

用于创建 SharePoint 应用程序的技术（第一部分，共两部分）

目录

- 简介
- 了解 SharePoint 应用程序
- 将 SharePoint 开发与其他形式的开发进行比较
- 语言和平台
- SharePoint 应用程序开发术语
- SharePoint 应用程序的类别
- SharePoint 功能
- SharePoint 开发人员方案
- 结论

简介

您可使用多种语言、框架、对象模型和工具在 SharePoint 2010 开发平台上构建可扩展的分布式 Web 应用程序。不熟悉 SharePoint 的开发人员很难知道使用哪些方法来实现各种方案，甚至不知道可以实现哪些任务。本文包含作为开发平台的 SharePoint 的高级概述，将 SharePoint 开发与其他类型的开发进行比较，并列出了 *SharePoint 开发人员构建基块*（用来构建 SharePoint 应用程序的技术）。或许，对于作为 SharePoint 开发人员的您而言，最大的难题是了解您对于构建基块有哪些选择，以及选择满足您的架构和设计目标的正确选项。

从较高的层次上说，从多个不同的视角检查 SharePoint 开发会很有用，如下所示：

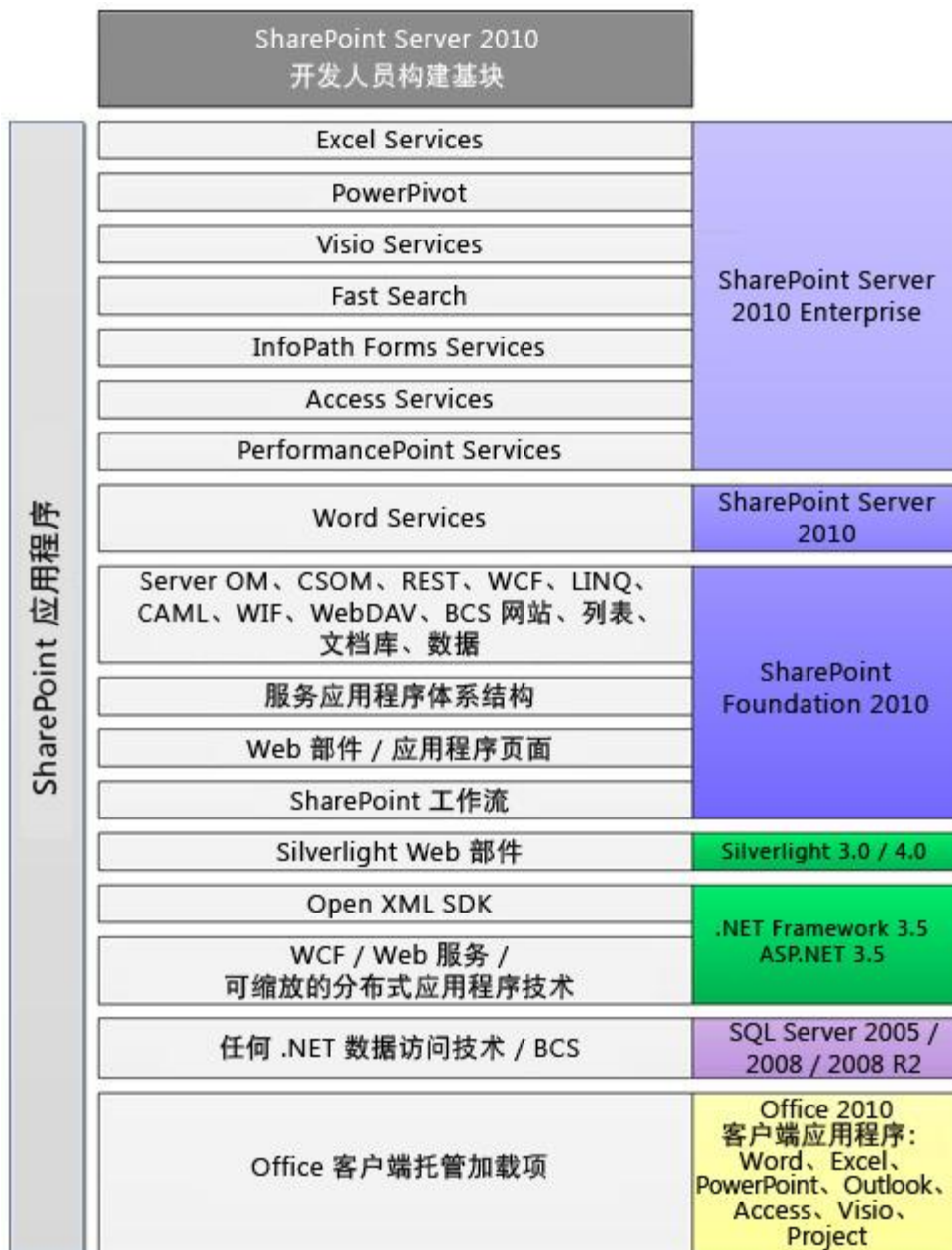
- 如何将 SharePoint 开发与其他形式的开发进行比较。
 - 可扩展的松散耦合 ASP.NET 应用程序开发
 - 数据库应用程序开发
 - 传统的富客户端开发
- 开发人员必须了解哪些语言和框架？

- 文档格式（如 Open XML）如何影响您的应用程序设计？存在哪些可能的方案？

本文假定已做出在 SharePoint 上进行开发的决定。这是大多数开发人员所处的位置。在某些情况下，会将开发人员指派给基于 SharePoint 构建的项目。在其他情况下，开发人员会积极开发其技术。

从较高的层次来说，SharePoint 2010 的开发人员体系结构包括多个技术层。它是一个基于 SQL Server 构建并在 Windows Server 2008 或 Windows Server 2008 R2 上运行的 Microsoft ASP.NET 应用程序。下图演示了多个更重要的构建基块如何使用基础系统来提供必要的功能。

图 1. 关键构建基块和基础系统的关系图



SharePoint 开发人员必须了解的技术的范围之广令人吃惊。我从事 SharePoint 开发时的起点与进入 SharePoint 开发领域的许多其他开发人员的起点是类似的。我是一个能干的开发人员，在某些领域有丰富经验，而在其他领域的经验较少并且只是大概了解许多其他领域。

我熟练掌握了以下技术：

- Microsoft Visual C++
- Microsoft Visual C#
- Microsoft .NET Framework
- LINQ
- SQL
- XML
- Open XML

我必须强化以下技术知识：

- ASP.NET
- ASP.NET 主题和皮肤
- HTML 和 CSS
- ECMAScript (JavaScript、JScript)
- ASP.NET Web 部件
- Internet Information Services (IIS)

尽管我了解体系结构，但作为一名开发人员，我需要了解以下技术的详细信息：

- SharePoint 对象模型
- SharePoint 托管客户端对象模型
- Microsoft Silverlight 和 XAML
- 工作流
- Windows Communication Foundation (WCF) 和 Web 服务
- SharePoint REST API

- Excel Services
- Windows Identity Foundation (WIF) 和基于声明的标识

我的观点是，即便我是一名资深开发人员，我也仍需面对一条明显的学习曲线。这些技术都需要我花一定的时间去学习。虽然基础概念不难，但需要经过一个较长的学习过程才能熟练掌握这些技术。

在本文中，我分享了有关 SharePoint 开发的各个方面的想法，并将 SharePoint 开发的各个方面与以前的经验联系起来。与我之前工作的开发环境有哪些相似之处？有哪些不同之处？我还讨论了对 SharePoint 开发的想法。SharePoint 应用程序由哪些部分构成？我在构建 SharePoint 应用程序方面有哪些选择？总体框架是怎样的，以及在我实现特定方案时应往哪些方面深入？我的目标是列出提高 SharePoint 开发效率所需了解的主要方面。

了解 SharePoint 应用程序

总体上说，SharePoint Server 是一个具有以下功能的系统：

- **网站。** 防火墙内部和外部的网站。
- **社区。** 通过 wiki、博客等进行的协作。
- **内容。** 企业内容管理、Web 内容管理和发布系统。
- **搜索。** SharePoint 产品和技术包括一个可根据您的需求自定义的搜索引擎。
- **洞察力。** 商业智能。
- **复合应用程序**包括数据访问、用户界面、过程和基础结构。

SharePoint 产品和技术提供了一个可扩展的体系结构，它适用于多个用户或一个有 100,000 名员工的大型企业。对于防火墙外部的网站，利用面向 Internet 站点的 SharePoint Server 2010，可以构建一个相当灵活的可扩展网站，通过维护 SharePoint 网站来维护该网站。

在大多数情况下，当您构建 SharePoint 应用程序时，会向核心功能添加其他功能。您可能需要简化某些重要用户的用户界面。您可能需要在面向某个特定垂直市场（例如，建筑，法律或制造）的多个 Web 部件中提供一些自定义功能。

在某些情况下，您可能会构建一个以编程方式使用 SharePoint 功能的独立应用程序。用户可能永远无法看到 SharePoint 用户界面，或无法了解到应用程序是通过将 SharePoint 用作其基础结构的一部分来开发的。

自定义的程度可低可高。您的整个自定义可以包含一个支持组或团队中某个方案的 Web 部件。或者，您可以构建一个包含数据流和进程的复杂商业应用程序。您可以与外部客户关系管理 (CRM) 或企业资源规划 (ERP) 系统进行集成（程度可低可高）。

SharePoint 与 Microsoft Office 2010 系统的集成很重要。可利用此集成按以下方式构建解决方案，不管用户是位于 Office 客户端应用程序还是 Web 浏览器中，都可以找到应用程序的功能，并轻松使用它们。我在本文中讨论了 SharePoint 开发与 Office 2010 客户端开发相交的位置。

将 SharePoint 开发与其他形式的开发进行比较

从三个视点检查 SharePoint 开发很有用：

- 为 .NET Framework 构建可扩展的应用程序
- 构建数据库应用程序
- 构建传统的富客户端应用程序

将 SharePoint 应用程序与可扩展的 .NET Web 应用程序进行比较

您可以从开发人员的角度检查 SharePoint 开发，该开发人员构建了在大型服务器场上运行的高度可扩展的松散耦合 Web 应用程序。这些应用程序每分钟必须处理数百个或数千个页面视图。

SharePoint 的核心基于 ASP.NET，并在 IIS 上运行，它可具有多个处理负载均衡的前端 Web 服务器。SQL Server 提供了 SharePoint 网站中存储的数据和文档的完整性、可伸缩性、可靠性和安全性。以下是可伸缩性影响 SharePoint 开发的一些重要方法。

- **API 设计。**可伸缩性将驱动 SharePoint 的编程接口的某些特征。当您了解编程接口的设计可提高可伸缩性时，就能更轻松的理解这些接口了。例如，[托管客户端对象模型](#)，抽象地说，该模型与 SharePoint Foundation 服务器端对象模型非常类似，但实际上它更为复杂，因为它使您能够在从服务器中检索数据或内容时明确进行控制。
- **解决方案设计。**可伸缩性将影响您设计基于 SharePoint 构建的解决方案的方式。您必须避免在服务器上导致不必要的计算或查询活动的设计。您必须编写资源消耗量不会多于应有资源消耗量的应用程序。例如，这意味着合理使用[协作应用程序标记语言核心架构](#)和 [LINQ to SharePoint](#) 来查询列表项。
- **最佳实践。**可伸缩性隐藏在作为 SharePoint 开发的[最佳实践](#)的某些编程方法和问题后面。例如，SharePoint 对象模型中的某些对象具有关联的非托管数据。因此，您必须了解并遵循对象处理规则。类似地，在使用 SharePoint 中的大型列表时，可考虑几个最佳实践。如果您不遵循这些规则，则可能会对服务器场产生负面影响。有关详细信息，请参阅[SharePoint Foundation 的最佳做法](#)和[SharePoint Server 的最佳做法](#)。另请参见[Best Practices: Using Disposable Windows SharePoint Services Objects](#)（该链接可能指向英文页面）和[释放对象](#)。您可使用自动化工具改进您的代码评审。有关详细信息，请参阅[使用 SPDisposeCheck 自动执行 SharePoint Dispose\(\) 代码评审](#)（该链接可能指向英文页面）。

有一些与构建高度可扩展的 Web 应用程序的开发人员所面临的问题相同或类似的问题。我听过这样一个情景，一个 SharePoint 开发人员编写代码以便按设定时间间隔循环访问其网站集中的所有文档

并收集要在树控件中显示的信息。这在其测试环境中能够正常工作。但是，代码设计会产生一个与文档和列表项的实际数量相关的性能问题。

可伸缩性通过两种不同的方式影响解决方案设计：

- 您必须构建可分发的应用程序，这些应用程序在部署到多个前端 Web 服务器上时可正常工作。例如，您可为在本地 XML 文件中存储数据的 Microsoft Business Connectivity Services (BCS) 构建小型 Create/Retrieve/Update/Delete Web 服务（请参阅 [Business Connectivity Services](#)）。但是，它在部署到负载平衡服务器场上时将无法正常工作。
- 您必须构建可正常执行的应用程序。例如，除非您确定某个列表将包含几个列表项，否则不要使用对象模型循环访问它；而是使用 LINQ to SharePoint，并为 SharePoint 提供优化机会。

有关 SharePoint 开发与 ASP.NET 开发的相似之处和不同之处的详细信息，请参阅 [针对 ASP.NET 开发人员的滑动路径](#)。另请参见 [ASP.NET 开发人员的 SharePoint 2010 开发](#)（该链接可能指向英文页面）。

[SharePoint Foundation 的最佳做法](#) 包含可帮助您避免对性能造成负面影响的缺陷的指南，其中包括有关对象处理、事件接收器、大型文件夹和列表以及代码性能优化的指导。

将 SharePoint 应用程序与数据库应用程序进行比较

数据库应用程序开发是用于查看 SharePoint 开发的有利位置之一。SharePoint 网站中的自定义列表与数据库表有很多相同之处。您可使用与列表中的列相关的丰富元数据来定义这些列。此外，SharePoint 列表可有效定义外键，以便您能对包含相关数据的更多有趣方案进行建模。SharePoint 将为删除操作提供级联和限制行为。您可以编程方式或声明方式创建这些列表，也可以编写使用用户定义的列表的程序。这些列表可以是可见的或隐藏的。您可使用 SharePoint 的安全功能来限制访问。

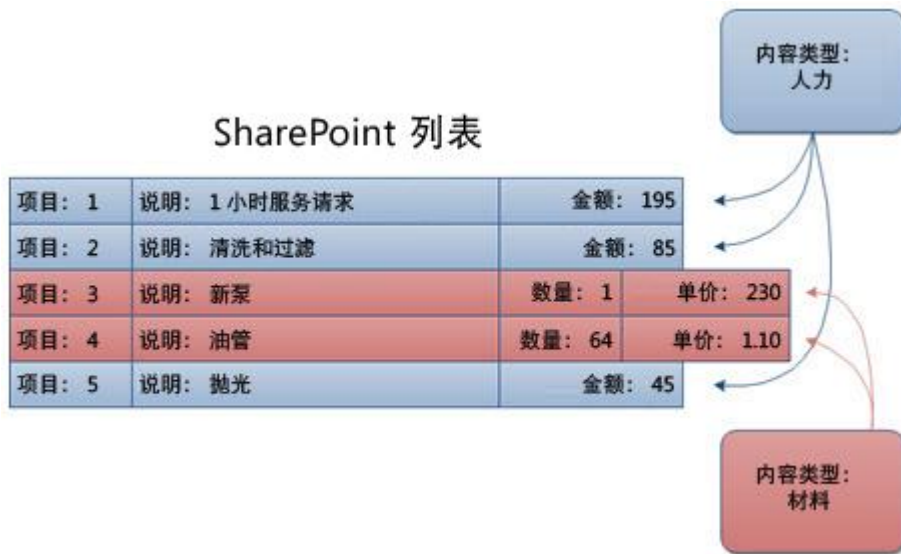
与 SQL 数据库形成直接对比的 SharePoint 的一个方面是，您使用非程序的声明性查询语言来检索数据。不过，将使用 LINQ to SharePoint 或使用通过 XML 编写的协作应用程序标记语言 (CAML)，而不使用 SQL。

SharePoint 与数据库技术的集成程度较深。您可通过 Business Connectivity Services 使用数据库、Web 服务以及几乎任何数据源。这些数据源将表示为外部内容类型。

SharePoint 的数据功能的一个有趣特征是，它不具有事务性保证。例如，您无法确保以下两个操作要么都发生，要么都不发生：在一个表中插入列表项，同时在另一个表中更新列表项。SharePoint 将不会用作实现事务性系统的平台。相反，应在可提供适当保证的外部数据库中实现此类事务性系统。然后，您可通过使用 [Business Connectivity Services](#) 在 SharePoint 中显示这些数据。当您设计 SharePoint 应用程序时，您必须考虑此特征。

SharePoint 的数据功能与传统数据库开发的数据功能之间的一个重大区别是，SharePoint 列表（可与数据库表进行比较）不一定是矩形。在 SharePoint 中，内容类型将定义构成列表中的列表项（可与行进行比较）的字段。可将其视为列表项的架构。SharePoint 列表可包含多个内容类型的列表项。下图表示一个包含两个内容类型的项的 SharePoint 列表：LABOR 和 MATERIAL。

图 2. 非矩形 SharePoint 列表



这将影响您设计和开发使用列表项以及列表项中的字段的应用程序的方式。如果您必须有矩形数据，则可定义列表，使其只能包含一个内容类型。如果您启用非矩形数据，则当您循环访问列表项时，您必须检查内容类型并相应更改代码的行为。

另一个有趣的特征是，由于存在内部实现详细信息，与使用 SQL 查询表相比，使用 LINQ to SharePoint 或 CAML 查询 SharePoint 的速度要慢得多。您不需要创建以下设计，其中的某些列表意外增长到 50,000 或 100,000 个项，并且用户可随意尝试在浏览器中的窗口中显示列表。SharePoint 2010 包含可阻止这类设计导致服务器场关闭的限制功能。但用户界面将会变得无法对用户作出响应。可以通过多种方式处理包含大量项的列表。您必须有意地处理此问题。有关详细信息，请参阅[处理大型文件夹和列表](#)。

SharePoint 基于 SQL Server 构建。文档库和 SharePoint 列表存储在数据库中。当您查看 SharePoint 安装的体系结构图时，您会发现这一点。实际 SharePoint 数据库与 SharePoint 开发人员之间并没有特别的关系。您从不直接访问此数据库。您总是使用编程接口来更改网站、列表和文档库。不过，您可使用相同的 SQL Server 安装来承载您直接使用或通过 Business Connectivity Services 使用的数据库。因此，它可以是您用来构建 SharePoint 应用程序的基础结构的一部分。

Microsoft Access 服务允许（但也有限制）您将 Access 数据库发布到 SharePoint 网站。这会带来一些很有用的机会，由于它允许 SharePoint 用户以一致且熟悉的形式共享数据。

客户端数据访问技术

您可以按编程方式从客户端计算机访问 SharePoint 数据。在从客户端计算机访问数据时，SharePoint 将使用网络技术向 SharePoint Foundation 或 SharePoint Server 发送请求，并从 SharePoint 中检索请求的结果。

客户端数据访问技术

数据访问技术	说明
Business Connectivity Services 客户端 API	允许对外部内容类型中的数据进行联机访问或脱机访问。您可从 Microsoft Word、Microsoft Excel 或 Microsoft PowerPoint 的托管加载项访问此对象模型。这有时是访问应用程序中的 SharePoint 外部数据的最有力且最方便的方法。
Web 服务	启用高性能设计，在此设计中，可在 SharePoint Server 上承载自定义 WCF 服务。WCF 服务可通过服务器对象模型访问数据和文档，并允许客户端通过您实现的服务访问数据和文档。
Excel Services ECMAScript 接口	允许您在页面上自定义、自动化和驱动 Excel Web Access Web 部件控件
Excel Services REST API	允许您通过 URL 直接访问工作簿部件或元素。
Excel Web Services	允许应用程序计算、设置和提取工作簿中的值，以及刷新外部数据连接。
ECMAScript 客户端对象模型	允许 ECMAScript 程序访问 SharePoint 数据。此编程接口是托管客户端对象模型的变体。
托管客户端对象模型	允许 .NET Framework 应用程序（包括独立应用程序、Silverlight 应用程序和 ASP.NET 应用程序）访问 SharePoint 数据。此框架类似于服务器对象模型，它包含允许您控制服务器中的网络通信的其他语义。在使用客户端对象模型时，了解并使用 CAML 很重要。
OData REST API	提供针对列表和列表项的 RESTful 编程接口。
SharePoint Web Services	提供您用来远程处理 SharePoint Foundation 部署的 Web 服务方法。
基于 Web 的分布式创作和版本管理 (WebDAV)	包含一组方法、标头和内容类型，它们扩展 HTTP/1.1 以允许创建和管理资源集合。利用针对 WebDAV 的 SharePoint 支持，可以上载和下载文档。WebDAV (RFC 4918 (该链接可能指向英文页面)) 是 Internet 工程任务组

	(IETF) 的工作文档。
--	---------------

服务器端数据访问技术

当编写直接在服务器上运行的代码时，不管您是构建一个沙盒解决方案还是服务器服务器场解决方案，您可在 CAML 中指定数据查询或使用 LINQ to SharePoint 指定数据查询。CAML 将返回弱类型 `SPListItem` 对象，而 LINQ to SharePoint 将返回在生成的代码中定义的强类型对象。

服务器端数据访问技术

数据访问技术	说明
协作应用程序标记语言 (CAML)	基于 XML 的非程序性查询语言
LINQ to SharePoint	语言集成查询

除了使用 CAML 或 LINQ to SharePoint 查询一组项目之外，您还可以使用服务器对象模型直接循环访问列表或库中的项目。不过，除非您必须处理列表中的所有项，并且您知道列表中所包含的项目数，否则您不应这样做。我通常会将应用程序设计为数据驱动的。我会设计一个列表来包含驱动行为的特定数据。我可能会这样设计，以便我始终知道有一定数量的记录，并且我必须处理所有记录以驱动我的应用程序行为。在此情况下，我会直接循环访问项，而不使用 CAML 或 LINQ to SharePoint 进行查询。请注意，当您使用托管客户端对象模型时，此方法无效。一旦使用托管客户端对象模型，您就必须在检索列表项集合时提供 CAML 查询。

将 SharePoint 应用程序与传统的富客户端应用程序进行比较

SharePoint 与操作系统有很多共同之处：

- SharePoint 包含存储。包含文件夹的层次结构的文档库与文件系统中目录中的文件很类似。
- SharePoint 具有可编程的用户界面。
- 在与操作系统一起使用时，您可编写以不同级别的特权运行的代码。通过使用 SharePoint 开发，您可编写必须使用服务器场管理凭据运行的代码。您可编写在沙盒解决方案中运行的代码，也可编写在客户端浏览器中运行的 ECMAScript。
- 在与操作系统一起使用时，您可编写服务，这些服务为构建具有复杂动态的 SharePoint 应用程序提供必需的基础结构。

再次说明一下，规模是最大的差异。规模将驱动 SharePoint 的可编程性特征。您可编写在数百台服务器上以高性能运行的软件，以便为企业提供一个可提高协作和工作效率的一致体验。当您从传统应用程序开发转向 SharePoint 开发时，处理可伸缩性问题是您要付出的准入代价。

语言和平台

所有新 SharePoint 开发人员要考虑的第一个问题是：*我需要掌握多少知识？* 在进行 SharePoint 开发时，列出您必须了解的语言和平台很有意义。下表中的第一个项最为重要。除此之外，它取决于您所执行的操作。您并不需要掌握下表中的每项技术。您只需了解适用于您的当前任务的技术。

- **Microsoft Visual C#、Microsoft Visual Basic 和 Microsoft .NET Framework。** 在为服务器场、沙盒网站集级别解决方案构建完全信任的应用程序或在 Microsoft Silverlight 应用程序内构建完全信任的应用程序时，使用一种形式或另一种形式的 .NET Framework。如果您了解各种 LINQ 技术（LINQ to XML、LINQ to Entities、LINQ to SharePoint 以及 LINQ to SQL），则您将成为一名工作效率更高的开发人员。
- **ASP.NET。** SharePoint 基于 ASP.NET 并在 IIS 中承载。它使用了母版页和应用程序页，这使您能够标准化页面外观和行为。它还使用了 Web 部件页和 Web 部件，这使用户能够配置自己的页面。您可以将 Web 部件相互连接，以便能创建父/子数据编辑功能或混合 Web 应用程序。
- **HTML/CSS。** SharePoint 开发包含多个方面，它们对编写和了解 HTML 和 CSS 很有用。
- **ECMAScript 和 HTML 文档对象模型 (DOM)。** ECMAScript 是工具箱中的一个功能强大的工具。您可使用 ECMAScript 客户端对象模型或 ECMAScript 中的 OData REST API，从您的网站中获取数据和文档。可利用 JQuery 和 AJAX 添加丰富的交互性。
- **Silverlight/XAML。** Silverlight 是一个开发平台，用于为 Web、桌面和移动设备内的多个屏幕创建引人注目的交互性应用程序。您可使用 Silverlight 代码内的托管客户端对象模型来读取和写入网站数据和文档。Microsoft Expression Blend 是一个有效工具，用于为 SharePoint 设计 Silverlight Web 部件。
- **XML。** 您必须了解如何对有效 XML 文档进行编码。有关使用 XML 进行开发的信息，我个人的建议是，了解并掌握 LINQ to XML。使用 XML 是一种更有效的方法。了解 XSLT 会很有用。您可手动编写自定义 XSLT 列表视图，尽管 XSLT 列表视图的主要用途是，允许您通过使用 Microsoft SharePoint Designer 将有趣的格式应用于列表。换句话说，XSLT 通常用作开发工具的输出，而不是用作常用开发人员工具。

SharePoint 应用程序开发术语

为了在谈论 SharePoint 应用程序开发时更方便，我定义了四个术语（来源于 SharePoint 营销网站），以便我在剩下的文章内容中讨论它们时会更容易。

SharePoint 应用程序。 由多个自定义构成，以满足市场或客户需求。一个应用程序包含自定义 Web 部件、工作流、事件处理程序、内容类型、预定义的列表或您用来增强 SharePoint 功能的任何其他技术的一些组合。

SharePoint 功能。SharePoint 功能的关键区域。开发人员基于这些功能区域进行构建以将解决方案结合在一起，以实现比各个部件简单相加更好的效果。这些功能包括：

- [网站](#)（该链接可能指向英文页面）
- [社区](#)（该链接可能指向英文页面）
- [复合](#)（该链接可能指向英文页面）
- [内容](#)（该链接可能指向英文页面）
- [洞察力](#)（该链接可能指向英文页面）
- [搜索](#)（该链接可能指向英文页面）

SharePoint 开发人员方案。开发人员希望为 SharePoint 用户启用的活动。例如，您可能需要允许用户基于存储在 SharePoint 列表中的数据生成几百个 Word 文档，从这些文档中生成 PDF 或 XPS 文件，以及打印这些文件或通过电子邮件发送它们。为了完成此方案，您使用了多个开发人员来构建基块。可能的一种实现是，在作为沙盒解决方案实现的 Web 部件中使用 Open XML SDK 2.0 for Microsoft Office（请参阅[欢迎使用 Open XML SDK 2.0 for Microsoft Office](#)）。然后使用 Word 服务从 DOCX 转换为 PDF。

SharePoint 开发人员构建基块。您用来进行开发的 SharePoint 中的技术。例如，Web 部件、工作流、事件处理程序、数据和服务。当您构建应用程序时，将通过使用构建基块进行开发，以便所有部件可进行协作。

SharePoint 应用程序的类别

SharePoint 开发人员构建的应用程序有哪些类别？此问题与询问有关可在操作系统上构建的应用程序的类别相似。开发人员正在设定涉及多个方案的 SharePoint 用法。以下是应用程序的几个类别。

- **垂直市场**。供应商为多个市场（例如，房地产管理、度假胜地和娱乐管理或设备出租）自定义 SharePoint。
- **自定义的团队应用程序**。有时，自定义应用程序可帮助大型组织的各部门以一种组织性更强的可见方式实现其目标。测试组织可构建一个系统，来维护测试仪器和资源。
- **内容发布系统**。这些系统可以像由单个作者维护的网站或网站集一样简单。它们也可以像由数百名作者、开发人员和经理使用的技术文档生成系统一样复杂（例如，Office.com 网站）。
- **会计系统**。尽管 Microsoft 建议不要在 SharePoint 中实现事务性系统，将数据从外部会计系统引入 SharePoint 中可极大地增强用户体验。

- **管理信息系统。**一些自动化管理系统不具有与会计系统相同的事务性要求。您可以通过使用默认 SharePoint 安装来维护包含客户、产品、资产和资源的列表。在以编程方式进行扩展时，您会提供一种很高级的体验。
- **销售和营销。**市场和销售系统在很多方面都与管理信息系统类似。将 SharePoint 与 CRM 系统（例如，[Microsoft Dynamics](#)（该链接可能指向英文页面））集成会生成功能强大的混合应用程序。另外，对 SharePoint 网站进行移动访问可为出差人士提供适当的工具，供其用来与执行其工作所需的数据进行交互。
- **项目管理。** Microsoft Project Server 2010 与 SharePoint Server Enterprise 的集成将提供项目和项目组合管理，以帮助组织对投资进行排序、调整资源和执行项目。您可使用其 API 来开发 Project Server。

尽管 SharePoint 应用程序的许多示例都适合这些类别中的某个类别，但此列表并不完整。我确信，有人已编写了一个在 SharePoint 上运行的游戏，此游戏会利用协作功能。另外，我猜测有人已使用 SharePoint 列表、事件处理程序等构建了一个系统，该系统将控制计算机自动化制造系统。为其他垂直市场（例如，计算机辅助设计或体系结构）提供服务的供应商将自行对 SharePoint 进行各种改进。

可为 SharePoint 构建的应用程序的类别几乎会随可为 Windows 操作系统构建的应用程序的类别的变化而变化。

SharePoint 功能

可从 [Microsoft SharePoint 2010 产品概述](#)（该链接可能指向英文页面）中找到有关 SharePoint 功能的更多信息。这些功能包括：

- **网站。**可使用 SharePoint 2010 开发和管理所有商业网站。SharePoint 2010 提供了一整套工具，可供用户用来创建任意类型的网站。SharePoint 提供了简化网站管理的基础结构。
- **社区。** SharePoint 2010 提供了协作工具，可供任何人用来分享想法、查找人员和专家以及定位商业信息。默认情况下，SharePoint 具有很多协作功能，但通过有意义的方式扩展 SharePoint 可使协作由被动变为主动。通过使用 [Microsoft Lync Server 2010](#) 将状态集成到 SharePoint 应用程序中，使同事之间更易于联系。
- **内容。**从国际出版组织到负责发布产品文档的小型部门，发布组织以多种形式存在。这些组织可使用 SharePoint 2010 来管理内容并使用 Word 2010 来编辑内容。我还将内容功能分为以下三个子类别：
 - **发布系统。**您可使用 SharePoint Server 2010 来管理全面的发布系统。您可以用 PDF、XPS 或其他一些形式交付最终产品。Open XML 是一项功能特别强大的技术，您可使用它来构建发布系统。您可能必须将复杂的转换编写为其他格式。例如，请参阅 [Open XML 到其他形式（例如 XHTML）](#)（该链接可能指向英文页面）。

- **Web 内容管理。**不管是使用面向 Internet 站点的 SharePoint Server 2010 来构建面向外部的网站，还是使用 SharePoint 在防火墙内构建 wiki 和页面，用户都可利用 SharePoint 通过完成最小工作量构建丰富内容。
- **企业内容管理。**许多组织都具有大量*组织体验*，其形式通常为通过使用修订、审批过程和元数据管理的文档。
- **搜索。**自定义搜索体验对于用户使用信息至关重要。利用 SharePoint Server 2010，您可自定义以允许 SharePoint 搜索引擎在自定义文档中搜索。有关详细信息，请参阅 [SharePoint 企业级搜索](#)。此外，您可以自定义搜索，以允许它在外部内容类型 (ECT) 中进行搜索。
- **洞察力。**在最近的一项调查中，询问了一些大型公司的 CEO 他们“最关心”的问题。其中，商业智能 (BI) 被提及的频率最高。Excel Services 和用于构建仪表盘的功能对于构建强大而灵活的 BI 系统尤为重要。SharePoint 的 PerformancePoint Services 提供了用于交付可自定义的 BI 仪表盘的其他选项。
- **复合。**创建将客户用户界面、数据访问、业务逻辑和服务结合在一起的创新性解决方案。您可使用工作流来记录和标准化业务流程。

SharePoint 开发人员方案

许多开发人员在设计软件时（但在编写功能性规格之前）会确定并记录开发人员方案。有时，此过程包含*情节提要*的构建，该情节提要演示用户如何从 A 点移动到 B 点以执行特定任务。有时，您可以执行正式用例分析。您甚至可以使用 PowerPoint 或建模工具来模拟某个方案。在大多数开发组织中，您编写概述并详述方案的正式规范。

方案的类别数多于应用程序的类别数。但是，方案有几个常规类别。

- 提供针对现有 SharePoint 数据的更好的用户界面的方案。例如，您可能需要显示基于多个 SharePoint 列表中的数据的计算值。
- 拉入外部系统中的外部数据的方案。您可以连接到现代 ERP 或 CRM 系统，也可以连接到专有数据库，该数据库是通过使用过时的硬件体系结构的仿真器在 UNIX 上实现的。
- 以文档生成为中心的方案。您可以基于 SharePoint 列表或外部系统中存储的数据生成数百个或数千个 Open XML 文档，然后将这些文档转换为 PDF 或其他格式。
- 围绕进程和过程的方案。方案既可以像文档审批流程一样简单，也可以像从生成建议书到签署合同这样的整个监管 RFP 过程一样复杂。

通过使用一个或多个 SharePoint 开发人员构建基块来实现每个开发人员方案。当然，作为系统设计员，您必须确定您自己的方案。然后，您的任务是熟悉可用于实现您的方案的可用构建基块。在许多情况下，您具有多个选项；每个选项具有不同的特性，并且一个选项可能比另一个选项更适合于特定方案。在很大程度上，您可能已经知道您需要选取的方向。高端营销和设计网站或奢侈品网站可能从

一开始就打算使用 Silverlight（如果需要），而工作组网站的兼职开发人员可能对如何以最快且最轻松的方式完成任务感兴趣。

结论

在本文中，我探讨了如何将 SharePoint 开发与其他类别的应用程序开发进行比较，这包括可扩展的、松散耦合 ASP.NET 应用程序开发、数据库应用程序开发和传统的富客户端开发。我列出了您需要了解的各种语言和平台，并讨论了一些 SharePoint 开发人员术语。我探讨了 SharePoint 的功能，并讨论了开发人员如何扩展这些功能以满足其用户的需求。