SharePoint 2010 Workflows IN ACTION

Phil Wicklund
Phil Wicklund

SAMPLE CHAPTER

MANNING

SharePoint 2010 Workflows in Action

Phil Wicklund

Chapter 1

brief contents

Part 1

Introduction to SharePoint workflows

This book is divided into three parts. With part 1, my goal is really just to introduce you to SharePoint workflows. Chapter 1 kicks this off by informing you of the types of workflows as well as what SharePoint objects on top of which you can run workflows. You'll also see an overview of the various tools used in the building of custom workflows, as well as the new workflow features that are a part of the 2010 release of SharePoint.

Chapter 1 also takes you through the process of analyzing and diagramming a workflow. This process is important to go through because different types of workflows won't work well for some tooling choices.

Chapter 2 puts workflows into practices. You'll be lead through the process of creating your first workflow utilizing one of the out-of-the-box workflows in SharePoint. In addition, you'll get familiar with how to configure and administer workflows. This includes learning how to add and remove workflows, start and stop workflows, and view a workflow's status and history. At the end of the chapter, you'll find high-level business cases for the rest of the out-of-the-box workflows.

1 SharePoint workflows for your business processes

This chapter covers

- Introducing SharePoint workflows
- Adding workflows on SharePoint objects
- Building a custom workflow
- New workflow features in SharePoint 2010

Business processes surround us and affect the typical employee daily. Whether you like it or not, the company you work for depends heavily on processes to get things done and be profitable. Someone who makes burgers at a fast food outlet, for example, has to follow a specific process that will transform raw materials into a finished burger.

Workflows are systems that manage the execution of a business process. They solve many of the most troubling problems that workers face. The burger outlet process is simple, but there's no doubt that large companies have complicated business processes, and it can be difficult to determine how far a process has progressed and what is delaying it.

Consider how often business processes are hindered because of poor communication. Does your business process live or die entirely by email? Email has become the default communication method for everything from conversations and decisions to tasks and documents. Consider a process that runs when a new person is hired into your company. That employee needs a new account, email, badge, phone number, benefits, direct deposit, and contract. In many cases, getting all that accomplished involves many people who communicate through email. Inevitably, things get lost. Email works for small companies, but what happens if you onboard 50 people per day? You need a system that will manage all of these activities; otherwise, you'll have confusion and inefficiencies. You need a workflow.

This chapter defines a workflow and shows how it relates to your business processes. We'll talk about how workflows function within the Microsoft® SharePoint platform and the architecture of a SharePoint workflow. After you've learned those basics, we'll take a closer look at all the tools and applications that go into building workflows in SharePoint and you'll discover numerous options. Beyond this introductory chapter is a world where your business processes come to life. The rest of the book is about building your company's workflows on the SharePoint platform. So, let's make sure we're all speaking the same language.

1.1 What is a workflow?

A workflow is primarily described as a process that manages the flow of work among individuals, offices, departments, or entire companies. Some work depends on numerous people or systems for completion. As these recurring dependencies are identified in a company, a business process emerges. Business processes run throughout a company and are often similar even in companies of different types.

Take, for instance, a business process that manages expense reports. Most companies need a defined business process to manage the submission and approval of employees' monthly expenses. The flow of work in figure 1.1 shows an employee tracking his expenses and then submitting them electronically to a manager.

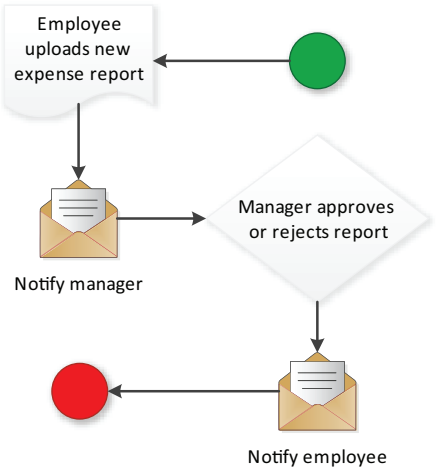

Figure 1.1 A workflow is generically described as a business process. This example shows a common workflow that manages an expense-reporting business process.

The flow is based on a business logic that determines who needs to approve the expenses and how the individual is reimbursed. A workflow helps to negotiate the execution of the steps in a process like this.

Business processes run regardless of whether a workflow manages them. Some

business processes are self-contained and easy for people to manage. Others are much more complicated and difficult for people to comprehend.

It's with these more complicated business processes that workflows show their value to a company.

Workflows can bring value to a company by highlighting where in the business process the flow of work is currently executing. Workflows can also help a company automate their business processes. Consider again the expense report example. If your business process allows for all food expenses less than $100 to be automatically approved and sent to accounts payable for reimbursement, the workflow could manage this business logic and automatically approve the expense, without having to directly involve your manager.

Workflows are also good at managing parallel processes or multiple instances of work running at the same time, for example, in a manufacturing company. A car manufacturer could have a workflow for the engine construction, and another for the frame, and another for the interior. Then a parent workflow could manage all of the *child* workflows and start another process as soon as a dependent workflow finishes.

It's easy to see that your investing in workflows would help to manage and automate your company's business processes. Minimizing human dependencies in business processes always saves a company's money. Because human costs are always the most expensive investment a company makes, let's make the people in our organizations work as efficiently and effectively as possible. That's what makes workflows such a great investment.

1.2 How does SharePoint help?

A SharePoint workflow is an automated flow of objects through a sequence of operations that are related to a business process. An object in SharePoint is a document or an item in a list like an announcement or a task. For example, one of the workflows that you get when you install SharePoint is the Approval workflow. You can attach this workflow to a document in a document library and specify individuals who need to approve the document for use before another action can occur.

> **SharePoint document libraries**
>
> SharePoint, in addition to being a collaboration platform (teammates sharing information), is a document management system. A document library in SharePoint is the tool you can use to upload documents into SharePoint.

The expense report system (figure 1.1) is a common example of document library use. Within SharePoint, users can upload their expense reports into a document library. The upload action will initiate the Approval workflow on the document, and a series of individuals will receive an email stating they need to approve the expense report. When all those individuals have approved the expense report, the document can be routed to the payroll team site where a payroll officer processes the expense report.

A SharePoint workflow, like the document Approval workflow, could be set up to manage the business process from start to finish. The workflow will handle all user interaction within the system. It will also manage the point of execution in the workflow. Additionally, SharePoint will provide an out-of–the-box user interface that reports on the status of the workflow, or, more specifically, who must act on the workflow before it can continue, or if it has finished executing.

This out-of-the-box experience is a compelling reason to manage your business processes within SharePoint, because it provides a user interface and other workflow fundamentals like security, reporting, and logging. These features make SharePoint workflows and your business processes a powerful combination.

Another great strength of SharePoint workflows is that individuals who are not technically savvy can configure their workflows directly through the browser window. Consider the expense report system again. If a company built this system from scratch, it would cost much more time and money because they would not have all the fundamental components that SharePoint provides out of the box. Rather, you can empower your end users to build these business processes and, at the most basic level, all they need is a browser and possibly a few minutes. That's cost effective!

In figure 1.2, notice that you can manage the settings of the document library that contains expense reports.

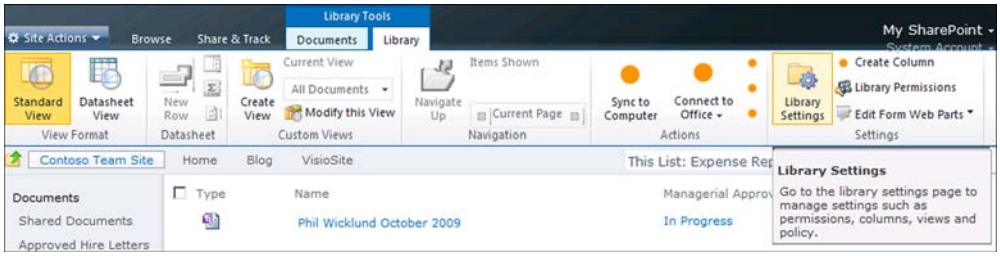

Figure 1.2 **To manage workflows on a list or library, go to that list or library's Settings page.**

In Library Settings, there's a Workflow Settings link under the Permissions and Management heading. On the Workflow Settings page (figure 1.3), you can add a workflow to a library. Select Add a workflow and choose the one you want. It's that easy!

After adding the workflow to the library and initiating the workflow process, a new column (figure 1.4) will display in the document on which the workflow process is running.

This column will track the execution point of the workflow. The workflow

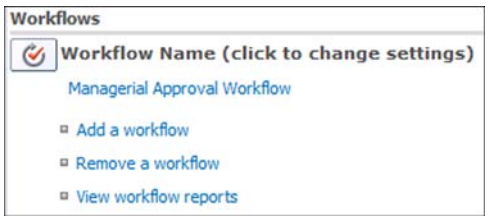

Figure 1.3 **Within the Settings page of a list or a library, you can add a workflow.**

Figure 1.4
After a workflow is started on a list item or document, an autogenerated column appears, showing the workflow status.

might be halfway through the process, waiting for user interaction. A manager might need to approve the expense before it can continue. This status would show up in the new column such as Pending Manager Approval. After the workflow has finished, Completed will show in this column.

1.3 SharePoint as a technology platform

We've been discussing a workflow from a business perspective and its execution within SharePoint. This only scratches the surface of SharePoint workflow foundations. SharePoint workflows leverage a separate platform called Windows Workflow Foundation (WF), which is part of the .NET 3.0 application development framework. This foundation has many applications totally unrelated to SharePoint—in fact, you can use WF to build all sorts of workflow-enabled business applications that never touch or interact with SharePoint. SharePoint benefits an application developer by providing a robust user interface and implementing the necessary persistence services required to run WF workflows. This means that SharePoint will manage the persistence of a workflow if it becomes idle and provide a user interface that end users can employ to start and stop workflows and determine where the workflow is executing. For example, the expense report workflow needs to be persisted as it awaits a manager's approval.

SharePoint also has out-of-the-box workflows built on top of the programming layer (figure 1.5) and, with tools like SharePoint Designer, you can customize those workflows if they don't meet your requirements. If you determine that a custom workflow is necessary, it's important to consider what type of workflow your custom workflow would need. The WF architecture and SharePoint support two types of workflows—sequential and state machine. Each type supports a unique type of business processes. Before we dig into the differences between these two types, let's take a look at the architecture of the foundation we're building on.

1.3.1 Windows Workflow Foundation architecture

The WF architecture is built upon three main tiers of services: the hosting layer (first tier), the runtime layer (second tier), and the programming layer (third tier) as seen in figure 1.5. You could say that SharePoint workflows sit on top of the third tier's services as its foundation. Out-of-the-box workflows, SharePoint Designer workflows, Visual Studio workflows, and our expense report example would build on top of this foundation.

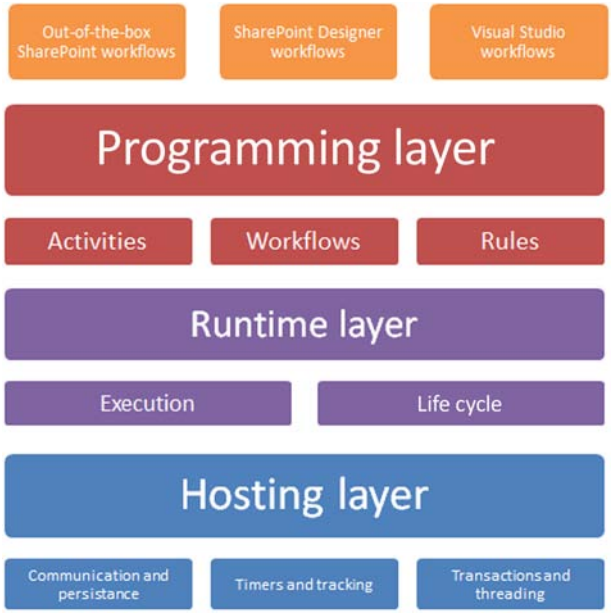

Figure 1.5 SharePoint workflows build on three layers of the Windows Workflow Foundation architecture. The programming layer is the interface between SharePoint and WF and resides on top of the core layers that manage the runtime and hosting.

HOSTING LAYER

A workflow is not an application and needs a host process in which to run. Similarly, if you install the .NET framework on a server, you don't have a line of business applications when you click Finish on the install wizard. WF acts as a platform you build on. From a hosting perspective, the WF requires the application to implement a few things to keep the lights on.

As previously mentioned, part of what is required of the application is the host process. The application must also provide persistence capabilities. A workflow is typically long running, meaning that it may start and then suspend for a while, possibly even many months. The *state* of the workflow needs to be persisted while the workflow is waiting for an action to occur and, when that action occurs, the workflow should resume where it left off.

Another area of responsibility for the hosting layer is to provide timer and tracking capabilities. As previously mentioned, a workflow may be suspended as it waits for an external action to occur or it may be time bound. For example, maybe the expense report workflow assigns a task to the manager to approve the report, but that task is not completed for seven days. At that point, the workflow wakes up and reassigns the task to someone in payroll. This also relates to tracking, in that you'll want to know, or track, where in the process the workflow is currently executing. In a nutshell, this is part of the responsibility of the timer and the tracking aspect of the host process. Transactions are another important aspect of the hosting framework, in that you can leverage transactions to roll a workflow back to a previous state if an error occurs, for example.

RUNTIME LAYER

This layer represents all the core services that come with WF. For instance, at runtime, the tracking, scheduling, and persistence services are all performing WF-critical activities that negotiate the workflow's execution and life cycle. This layer has interfaces that the hosting layer uses to connect the outside world to the WF engine.

PROGRAMMING LAYER

The programming layer is the SharePoint developers' favorite layer and is typically the only layer they need to worry about. This layer has out-of-the-box activities (actions for SharePoint Designer workflows) that can perform various functions in the workflow, and it allows for custom activities and rules that workflows interact with. Activities are the building blocks of a workflow. Activities do the work. A workflow is a structure that contains the activities and manages the choice and the timing of activities' execution. Rules allow the developer to declaratively create business logic that the Visual Studio workflow can use to structure its activities. For example, rather than hard-coding logic into the workflow, you can create a rule for that logic. Then, subsequent changes to the rule won't require an update to the workflow's dynamic-link library (DLL). Conditions in SharePoint Designer are loosely similar to rules in Visual Studio workflows.

1.3.2 Types of workflows

Workflows execute from one to another step in the process in two ways. A workflow is either sequential—in that the steps within the workflow execute sequentially, one after another—or a workflow is a state machine, whereby it executes in no particular order. A sequential workflow always progresses forward, never going back to a previous step (figure 1.6).

A state machine, on the other hand, has no such constraint but moves from one state to another until the logic concludes the workflow has completed. A good example of a state machine is a bug tracking workflow that tracks bugs in a computer program (figure 1.7).

When the workflow starts, the bug may be placed in a pending state, where it waits for a developer to be assigned to the bug and begin working on the bug. Thereafter, the developer starts working on the bug and fixes it, putting the bug into a fixed state. When a bug is fixed, a tester tries to confirm the resolution of the bug. If they find that it was not fixed, they place the bug back into a pending state. This ability to go back in time or to a previous state is only available with state machine workflows.

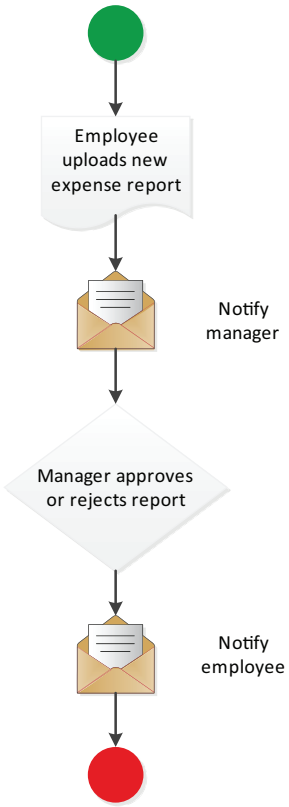

Figure 1.6 A sample sequential workflow that, in the process, always advances forward, never backward

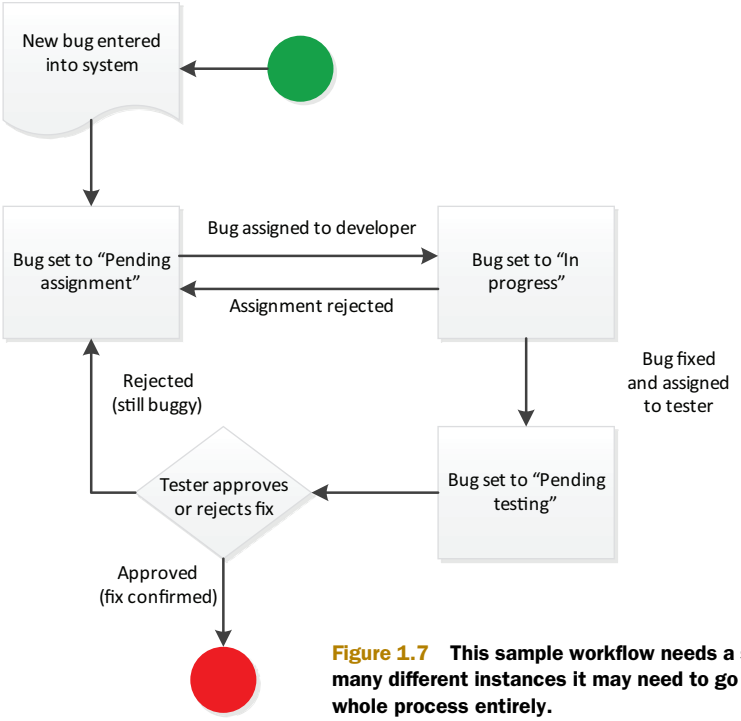

Figure 1.7 This sample workflow needs a state machine because in many different instances it may need to go back a step or repeat the whole process entirely.

Some business processes will require a state machine and others won't. It's important to think through the requirements of your business processes before you begin building a custom workflow because it's difficult to change a workflow from being a sequential workflow to a state machine workflow and vice versa. As you progress through this book, you'll notice that some workflow tools can do only sequential workflows (as with SharePoint Designer), whereas other tools can do both types (as with Visual Studio). If you start building a workflow with a tool, and that tool doesn't support state machines, you may find yourself starting over when you realize you need to change course. The bottom line is—understand the business requirements to determine which type of workflow is required before you start.

1.4 Workflow-enabled SharePoint objects

Now that you have an understanding of what a workflow is, you might be wondering how a workflow displays and shares information with users. For that task, SharePoint workflows depend on many different types of objects, such as lists, list items, libraries, documents, forms, content types, site columns, views, web parts, sites, and site collections. Going back to our expense report example, the workflow runs on a document that was uploaded in the document library. In addition to documents, there are several other types of SharePoint objects that workflows can execute on.

1.4.1 List items

As with documents, workflows can run on generic SharePoint list items. For instance, you could set up an approval process on an announcement list. With this setup, announcements won't be displayed to end users until they're approved. Another effective use of list items is on task lists and issues lists (both are out-of-the-box SharePoint list types). When a task or issue is assigned to someone, and that individual resolves the task or issue, a workflow might forward the task to another individual who is responsible for verifying its completion before it can be finalized. If that individual finds an error, the Workflow could reassign the list item back to the original user. Figure 1.8 shows the workflows menu item on a list item.

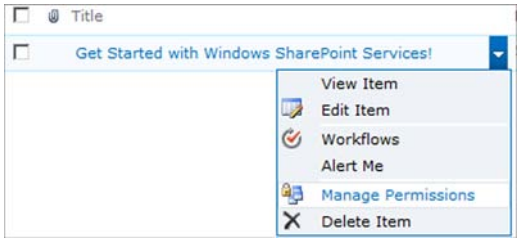

Figure 1.8 **To start a workflow on a list item or document, click the dropdown on that item, and then select the Workflows menu item to take you to a page where you can initiate a new workflow instance.**

Through this menu item, you can start a new workflow instance on the item.

1.4.2 InfoPath forms

When an InfoPath form has its data stored in a form library (figure 1.9), it's considered a document and it falls under the document library category of SharePoint objects.

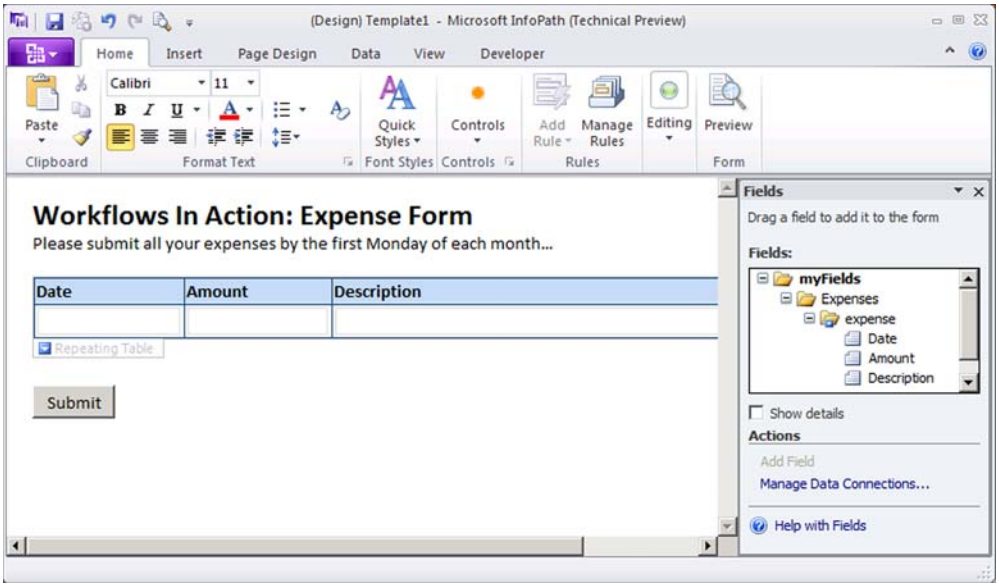

Figure 1.9 **InfoPath forms are excellent tools for developing custom forms for your workflows. This example shows the InfoPath Office client in design mode.**

It's commonplace to attach a workflow to a form. Take, for instance, the expense report system. Often, a company requires their employees to use a standard expense report template. Such template usually has special requirements and a look unique to that company. The Microsoft Office InfoPath 2010 client application is an effective tool for building forms that your users can fill out.

The strength of InfoPath is its ability to make form creation easy even for novice users. If you're familiar with Microsoft Word, you'll catch on to InfoPath quickly. InfoPath gives you the flexibility to control the look and feel of a form. You have far more user interface flexibility with InfoPath than with an Excel document. As with Microsoft Office documents, workflows can be bound to an InfoPath form and, when the user fills out all the appropriate areas in the form, the workflow can manage the business process behind it and get that form approved or denied. We discuss other form options in section 1.6.4 of this chapter and in greater detail in chapters 7 and 9.

1.4.3 Content types

Content types in SharePoint are an important concept; however, they are already highly documented elsewhere and won't be covered in great detail in this book. At a basic level, content types are a way to package pieces of metadata and make metadata collections reusable. For instance, let's say you wanted three columns on every list or library in your site collection. Rather than go to each list and add each column manually, you can create a content type that has those three columns and then add the content type to the lists or libraries. This can be a significant time saver and provide substantial reuse benefits when you need to make changes to the content type.

Additionally, a content type can have one or many workflows assigned to it. If you have a complex business process with many types of workflows that all need to execute simultaneously, deploying that workflow to a content type may be a good idea. When a workflow is deployed into a content type, new instances of that workflow can be initiated wherever list items of that content type exist no matter which SharePoint list they reside in. This introduces reusability across more than one list to your workflow.

Note the new expense report dropdown (figure 1.10).

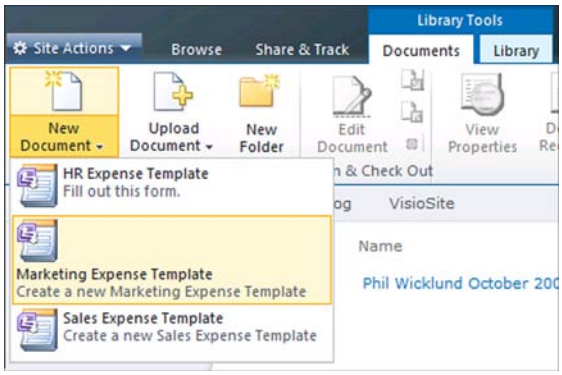

Figure 1.10 Workflows often run on a content type. As you add content types to a list or library, more options appear in the new dropdown allowing you to select a content type.

This demonstrates how you can use content types to allow several different expense report templates, for example, one for each department in the company. The form and workflow for the HR department's expense reports may be different from the workflow for the sales department, in which case using two different content types would allow users to choose a form and workflow that suites them.

1.4.4 SharePoint sites

You can also bind a workflow to a SharePoint site, which is then called a site workflow. While all of the other SharePoint object examples boil down to list items (whether it be a document, form, or content type), a workflow deployed onto a site can run actions on and react to events across all lists, document libraries, and items in that site. For example, take a site that has many document libraries and many documents in each. A workflow that would be well suited for running at the site level could check each document within that site and ensure that all the documents have been routed for approval and that none have been declined. Workflows can execute across an entire SharePoint site and are initiated from within View Site Content.

1.5 Out-of-the-box SharePoint workflows

Now that you have an idea of all the types of objects in SharePoint on which workflows execute, we should explore further the workflows that come out of the box. In section 1.2 of this chapter, we introduced the Approval workflow. This is one of several workflows that are available in SharePoint. SharePoint provides six workflows out of the box that end users can configure on their sites. You don't need to be a programmer to introduce valuable workflows into your organization. These workflows are available for adding to various SharePoint objects right through the user interface and require little if any configuration. Figure 1.11 shows the Start a New Workflow page for an Excel document in SharePoint.

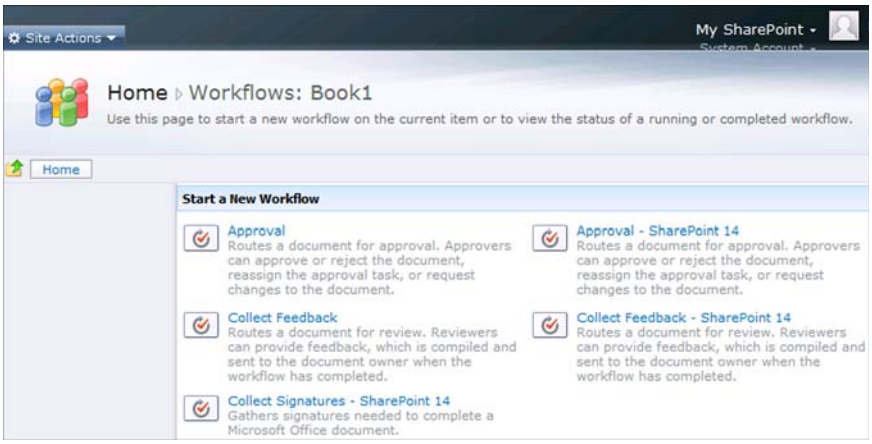

Figure 1.11 The out-of-the-box workflows that come in SharePoint. This is a list of out-of-the-box workflows that can be started on an Excel document.

It's easy to start a workflow that kicks off a wizard-like experience for end user configuration. Let's get a bird's eye view of all of the out-of-the-box workflows.

CHECK OUT CHAPTER 2 FOR MORE DETAIL Refer to the last section in chapter 2, "Additional out-of-the-box workflows," for a more detailed overview and a flow diagram of each of the six workflows.

1.5.1 Three-state workflow

The Three-state workflow by default is leveraged on issue tracking lists in SharePoint. The issue tracking list tracks an issue through three states—Active, Ready for Review, and Complete. This workflow can be customized and used in other lists or in custom SharePoint lists, and the names of the three states are configurable.

SHAREPOINT FOUNDATION VERSUS SHAREPOINT SERVER When you install SharePoint, you have at least SharePoint Foundation installed. SharePoint Server on the other hand is an add-on, which is installed on top of SharePoint Foundation. You get a few extra out-of-the-box workflows when you purchase the Server. Foundation comes with only one workflow—the Three-state workflow. Server gives an additional five out–of-the-box workflows. Regardless of which version of SharePoint you install, you have all of the functionality for building custom workflows.

1.5.2 Approval workflow

One of the simplest workflows, the Approval workflow is certainly the most popular. It involves routing a piece of SharePoint content to all designated approvers requesting their approval or denial of the content. The submission process can either be serial, where the order of approvers is predetermined, or parallel, where any approver can approve at any time.

1.5.3 Collect Feedback workflow

The Collect Feedback workflow gives submitter the ability to acquire feedback for their peers on the status of the submitted document. This workflow routes the document to the specified team members, in which case each can weigh in and contribute their feedback on the document. After it has circulated through the team, the feedback is compiled and the submitter is notified.

1.5.4 Collect Signatures workflow

An alternative to the Approval workflow where the approval process makes no changes to the document, the Collect Signatures workflow will require each approver to place a digital signature on the document. After those signatures have been acquired through the workflow, you can take the document off line and the approval will still be recognizable. Note that this workflow can only be initiated from the Office

client, such as Microsoft Office Word. It cannot be initiated from the browser like other workflows can.

1.5.5 Disposition Approval workflow

This workflow allows you to manage document expiration and retention. This enables you to decide what will happen to documents when they expire. A possible option, instead of deleting a document, is to archive it and send email notifications.

1.5.6 Translation Management workflow

Use this workflow to help facilitate the manual process of translating office documents from one language to another. This workflow works with two list types—a Translation Management Library and a translators list. A document that needs to be translated is uploaded into the Translation Management Library, and translators in the translators list receive tasks to start translating the source document into their respective languages. When all the translation tasks are completed, the Translation Management workflow is completed.

1.6 Tools for building custom SharePoint workflows

Under the SharePoint umbrella, there are many layers of tools for a variety of audiences to use when building your SharePoint workflows. Some of them are entirely optional, and others are not. An end user might love digging into SharePoint Designer but be daunted by Visual Studio, yet Visual Studio may be necessary for what they're trying to do. It's critical to know about all the available tools, how they can be helpful, and what purpose each serves.

1.6.1 SharePoint Designer 2010

SharePoint Designer is a powerful tool that can be used to customize SharePoint sites. Many of its unique capabilities are used to change the look and feel (brand) of a site, create, add, and move web parts, and bring list data and external data onto SharePoint pages. You can do a great deal in SharePoint Designer including building workflows.

Building workflows with SharePoint Designer is a popular and widely used approach. The tool is easy to use because it provides the user with a wizard-like experience (figure 1.12) that is more familiar than the Visual Studio's code editor.

SharePoint Designer isn't a tool for the average end user, however. Microsoft would categorize the tool in the power users group—people who are not programmers but who are savvy enough to be proficient with other Microsoft Office tools like Excel and Access. Using SharePoint Designer is unlike using the browser, where things are simpler and more intuitive. This book will cover SharePoint Designer workflows in much greater depth in chapters 3, 4, and 5.

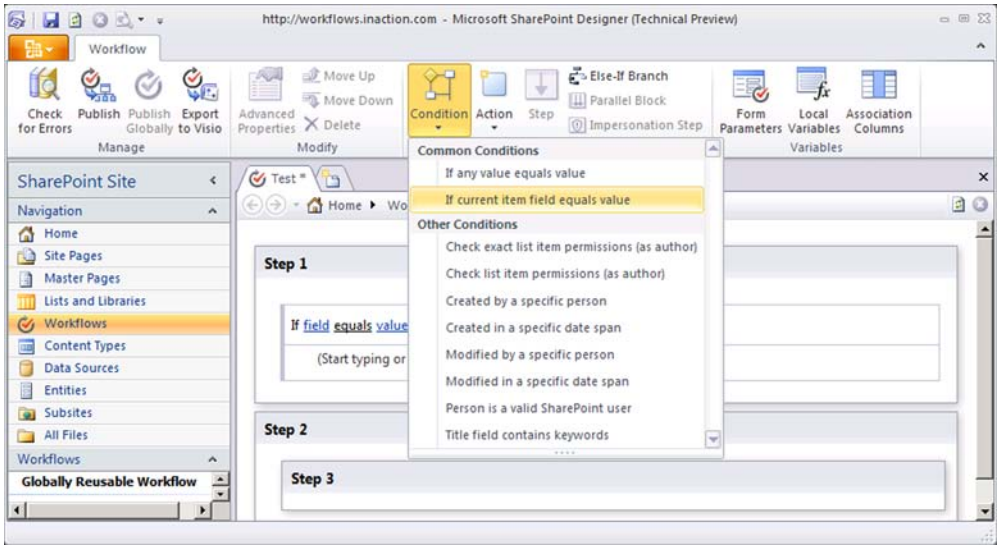

Figure 1.12 **The SharePoint Designer workflow engine provides a rich suite of customizations for workflows, allowing you to easily meet unique and sometimes complex workflow requirements**

1.6.2 Visual Studio.NET 2010

Although SharePoint Designer is a highly usable and robust workflow tool, your business requirements might require more than SharePoint Designer can deliver. This is where building custom workflows within Visual Studio comes into play. Visual Studio adds flexibility in terms of the types of activities your workflow can perform. This is because Visual Studio provides a full fidelity development experience, whereas SharePoint Designer is wizard based.

Visual Studio gives you a designer interface into which you can drop activities. Also, each workflow and activity will have its own code that you can call into or extend. Workflows built in Visual Studio leverage the .NET 3.5 Framework, and the Windows Workflow Foundation platform. Windows Workflow Foundation workflows can be packaged and deployed into SharePoint. This will allow you to meet even the most complicated business requirements.

Figure 1.13 shows what a workflow looks like within Visual Studio. The various activities are laid out on the workflow designer surface. By looking at the activity names and how they're associated with one another, you can see what the workflow is doing.

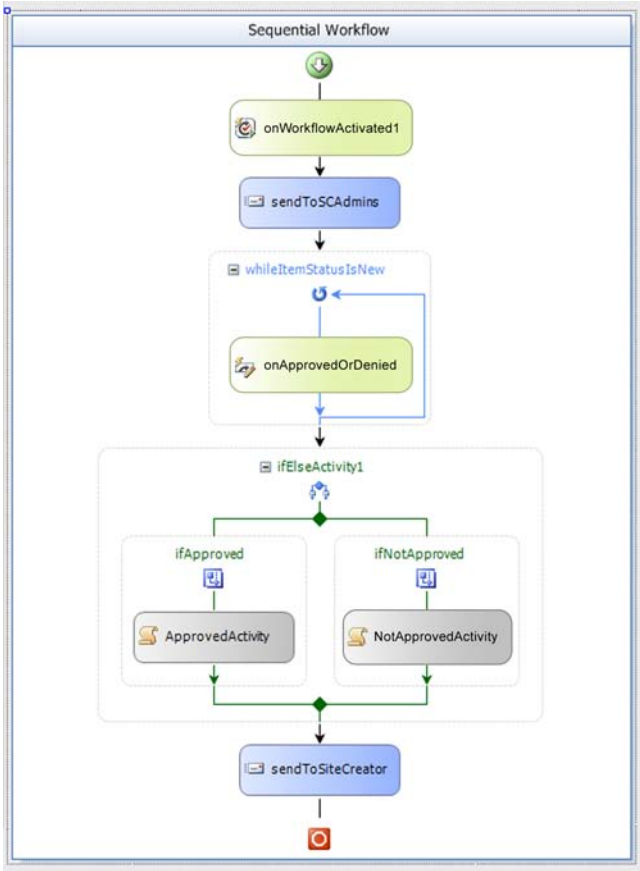

Figure 1.13 **Visual Studio is the utmost in workflow customization flexibility. This is a view of the workflow designer surface within Visual Studio.**

1.6.3 Visio 2010

Microsoft Office Visio 2010 comes with an excellent new feature for SharePoint workflow implementers—a new template called Microsoft SharePoint Workflow. This template (figure 1.14) allows business analysts to model custom workflows that need to be built out by power users (in SharePoint Designer) or developers (in Visual Studio). In earlier versions of Visio, you could do this using a generic flowchart template and adding shapes and connectors onto the Visio designer surface to model out the basic flow of events. With the new SharePoint Workflow template, power end users or developers can take the work you have done in Visio and import it into SharePoint Designer and then into Visual Studio. This reuse saves time and benefits the requirements integrity because of the connectedness among tools. Details on how to implement a Visio 2010 SharePoint workflow will be discussed in chapter 6.

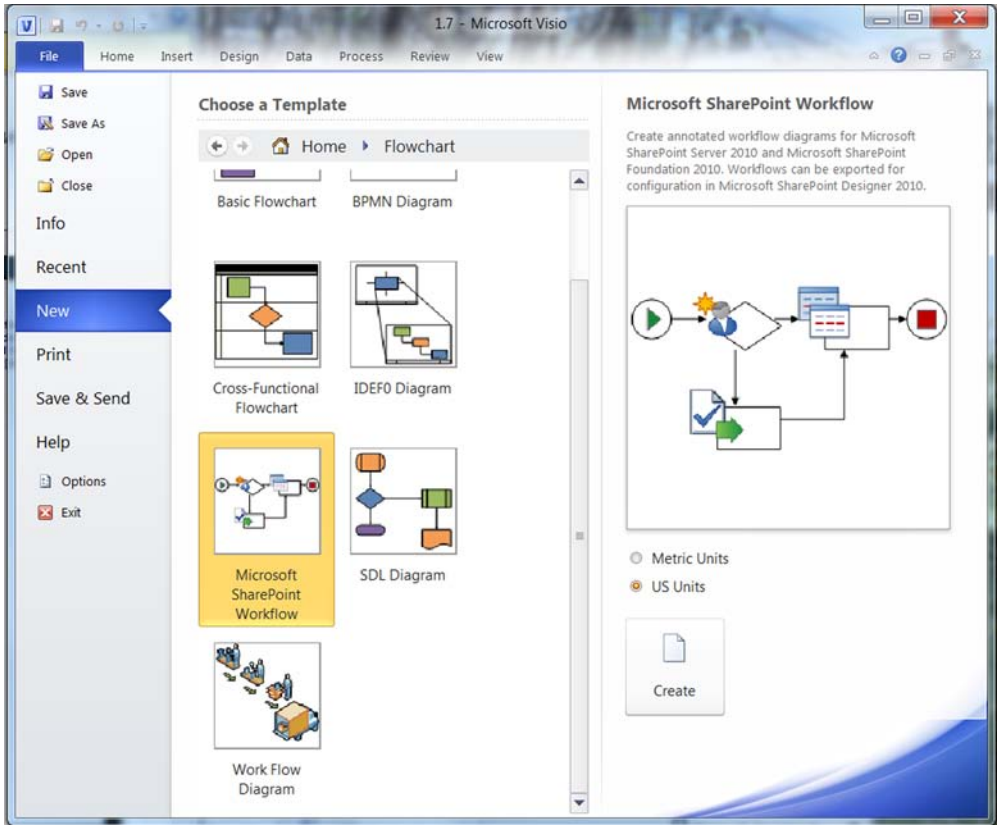

Figure 1.14 Visio 2010 comes with a SharePoint workflow template that can be used to model your business process before you build it.

1.6.4 Forms

Forms and SharePoint workflows make a powerful combination. Much of the data in a business process is captured within a form and, thereafter, the business process reacts to the values specified in the form. For an expense reporting workflow, employees typically fill out a form and enter their expenses. Submitting the form will initiate a business process to retrieve approval for that expense. Countless other form and workflow combinations can be found in an average business. Common form examples include forms for gathering personal information (health insurance benefits), purchase order requests, invoices, paid-time-off requests, new SharePoint site requests, and helpdesk tickets. This strong relationship between a business process and a form makes SharePoint workflows and forms a powerful combination. The retention and digital availability of form data, paired with the automation of these business critical processes, realizes substantial return on investment to an organization. Paper is the past, and workflows are the future!

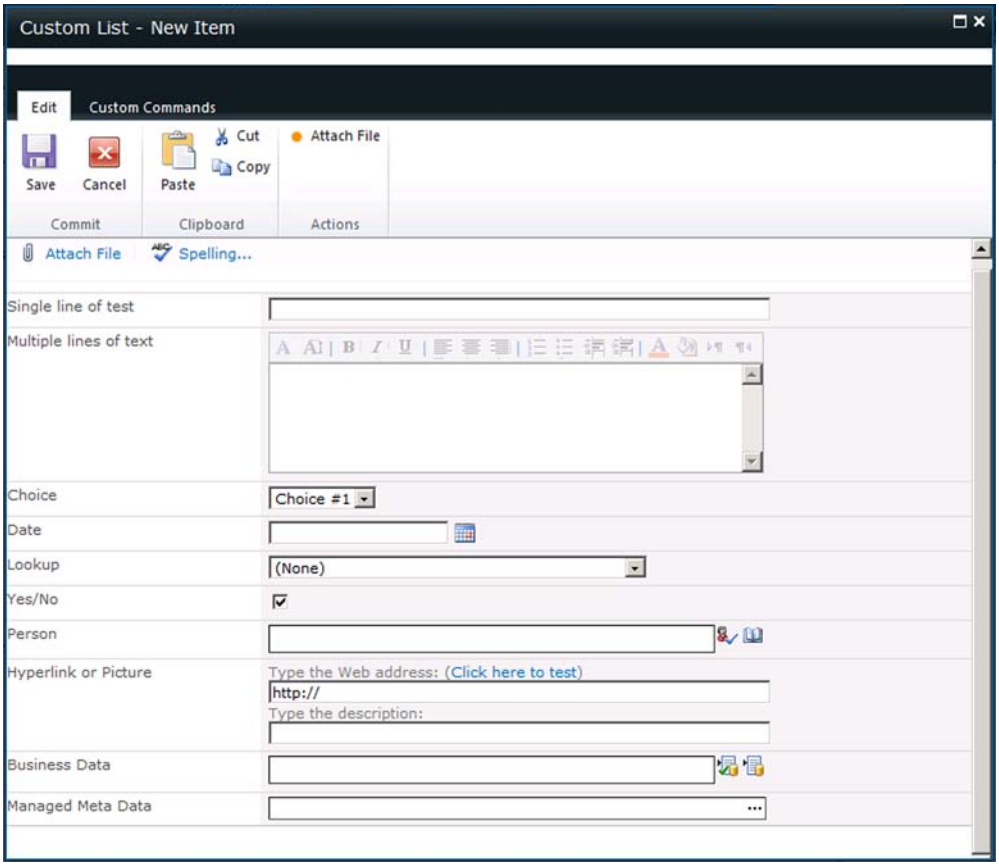

Figure 1.15 **An example of a form where a user can enter and submit data. Forms and workflows are powerful combinations. A workflow often reacts to a user's submitting information in a form.**

In SharePoint, three main tools are used in the building forms that workflows can execute on. You can use the out-of-the-box forms that SharePoint allows you to build by adding metadata to lists and libraries. For more complicated forms, you can use Office InfoPath 2010 and ASP.NET forms. InfoPath is an excellent nonprogrammer form design tool for end users, while ASP.NET tries to resolve the most demanding requirements developers face. Figure 1.15 shows a sample out-of-the-box form in SharePoint that can be used to gather information from a user and kick off a workflow.

Chapter 7 offers detailed instructions on form fundamentals such as building custom InfoPath forms and using forms in SharePoint Designer workflows. For Visual Studio workflow form, refer to chapter 9.

1.6.5 Object models

At times, you may need to programmatically interact with workflows through custom code. An example is when there's a custom report that shows the statuses of various

business-critical workflows on a manager's dashboard. Another example is when you need to start a new workflow on a weekly basis but you want the initiation of that workflow to be automatic and independent of human interaction. Both of these cases lend themselves to code deployment that programmatically interacts with the workflows through the respective object models. Object model techniques will also be covered in detail in chapter 12. With these object models, you can:

- Start or stop a workflow on a SharePoint object.
- Get a list of the running workflows on a SharePoint object.
- Detect and delete orphaned workflows.
- Report on the state of a workflow.

1.7 New workflow functions

With the release of SharePoint 2010, a host of new functionalities for workflows is available. This is true for custom workflows, where a few of the new features make developing workflows much easier. Take, for example, the introduction of the reusable workflow for SharePoint Designer 2010. With 2010, SharePoint Designer workflows can be deployed in a reusable fashion, which enables Designer users to work more efficiently. In the 2007 version, you had to recreate each workflow onto every list instance for deployment (except for Visual Studio workflows). In the 2010 version, you create and maintain a workflow in a single place. Moreover, you can install it onto many lists and receive updates if someone edits the original workflow. In addition to reusable workflows, many useful enhancements have been made to workflows in the 2010 release. Some of the key improvements are outlined in the sections that follow.

1.7.1 Visio 2010 SharePoint workflows

The new functionality in Office Visio 2010 will delight SharePoint business analysts. With Visio 2010, you can model your SharePoint workflows and leverage that model to help elicit business approval. The best part is that, after you've solidified the high-level flow, you can export the workflow as a template and import it into SharePoint Designer and start building all the steps! This will greatly improve the efficiency of requirements gathering and translation for developers. See chapter 6 for more information.

1.7.2 Customizing the out-of-the-box workflows

Have you ever used an out-of-the-box workflow in SharePoint 2007 but realized that it didn't do exactly what you required it to do? If so, you'll be pleased to hear that these out-of-the-box workflows can be customized in SharePoint Designer 2010 (figure 1.16). See chapter 5 for more information.

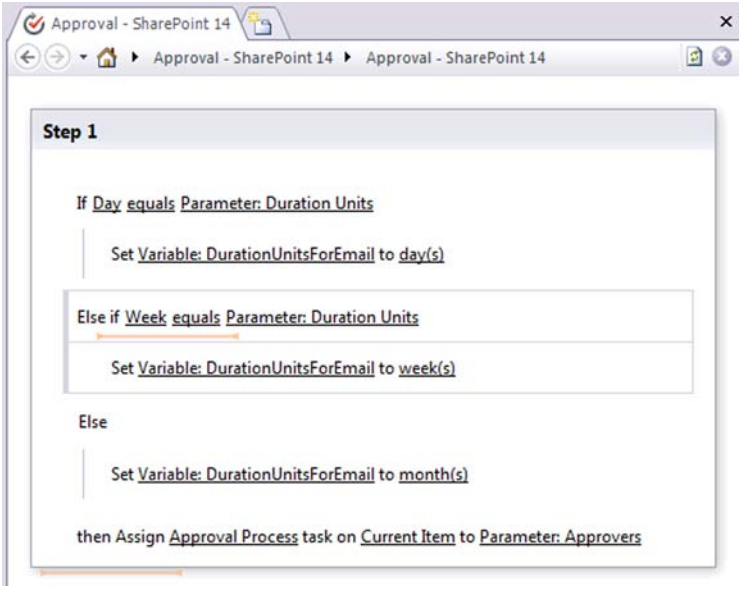

Figure 1.16 **Editing the out-of-the-box Approval workflow in SharePoint Designer 2010**

1.7.3 New actions and conditions in SharePoint Designer

SharePoint Designer 2010 offers a host of new out-of-the-box conditions and actions. For example, you can now manage permissions in a workflow, which you couldn't do before (think of the possibilities). Additionally, you can interface with a Records Center to satisfy your retention policies.

1.7.4 Reusable workflows

As mentioned in the introduction to this section, you can now create reusable workflows and deploy them to the site or site collection level to be consumed by various objects within that scope. The major benefit is that you no longer need to maintain more than one copy of the same workflow. Chapter 3 shows you how to build a reusable workflow in SharePoint Designer.

1.7.5 Site workflows

Site workflows take the concept of reusable workflows a step further. In addition to being capable of running on a list, library, or list item within a site, a site workflow can run on the site. A good business example of this is a workflow that ensures that the entire site's documents, regardless of the list in which they reside, are approved. A site workflow could iterate through all of the libraries and check each document to see whether it has been approved or not. There is a host of other applications for deploying a workflow. See chapter 3 for more information.

1.7.6 Task processing customization

Most workflows delegate tasks to certain individuals. When a task is assigned, the workflow typically waits for an action to occur on that task and then the workflow resumes processing. In SharePoint 2007, task processing was static—you could not alter the way the out-of-the-box workflows handled the tasks and events associated with them. In SharePoint 2010, you can fully customize the actions that follow task events. Events can react when a task is assigned, expires, is deleted, and is completed. When each of these events occurs, you can inject your custom activities to change the way the task processing flows. Task customization for SharePoint Designer workflows is discussed in detail in chapter 4. For tasks in Visual Studio workflows, see chapter 10.

1.7.7 Workflow templates in SharePoint Designer

In SharePoint 2007, you couldn't move a SharePoint Designer workflow from one farm to another. So, if you had a development, test, and product series of farms, you couldn't prototype a workflow in, say, development and then promote it to production. The 2010 version enables you to save a workflow as a SharePoint solution package (WSP) file and export and import that SharePoint Solution into another farm!

1.7.8 Viewing workflow status with Visio web access

A useful new reporting feature available in SharePoint Server Enterprise edition is the ability to view a workflow's status through a Visio diagram. If you first build your workflow in Visio 2010, and then import that workflow into SharePoint Designer, you can enable Visio web access on that workflow. Throughout the workflow's lifecycle, the Visio diagram will dynamically update to reflect where the workflow is currently executing. Notice the checkboxes in figure 1.17.

You can see the path that the workflow has taken and where it's executing. In this case, it has finished executing. For more information on how to set this up, see chapter 6.

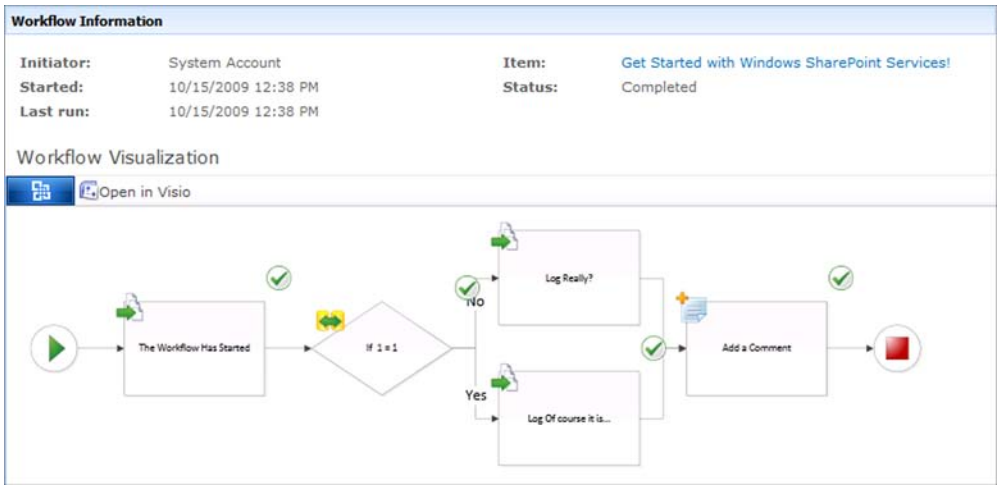

Figure 1.17 **Workflow status can now be observed through a dynamic Visio 2010 diagram.**

1.7.9 Importing SharePoint Designer workflows into Visual Studio

A stunning new feature that comes with the 2010 version is the ability to export a SharePoint Designer workflow into Visual Studio (figure 1.18).

You would typically first create a workflow in SharePoint Designer because it's such an easy tool to use. After a year, you might realize that your business requirements have become more complicated, necessitating a Visual Studio workflow. In SharePoint 2007, you would have had to recreate the SharePoint Designer workflow from scratch within Visual Studio. Now, thanks to the new export and import functionality, you won't lose the valuable man-hours it took to build the Designer workflow in the first place.

1.7.10 Visual Studio 2010 environment improvements

Clearly, building custom workflows within Visual Studio has become dramatically easier with the 2010 releases of SharePoint and Visual Studio. Many would agree that most of the effort to build Visual Studio 2008 workflows in SharePoint 2007 was for the packaging and deployment of the workflow itself. You had to build all of the features, Diamond Directive Files (DDFs), manifests, keys, tokens, globally unique identifiers (GUIDs), and everything else by hand! This is no longer true with Visual Studio 2010—all of the necessary features and solution packages necessary to deploy a workflow into SharePoint are generated for you automatically. Right-click and deploy! For the steps in detail, see chapter 8.

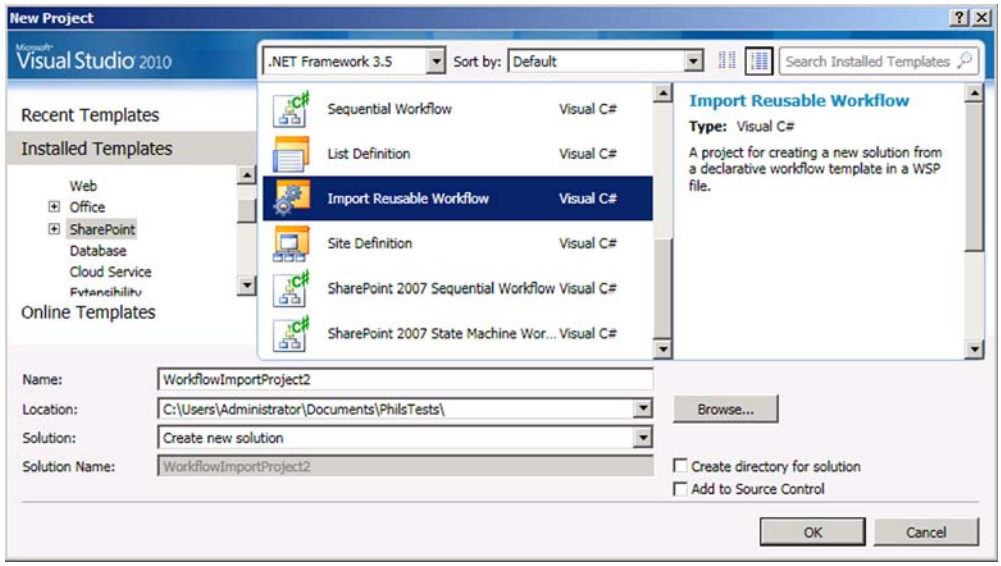

Figure 1.18 **Visual Studio 2010 comes with a new ability to import a workflow created in SharePoint Designer.**

1.7.11 Pluggable workflows

SharePoint 2007 offered no easy way of running workflows to receive updates from the outside world such as a payroll system, for example. With the new pluggable capabilities, SharePoint 2010 workflows can execute up to a certain point and then wait for information from an external process. The developer needs to implement an event handler or web service to handle the request of the external process and then respond by calling into a method within the workflow itself, informing it to continue processing. For more on this topic, see chapter 12.

1.7.12 New event handlers

SharePoint 2010 supplies a couple of new event handlers, including the ability to react to a workflow's initialization or completion. These handlers may be external to a workflow or embedded within a workflow. For instance, you may want to fire some code that logs in a centralized repository every time a workflow of a certain type has been started and completed. With the new event handlers, this is easy. See chapter 12 for more information about event handlers.

1.8 Building custom workflow solutions

Before you jump into the later chapters and start building workflows, it's important to consider how a workflow is born and progresses to completion. We've already discussed the many tools you can use to build custom workflows, but you could also use a careful comparison of these tools and a discussion of workflow diagramming. In this section, you'll diagram, design, scope, and choose the authoring tools for a generic business process.

1.8.1 Diagramming business processes

When gathering requirements for any software application, you usually work from the top down. First, you determine the high-level requirements, and then you get more and more detailed, as necessary. Working with SharePoint workflows is no exception. You first model the high-level business process. In effect, determine what you need to build. After you've received the approval from business stakeholders for *what* you're going to build, you can consider *how* you're going to build it.

To start diagramming, open Visio 2010. You'll notice a new diagram template called SharePoint Workflow. You may feel compelled to start with this template because, after all, you *are* building a SharePoint workflow. The problem with this template is that it's easy to get into the *how* prematurely. Many shapes in the template assume a fair amount of SharePoint knowledge. Take the Create List Item, Send Document Set to Repository, and Wait for Field Change shapes. For a SharePoint person, this may not be confusing. However, you probably wouldn't want to use that diagram in front of nontechnical stakeholders.

Instead of the SharePoint Workflow template, start with a standard flowchart template. With the flowchart template, focus on the *what* to specifically meet the business

need. With the first diagram you should focus on identifying the various states the workflow may use. A state defines the time a workflow waits for something to happen before proceeding to the next step. Figure 1.19 shows a flowchart model of a shopping cart workflow for an internet business that sells golf equipment. At a high level, there are only four states in this business process:

1. *Pending payment*—When an order is submitted, the workflow waits for the payment to be processed.
2. *Pending manufacturing*—In this example, a buyer may order golf clubs fitted to a custom length, which causes the workflow to wait for the manufacturing department.
3. *Pending shipping*—Before the order is fulfilled, the workflow must wait for it to be shipped. This may involve the assembling, packaging, and delivering of the ordered equipment to a shipper.
4. *Fulfilled*—After the order is shipped, the order is complete.

The corresponding workflow will certainly be more complex than this diagram shows, but it's a good start. Someone viewing this diagram can quickly see what the workflow will do. With this high-level diagram accepted by business stakeholders, you can move to a lower level.

Taking it to a lower level doesn't mean going to a SharePoint Workflow template in Visio just yet. First, let's expand on each of the four states. Figure 1.20 shows how you would drill down through the Pending payment state.

The Pending payment state first checks the payment type. If payment is by credit card, the card is processed. If it's by check, the workflow waits for the check to arrive in the mail and clear the bank. In both the credit card or the check methods, if the payment clears the Pending payment state, the workflow completes. Otherwise, a different workflow that handles bad payments is started, and the current workflow terminates.

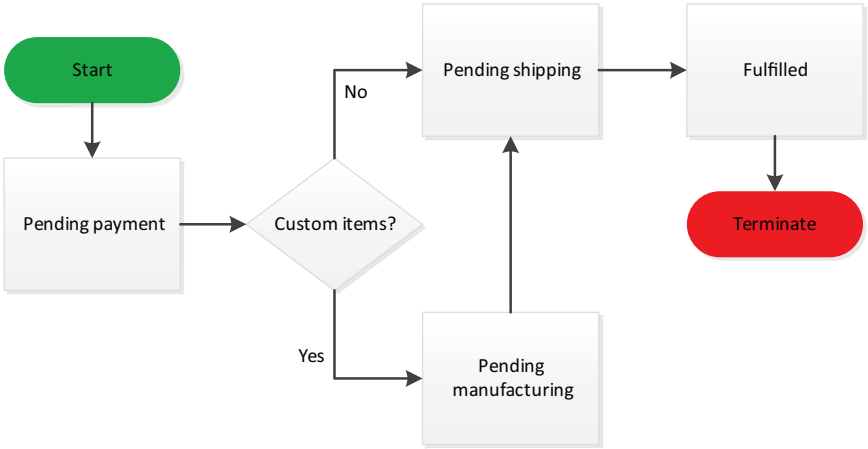

Figure 1.19 Start molding your business process by showing how the workflow will move from one state to another. Agree on the high-level requirements before getting too technical.

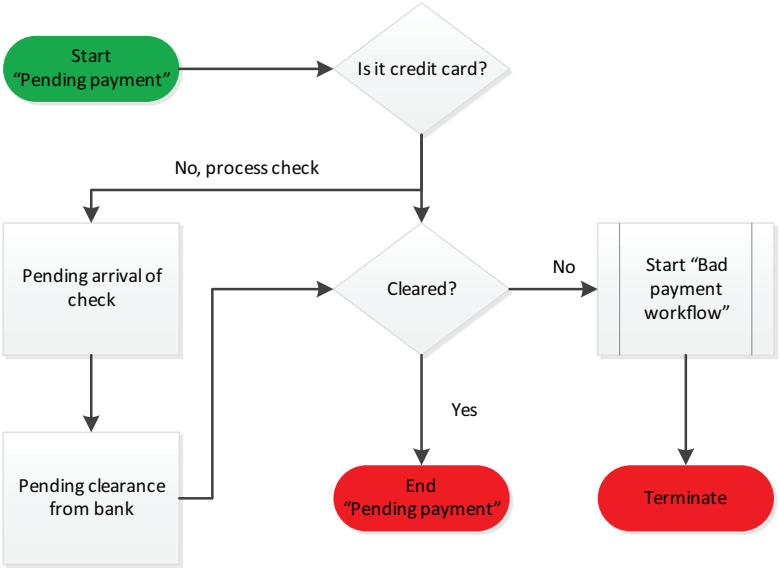

Figure 1.20 **Take your high-level model and drill down through each of the states by creating a new diagram for each state. This will provide valuable detail but remain nontechnical for easy consumption.**

Figure 1.20 is still fairly high level. It would be easy to get into the weeds of *how* to do this but, again, this early in the process it's more important to focus on *what*. Create diagrams for the other three states and run the flowchart past stakeholders. If, at this point you get a green light to go forward, you're probably ready to start using the SharePoint workflow template.

1.8.2 Identifying human interaction and SharePoint objects

Now that you know what the high-level business process is, it's time to turn your attention to how you're going to implement it. The first step is to take the flowcharts you created and determine where you'd expect human interaction. Examples of human interaction could be uploading a document or perhaps submitting a form with data. You also must consider what SharePoint objects your workflow will touch. In the case of the golf equipment e-business, the workflow starts when a buyer submits their order. That order could be logged into a SharePoint list, and your workflow would automatically fire when a new item is added.

From a human interaction perspective, let's consider the Pending payment state again. Notice that, if the payment is in the check form, the workflow waits for the check to arrive in the mail and clear the bank. Both of these points are examples where human interaction can help the workflow progress. For instance, mail department personnel who log into SharePoint after they receive checks could tell the workflow that the check for which the workflow is waiting has arrived. Similarly, an Accounts Receivable employee may monitor bank deposits and notify the workflow when the check clears.

One way to accomplish this is through tasks. Before the workflow waits for the check to arrive, it could create a task assigned to an individual in the mail department. When the mail clerk processes the check, he would edit this task in SharePoint, informing the workflow that the task has been received. You could similarly assign a task to someone in the Accounts Receivable department. That person could edit the task when they see the check has cleared.

Both of these examples of human interaction illustrate a workflow waiting for an event to occur before proceeding. You'll need to think through these points of interaction and how to inform the workflow to proceed.

This is where the SharePoint diagram template in Visio Premium can help. This template definitely lends itself to designing the workflow. With the task example, there's a shape called Assign a To-do Item. By dropping this shape onto the diagram, you specify that the workflow will create a task assigned to someone. This is shown in figure 1.21, where our original high-level flowchart has been created with the SharePoint workflow template.

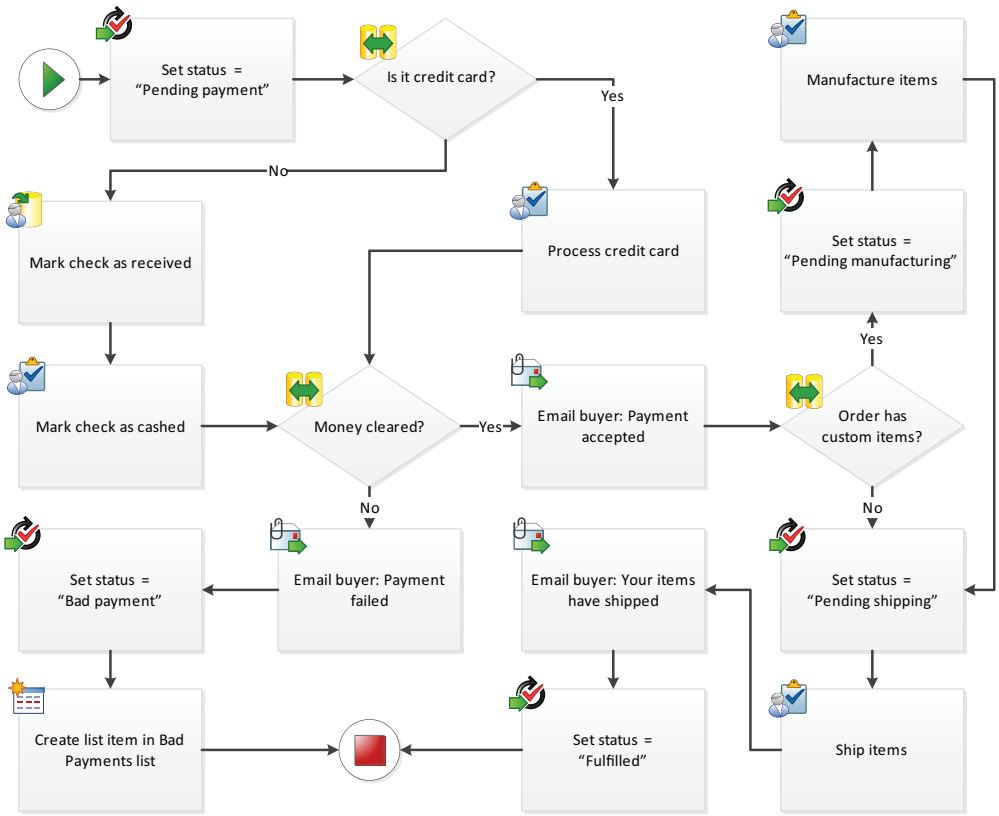

Figure 1.21 **The SharePoint Workflow template in Visio produces detail at a much lower level than the standard flowchart diagram. The advantage is the design is more clearly seen, but the disadvantage is it may be harder for nontechnical people to read.**

You'll notice the two boxes for check processing have been replaced with the Collect Data From a User shape and the Assign a To-do Item shape. You'll also notice the diagram is considerably more granular. Notice the shapes signifying that an email is sent, which you can see if the payment was bad. First, the workflow sends an email to the buyer when they submit a bad payment. Then the Create List Item follows, signifying that the workflow will create a new list item in a separate list. This, in effect, starts the Bad payment workflow, which is defined elsewhere. After the secondary workflow is started, the current workflow sets its status to Bad payment and then terminates. The other three states (Pending manufacturing, Pending shipment, and Fulfilled) have slightly been developed further.

What type of person uses the SharePoint Workflow Visio template?

Each shape in the Visio workflow template corresponds to an action in a SharePoint Designer workflow. Because of this, the Visio diagram developer needs to know quite a bit about SharePoint Designer workflows to model the workflow using the Visio workflow template. That's why I recommend that nontechnical people use the flowchart template and those with more technical experience use the SharePoint workflow template. My recommendation is to let the nontechnical business analyst focus on the high-level business process and to let someone with experience building workflows do the design work in the SharePoint workflow template.

This section has been a brief introduction for chapter 6, "Custom Visio SharePoint workflows." For more detail on how to diagram workflows in Visio, take a look at chapter 6. In addition, one of the best features of workflows diagramed in the SharePoint workflow template is that the diagram can be imported into SharePoint Designer. This is a big timesaver because all of the actions and conditions are precreated for the workflow developer.

1.8.3 Determining the deployment scope

With the design of the workflow now at a fairly low level, it's time to turn our attention to the scope of the workflow. Scope encompasses a few different things. First, what does the workflow run on (for example, a site, a document, a list item, or a content type)? Secondly, if the workflow needs to be reusable, where is it made available (site, site collection, or farm)?

For our golf equipment example, it's easy to see that the workflow runs on top of a list item. This list itself may be called Orders, and every order (list item) has its own workflow instance. Alternatively, a workflow could run on a document, a content type, or on the site itself. That latter is called a site workflow, where the workflow runs on a site, not an item.

Reusable means that you want users to be able to add the workflow onto multiple lists or sites. For example, all the out of the box workflows are reusable; they can be

used more than once. Our golf e-business example would not need to be reusable because it meets a one-time business need. If it needed to be reusable, you'd decide which scope the workflow should be deployed at. This involves four possible options: site, site collection, web application, and farm.

Workflows targeted to either site or site collection can be created with either SharePoint Designer or Visual Studio. In SharePoint Designer, you would create a *reusable workflow* to make a workflow available across a single site. If you want that workflow to be available across an entire site collection (multiple sites), you'd promote the workflow to be a *globally reusable workflow*. Alternatively, if you want the workflow to be available across an entire web application or the entire farm, you'll need a Visual Studio workflow.

1.8.4 Choosing appropriate workflow authoring tools

There are a few tools you can use to build custom workflows, but SharePoint Designer stands out for many reasons including its ease of use and speed. SharePoint Designer has some downsides where tools such as Visual Studio are needed to fill the gap, and it's important to know which tool is right for the job.

SHAREPOINT DESIGNER

SharePoint Designer is the first of the two authoring options. SharePoint Designer workflows are what's called *declarative*, meaning the workflow is based on Extensible Object Markup Language (XOML), rather than compiled code (as is the case with Visual Studio workflows). The following are six compelling reasons to use SharePoint Designer for your custom workflows:

- *It's easy to learn*—The language and terminology within the workflow components is easy to understand because it's written in plain English. Wizards and familiar interfaces are also used. For example, when composing a workflow-based email message, the dialog box is structured to look like an email message in an email client.
- *It's fast*—Writing software, which is essentially what a workflow is, typically requires a complex programming language such as .NET or C#. A workflow that might take all day to develop with .NET could be written in a few minutes with SharePoint Designer (although there are many limitations when compared to .NET workflows). This speed is due in large part to SharePoint Designer's point and click interface.
- *It's more powerful than previous versions*—SharePoint Designer 2007 allowed users to create workflows, but the new version includes many more features and has refined the development process significantly. It's now easier to work with and move workflow steps. Also, there are many more actions available than before, such as the utility workflows that allow you to manipulate text strings. It's also possible to access the users' profile data, such as their phone numbers and the managers' names through the workflow. Also, some complex tools have been

introduced, such as parallel blocks, which allow multiple activities to happen simultaneously. All of these concepts and many others make SharePoint Designer 2010 more powerful than previous versions.

- *Its workflows can be moderately complex*—It's easy to configure your workflows to make decisions using components called conditions. Large selections of actions allow the workflows to modify data and send notifications to users.
- *It doesn't require Visual Studio .NET coding or expertise*—As powerful as Visual Studio .NET is, it takes time to become proficient with it. SharePoint Designer 2010 requires no code but still allows for the user of common programming tools such as variables, which store data temporarily, and functions (in this case, called actions), which perform a variety of tasks. The learning curve for SharePoint Designer is much shorter than for its Visual Studio counterpart.
- *It uses a familiar interface*—SharePoint Designer uses the same interface as the rest of Office 2010, allowing business users to adopt it more quickly. In addition, the hyperlink-based configuration process will be accessible to anyone familiar with web browsing.

Anyone who needs to create SharePoint workflows will find value in SharePoint Designer 2010. In the past, business people did not have a role in the development of new software functionality. SharePoint and SharePoint Designer have changed that, allowing people who do not write code to create functionality that can be shared with other users.

.NET developers will be impressed with how quickly a complex workflow can be created compared to starting from scratch with a code-based solution. They may even find that creating a new workflow in Designer initially, then using Visual Studio .NET to add more functionality yields results faster than using Visual Studio .NET alone. This process is aided by the fact that workflows created with SharePoint Designer can be imported directly into Visual Studio.

Most businesses have a group of people who know the businesses processes well but are not developers by trade. These users will appreciate how easy it is to translate existing processes into SharePoint workflows using SharePoint Designer.

Finally, SharePoint administrators and power users will likely find the most value from SharePoint Designer 2010 workflows. This is because they are often asked to quickly translate business requirements into SharePoint solutions.

VISUAL STUDIO WORKFLOWS

When you realize you need a workflow, and you're considering using SharePoint Designer because of its apparent ease of use, it's important to stop and compare the process with Visual Studio workflows. There are some key pros and cons that ought to be considered before moving forward with one or the other. Visual Studio workflows are covered in detail in chapter 8.

The foremost thing to consider is how technically skilled the developer of the workflow is. SharePoint Designer is primarily for power users and site administrators. Visual

Studio is primarily for .NET developers with a programming background. If you don't have any resources available that have a .NET programming background, it's easy to conclude that a SharePoint Designer workflow is your best choice. Be aware that SharePoint Designer comes with some out-of-the-box functionality and, although it's rather extensive, it may not be able to accommodate the most complicated of business requirements. This may necessitate going outside of your organization for a .NET developer.

Beyond these high-level deciding factors, there are several other comparisons that should be made between SharePoint Designer and Visual Studio workflows. Table 1.1 highlights the main differences between the two tools. The ordering of the table brings to attention the most common deal breakers that necessitate Visual Studio or SharePoint Designer at the top, followed by differences that are not quite as critical.

Table 1.1 Comparison of the main differences between SharePoint Designer and Visual Studio workflows

SharePoint Designer	Visual Studio
Code-free development (limited to safe, predeployed activities).	Code-centric development (anything goes).
Sequential workflows only.	Supports both sequential and state machines.
Deployable across a site collection but not beyond.	Deployable across entire farm via a feature.
Automatically deployed into SharePoint.	Must be packaged within a feature and deployed by a farm administrator or in a sandbox.
Visio can be used to model workflow logic.	No support for Visio.
No debugging available to step through a workflow at runtime.	Full debugging experience available.
Intuitive support for forms customization.	Less intuitive InfoPath integration. Availability for ASP.NET custom forms.
Workflows cannot be modified at runtime.	Workflows can be modified while running (see chapter 9).
Compiled just-in-time.	Compiled at design time.

The first differentiator was already mentioned—SharePoint Designer is a code-free environment, whereas Visual Studio is a code-centric environment intended for programmers. This has obvious implications.

The second differentiator is that SharePoint Designer supports only sequential workflows. This can be a problem because many business processes don't take place sequentially but go from state to state without any foreknowledge. For instance, consider a bug tracking system for a computer program. A bug may be placed into a *Pending* state, waiting for a developer to start working on the bug. That developer starts working on the bug and fixes it, putting the bug into a *Fixed* state. Thereafter, a tester confirms that the bug was fixed and finds that it was not and must place the bug back

in a *Pending* state. This ability to go back in time or to a previous state is only available with state machine workflows and not present with sequential workflows—ruling out SharePoint Designer as a viable option.

Although several other differentiators are mentioned in the table, the third in the list is the last of the most common deal breakers, necessitating Visual Studio. SharePoint Designer supports deploying workflows globally. This global deployment means deploying the workflow across one particular site collection. You may have a requirement that a workflow can be instantiated from anywhere, on every site, across the entire farm. This true global deployment is only achievable with a Visual Studio workflow deployed via a feature. You will find more on Visual Studio and features in chapter 8.

WHAT'S RIGHT FOR THE GOLF EXAMPLE?

The golf equipment online sales workflow example lends itself to a Visual Studio workflow. The most apparent reason is the bad payment requirement. If someone makes an order online, and they type their credit card number incorrectly, you don't want the workflow to stop. The example shows it starts a new workflow instance, but wouldn't it be better if the workflow set its state back to pending payment and requested from the buyer resubmission of information? This would require a state machine to go back in time. Because you can build state machines only in Visual Studio, it remains the best option. If starting and terminating a new workflow is acceptable, you could still use SharePoint Designer.

1.9 Real-world examples

The examples found throughout this book can easily be applied to real-world business needs. Rather than make you search through the pages to find all of the examples, this section will serve as your table of contents for the examples you can take to your company. Table 1.2 briefly describes each example and what it does.

Table 1.2 Examples in this book

Chapter	Example name	Description
1	Purchase order and fulfillment workflow	At the end of chapter 1. This workflow is conceptually architected from start to finish. You'll get a good idea of the decisions and techniques used to plan and design your own custom business process.
2	Requirements document workflow	This simple example uses the Three-state workflow to compile and get the approval for a requirements document for a software application.
3	PTO request workflow	This SharePoint Designer workflow allows for a user to submit a request for paid time off (PTO) and get the manager's approval for the request.
4	Capital expenditure request workflow	We build a form that allows users to submit a capital expenditure request when they need funds for a new company initiative. A custom task process is used to get the approval for the request.

Table 1.2 Examples in this book *(continued)*

Chapter	Example name	Description
5	Document sets, security and expense reports, and sales order external data workflow	This chapter has three examples. The first includes a walk-through of how to send a document set containing sales presentation materials to a records center for retention. The second involves an expense report workflow that manages security. Users can upload their expense reports, and the SharePoint Designer workflow alters the security on the reports so only their managers can see and approve or reject the expenses. The final example is a SharePoint Designer workflow that tracks sales orders stored in an external SQL database. The workflow uses Business Connectivity Services (BCS) and external content types to connect to the external data.
6	Training request workflow	Visio 2010 is used to model a training request workflow. After the workflow is built in Visio, it's imported into SharePoint Designer and then published to SharePoint.
7	Expense report InfoPath form	InfoPath is used to create a custom expense report form in which users can enter and publish their expenses. This is a great primer for anyone new to InfoPath. Advanced InfoPath topics like creating dynamic dropdowns from SharePoint Data, customizing the out-of-the-box forms, as well as working with initiation forms in SharePoint Designer workflows are also covered in this chapter.
8	Maintenance order fulfillment workflow	A workflow is created with Visual Studio to manage the submission and fulfillment of a maintenance request for a college dormitory.
9	Service request workflow	A Visual Studio workflow is created for the fulfillment of service requests for a technical helpdesk. The example relies heavily on custom InfoPath and ASP.NET forms for the submission and fulfillment of the requests.
10	Capital expenditure request workflow	A workflow is created similarly to the one in chapter 4, except this time it's created in Visual Studio rather than SharePoint Designer. This is a good comparison between the two authoring tools.
11	Create custom subsite action and subsite exists custom condition	A custom Visual Studio workflow activity and condition is created and later published to SharePoint Designer for power end users. The action provisions subsites from within SharePoint Designer workflows.
12	Pluggable workflow	A local service workflow is created to send and receive messages from the outside world. The example shows an event handler communicating with a running workflow instance.

1.10 Summary

SharePoint workflows are excellent tools for automating, tracking, and organizing your company's business processes. Automating many of your most common business processes including expense reporting systems, paid time-off requests, and capital

expenditure requests is easily feasible within SharePoint, and you can see how this automation can substantially benefit your organization.

SharePoint workflows can execute on list items, documents, forms, content types, and even across an entire site or site collection within SharePoint. The boundaries of this technology and platform are almost limitless. Without much effort, you can put to use several compelling out-of-the-box SharePoint workflows and introduce immediate value into your SharePoint sites. If those workflows don't meet your business's unique needs, there's a host of workflow customization tools available like Visio diagramming, SharePoint Designer, InfoPath forms, and Visual Studio.

The 2010 release of SharePoint has introduced a slew of new functionality that greatly improves how workflows are architected in SharePoint. Tools like Visio will allow a nontechnical business analyst to model a workflow and hand that diagram to a SharePoint designer or programmer to import and use—saving time, energy, and miscommunication problems. Another useful improvement in 2010 is the ability to create reusable workflows. With SharePoint Designer 2007, you had to recreate a workflow on every list in the entire farm where you needed that workflow to execute. This was inefficient and costly. With the 2010 version, you can publish workflows globally, which saves time and drives consistency.

Next, we'll take what you've seen at a high level in this chapter and walk through the details and specifics on how to set up your first out-of-the-box workflow. In later chapters, you'll see a detailed walkthrough of all the out-of-the-box workflows and how to implement them in your company. Much care will be given to describing specifically how to build custom workflows from scratch.

MICROSOFT SHAREPOINT

SharePoint 2010 Workflows IN ACTION

Phil Wicklund

You can use SharePoint 2010 workflows to transform a set of business processes into working SharePoint applications. For that task, a power user gets prepackaged workflows, wizards, and design tools, and a programmer benefits from Visual Studio to handle advanced workflow requirements.

SharePoint 2010 Workflows in Action is a hands-on guide for workflow application development in SharePoint. Power users are introduced to the simplicity of building and integrating workflows using SharePoint Designer, Visio, InfoPath, and Office. Developers will learn to build custom processes and use external data sources. They will learn about state machine workflows, ASP.NET forms, event handlers, and much more. This book requires no previous experience with workflow app development.

What's Inside

- Out-of-the-box and custom workflows
- How to integrate external data
- Advanced forms with InfoPath and ASP.NET
- External events with pluggable workflow services
- Custom workflow actions and conditions
- Model your business process in Visio

As a SharePoint consultant and trainer for RBA Consulting, **Phil Wicklund** has implemented countless workflows. He is a frequent speaker at SharePoint conferences and he blogs at www.philwicklund.com.

For online access to the author and a free ebook for owners of this book, go to manning.com/SharePoint2010WorkflowsinAction

Free eBook
SEE INSERT

"Covers all aspects of SharePoint workflows."
—Wayne Ewington, Microsoft

"Great for learning or reference."
—Raymond Mitchell
Inetium, Inc

"A must-have."
—Justin Kobel
KiZAN Technologies

"The go-to resource."
—Andrew Grothe
Triware Technologies Inc.

"Every SharePoint dev needs this book!"
—Nikander & Margriet Bruggeman
Lois & Clark IT Services

ISBN 13: 978-1-935182-71-9
ISBN 10: 1-935182-71-4
54499
9 781935 182719

MANNING $44.99 / Can $51.99 [INCLUDING eBOOK]