That's where SharePoint comes in.

SharePoint 2010 provides a number of tools and features designed to shift more of the development power away from programmers and into the hands of the people who understand their particular business problems best. By and large, these are businesspeople who could care less about technology and are just trying to do their jobs. The SharePoint development tools allow users to create no-code solutions in a self-serve manner; no need to interact with IT. This takes some of the time burden off of the IT department so it can focus on (say) creating stable, redundant, always-available, and secure computing environments. Everybody wins — at least that's the idea.

As a technology consultant, I'm constantly filling the role of intermediary between the business users and the technology people. I'm very excited to see this quiet change in development responsibilities. Because the SharePoint 2010 platform and tools for developing business solutions are designed with the end users in mind, a new level of business efficiency and productivity becomes available.

Tracing the Origins of SharePoint

Computers created an information revolution that fully blossomed as they started to get connected together into large networks. The biggest network of all, the Internet — and the fact that computers became cheap enough to sit on nearly every desk at home and work — created a recipe for sharing information. To fulfill this need, technology such as e-mail and the World Wide Web cropped up in the 1990s. Soon Web sites flourished, nearly everyone had an e-mail address, and information flowed freely. Throughout the 2000s organizations rapidly adopted corporate e-mail systems in order to facilitate communication.

In addition to e-mail, organizations also adopted large enterprise computer systems that handled everything from inventory to human resources. These systems are known as Enterprise Resource Planning (ERP) — with some of the biggest players being SAP, Oracle, Microsoft, and Sage. As companies adopted ERP systems in the '90s, others followed suit to stay competitive. As we head into the 2010s, the next wave of efficiency and productivity for knowledge-based workers will come from communication, collaboration, and centralized information sharing.

As SharePoint has come of age, it's put more tentacles into the Microsoft Office applications — resulting in a centralized, integrated portal platform that's controllable from familiar applications — so the race to implement SharePoint continues. Various companies offer software products for creating and using a company portal space — but so far the clear winner is SharePoint.

So What Exactly Is SharePoint, Anyway?

Microsoft highlights these defining characteristics of SharePoint:

- SharePoint Server is a suite of integrated server capabilities. In essence, if you install SharePoint on your server, you give it a new range of information-wrangling powers that work together.
- SharePoint manages content (your business information) and provides enterprise-wide search capabilities. Wherever the needed information is in your organization, SharePoint can find it, make it usable, and keep it secure.
- SharePoint, when used to full advantage, accelerates business processes that require collaboration by making communication easier (and information easier to share) across departmental boundaries.

- SharePoint can help improve the administration of your company server, extend existing software applications with server-based capabilities, and help a wide range of software and hardware play well (and work well) together.

These points are great and tell you what SharePoint can do, but they don't really tell you what it is. If you're like me, you're looking for an explanation that has some roots. Something concrete. Is SharePoint a software program or a standalone supercomputer? Is SharePoint something that runs on my local computer like Microsoft Office or does it live somewhere in a data center?

To get a handle on SharePoint, start from the bottom and work up through its components — which you can see in Figure 1-1. The following sections explain each part of Figure 1-1 in more detail.

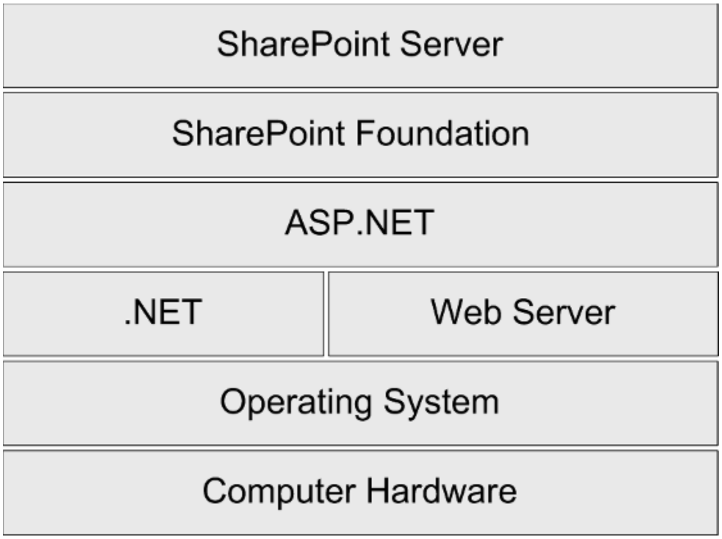

Figure 1-1: The components that make up a SharePoint environment.

Starting with the roots — computer hardware

Every computer system starts with physical components — central processing unit (CPU), memory, a hard drive, power supply, and motherboard. Typically a computer comes into your workplace in one of two ways: You can buy all the individual pieces and put them together yourself or you can purchase an already-assembled computer from a vendor that specializes in building them, such as HP, Dell, or IBM. What this means is that a range of components from various makers and vendors have to work together without compatibility hassles. In order for all these hardware components to work together, you need a specialized piece of software designed to make everything hum. This software — the operating system (OS) — is what Microsoft began its corporate life creating.

TIP

A new trend that you may have heard about involves virtual computers. A *virtual computer* (also called a *guest computer*) is essentially a self-contained operating system running within a host computer; it behaves just like a separate computer — except it interacts only with its host computer. The host computer handles all interactions with actual system hardware. You can have multiple guest computers — all performing different server functions — running on the same host computer; that, in essence, is *virtualization.* Running an entire operating system (such as Windows 7 or Windows Server) as an application within the host computer's operating system allows you to separate the various server duties to different guest computers without having to run separate physical hardware.

As you can imagine, the host computer becomes crucially important. You don't want the host to crash and take its guests with it, so virtualization companies such as VMware have created very clever software to manage host and guest computers. Because the guest computers are really just applications, they can be moved quickly between host computers (the physical computers) without even needing to be shut down. And nobody has to go get the forklift to move all those "computers."

The reason it's critical to understand how the hardware and operating system work is that SharePoint only runs on Microsoft operating systems. Although you can have a Microsoft operating system installed on a multitude of different hardware types — HP or Dell, for instance — SharePoint can only be installed on a computer running the Microsoft operating system. You might already have a server running a different OS (such as UNIX or Linux), but unfortunately you can't install SharePoint on it. There is one limited workaround: If your server computers are running virtualization software such as VMware, then the guest computers can be installed as virtual Windows machines — even though the host system is running (say) Linux. In this scenario, you're back in business and can run SharePoint on the virtualized Windows operating system. If all this is giving you a headache, — don't worry about it. Call in your trusty IT team; it's their bailiwick, and you pay them the big bucks to worry about it.

Software that talks to the hardware — the operating system

The software that makes all the hardware components actually do something — the *operating system* (OS) — usually comes in two major versions — one for end users (client) and one for the organization's heavy computing requirements (server). In the Microsoft world, both OSs are called *Windows.* Windows 7 is the latest client OS; Windows Server 2008 R2 is the latest server OS. Note that the R2 means Release 2. The previous version of the Windows Server OS was called Windows Server 2008, without the R2. (Okay, I could make a cheap *Star Wars* joke here, but I won't.)

If you're going to run server software such as SQL Server and SharePoint, you need to have the appropriate version of Windows Server OS installed on your server computer(s). Normally end users throughout the organization have the client OS running on their desktop or laptop computers. All those personal computers have to connect to the server computers (running somewhere in a data center) in order to interact with the server software.

Software frameworks and servers — .NET and IIS

The .NET software framework runs on the Windows OS (both client and server versions) and keeps the various hardware components on speaking terms with each other. (I discuss the .NET framework — and its role in maintaining compatibility — in more detail in Chapter 11.)

Also running on the Windows Server OS is the Microsoft Web server called *Internet Information Services* (IIS).

In nearly any discussion of corporate networking, you hear the term *server* used to describe both hardware (the server computer) and software (the server operating system that tells the hardware what to do). True, a computer running a server operating system *is* a server, but so is the operating system itself (a computer can't function as a server without a server OS) — *and* (are you ready for this?) so is the software designed to work with a server OS. They're all called "server." (It's like saying, "This is my sister Kate, this is my *other* sister Kate, and this is *her* sister Kate." Only worse.) But all sanity is not lost; get a grip and hang on for this example:

- Windows Server 2008 R2 is an operating system designed to run specialized software optimally — on a server computer.
- The networked computer on which you install the Windows Server OS is your server computer. Usually server computers are high-capacity and high-performance machines designed and built for that job.
- Server software applications act like short-order cooks and waiters: They serve up information to client computers in the form of Web pages, FTP sites, or e-mail communications. SQL Server and SharePoint Server are such applications.

Put those three aspects together, and you have a functioning server; just be careful which aspect of it you're talking about. And here's where the function of a server can help dispel the confusion of the term: What a server *does* is provide a consistent place where the network's users can access data. Because data can play many roles, servers can specialize accordingly — as (say) a database server, file server, or content-management server.

The *.NET Framework* is a bundle of computer languages, programs, and standards designed to make software programmers more efficient — so long as they're programming for Microsoft operating systems. The Microsoft Web server is called Internet Information Services (IIS) and it is responsible for serving up Web pages. Since SharePoint is all about Web pages IIS is a critical component. The portion of the .NET framework that is specifically designed to work with IIS and thus the tools Microsoft used to develop SharePoint is called ASP.NET. For more about ASP.NET read on.

TIP

The .NET framework has nothing to do with the `.net` that you often find at the end of a Web address such as `www.iis.net`. They are completely separate things that have nothing to do with each other.

A computer language for the Web — ASP.NET

ASP.NET is a specialized extension of the .NET software Framework for building Web applications (including custom Web pages) to run on — and be served up by — the IIS Web server (Microsoft's Web server). Typically you find ASP.NET used to build custom Web pages, often in programming languages such as C#.NET or VB.NET. (For more information about .NET and programming languages, check out Chapter 11.)

The ASP stands for Active Server Pages; the .NET declares it as part of the .NET Framework. So, even though ASP.NET sounds like a Web address, it isn't — although (stay with me, now) you *can* go to a Web address that incorporates the name — `www.asp.net` — for more information about ASP.NET the Microsoft product. ASP.NET itself is a framework for building Web applications on the Microsoft platform using the IIS Web server; the Web address offers information about the ASP.NET framework. (As if life isn't confusing enough, right?)

The first step into the SharePoint world — SharePoint Foundation

ASP.NET commands are what run SharePoint Foundation, previously known as Windows SharePoint Services (WSS), a basic set of software features that demonstrate some vital SharePoint capabilities. SharePoint Foundation is essentially a "lite," or limited, version of SharePoint: It provides some collaboration and communications features (such as lists and document libraries) that developers can build into custom applications and Web sites. Because SharePoint Foundation is built on the ASP.NET framework (which is, keep in mind, an extension of .NET), it provides endless opportunities for customizing applications. SharePoint Foundation, however, isn't quite powerful enough to pinch-hit for industrial-strength SharePoint if an organization is large and complex. (But then, that's what the Enterprise edition of SharePoint is for, as detailed in the next section.)

A finished product — SharePoint Server

Microsoft used SharePoint Foundation as a solid footing on which to build SharePoint Server. The goal was to create a software product that could solve business problems with its built-in features, cutting down on the time-intensive (and expensive) customization of software.

As a full-featured version of SharePoint, SharePoint Server is designed as a large-scale, enterprise-level portal platform: Your organization can use it for content management, communication, collaboration, information portals, doing enterprise-wide searches for specific information, and the documentation of business processes (even including the design of forms).

Untangling the versions and editions of SharePoint

Like other software, SharePoint has a history of versions and editions. A *version* is a dated (or numbered) release of a software product that usually incorporates significant updates. For example, the previous version of SharePoint was released in 2007 and the latest version of SharePoint was released in 2010. Each version has two primary editions. The first is a “free” edition of SharePoint that comes along with the Windows Server operating system. The second is a deluxe edition, purchased separately.

The previous version of SharePoint consisted of Windows SharePoint Services (WSS) 3.0 — the free edition — and Microsoft Office SharePoint Server (MOSS) 2007 — the deluxe edition. The current version of SharePoint consists of SharePoint Foundation 2010 (the successor to WSS) and SharePoint Server 2010 (the successor to MOSS 2007). One great aspect of this shift in marketing terminology is that Microsoft actually simplified the product names (what a concept!). Now instead of the two major editions of SharePoint being called WSS and MOSS — do you remember what they mean (no fair peeking)? — the two SharePoint editions are simply called SharePoint Foundation and SharePoint Server. Easy enough to remember.

You may wonder why there isn’t just *one* SharePoint edition. Here’s the short answer: Because no two businesses are exactly the same in terms of size, complexity, or mission — and Microsoft wants to offer editions of SharePoint that all will find appealing. Thus the Foundation version of SharePoint and the deluxe Server version. If you’re considering the Foundation version, note that “free” is relative — and (in this case) proprietary: SharePoint only runs on the Windows operating system. In order to get the “free” Foundation version of SharePoint, you have to purchase the operating system. The Foundation version, essentially SharePoint lite, contains features and functionality that are critical to organizations of all sizes but doesn’t include heavy-duty enterprise-level features such as Excel Services, InfoPath Services, PerformancePoint Services, Visio Services, and Access Services.

TIP

The specific features of each edition of SharePoint can be found online here:

```
sharepoint.microsoft.com/en-us/buy/Pages/Editions-Comparison.aspx
```

To segment SharePoint Server even further, Microsoft breaks the licensing of the product into two primary categories. The first is geared to sites that face the wilds of the Internet; the second is geared to sites that face the (internal) corporate intranet. Each of these site categories, *external-facing* and *internal-facing* respectively, comes in a Standard edition and an Enterprise edition.

Climbing the SharePoint development ladder

The easiest way to think about ASP.NET, SharePoint Foundation, and SharePoint Server is to picture them as rungs on a ladder of software sophistication: Each adds capabilities and value as you move up in complexity and scale.

Starting (relatively) simple, if you need a specific solution that provides features such as communication, collaboration, and document management, then you could pay developers to build all those capabilities from scratch. Just be prepared to make a time investment. Trust me on this one: Back in grad school, a team of us did just that, using the Java programming language. Our Web application was a content-management system that tracked electronic content and allowed people to check content in and out, as well as purchase content from an online store. This took us more than 6 months to put together; it involved team members from Germany, China, Colombia, and at least one other exotic locale (San Francisco).

So suppose you've given your Microsoft-savvy developers a similar task: "Build nearly everything from scratch and come up with a solution that provides communication, collaboration, and document management." They might assume they'll have to work the whole thing up in ASP.NET. But if they start with the SharePoint Foundation framework, they can use the ready-made document-management and collaboration components of SharePoint — and then just build and customize the rest of the solution to your specifications. Definitely faster.

But suppose your company has finally graduated from, "Where did I put my Word document?" to "How can we integrate our portal with our business in order to increase efficiency and productivity?" For that you need heavy-duty bang for your buck.

That's why Microsoft used ASP.NET and SharePoint Foundation to build into SharePoint Server nearly all the features you'd want in that custom-made solution. SharePoint Server is customizable, of course, but Microsoft has already done most of the heavy lifting by building the product. All you have to do is pay for it, install it, and begin developing solutions on the platform.