

第 2 章 信息系统开发的理论基础

本章学习目标

信息系统的分析与设计到最后都是要开发出相应的软件系统，长期以来，系统开发人员总是希望开发软件能够像工厂生产产品或者是建房子一样，能够有一套规范和标准，并能够对软件的质量进行检验。经过几十年的摸索研究，目前的软件生产已经在软件的开发方面总结出一些行之有效的方法，那就是在对信息系统进行分析和设计的时候，要使用软件工程的思想和方法。软件开发技术、软件工程技术和设计方法受到了更多人的关注，软件工程对软件产业的形成和发展起着决定性的推动作用。通过本章学习应该达到以下学习目标：

- 掌握软件工程的概念和软件开发的方法
- 熟练掌握生命周期法
- 熟悉系统开发的工具
- 了解系统的开发模式

2.1 软件工程

2.1.1 软件工程概述

1. 软件

程序和软件是有区别的，是两个不同的概念。程序（program）是为了实现特定目标或解决特定的问题而使用相应的计算机语言编写的一系列指令的集合。软件（software）并不只是包括可以在计算机上运行的程序，还包括与这些程序相关的文档。简单地说，软件是计算机系统中与硬件相互依存的另一部分，包括程序、相关数据以及说明文档。

软件作为一种产品，具有如下特点：

（1）软件是一种逻辑产品，具有抽象性。人们可以把它记录在白纸、磁盘或者光盘等介质上，但却无法看到软件本身的形态，只有通过观察、分析、思考和判断才能了解软件的功能和性能。

（2）软件对硬件和环境有着不同程度的依赖性，这样就带来了软件升级和移植的问题。计算机硬件和支撑环境不断地升级，软件也要不断地进行维护以适用运行环境的变化，而维护的成本通常比开发成本高许多。

（3）软件作为一种产品，它的生产至今仍然采用的是手工方式，随着计算机技术的发展，对软件功能和性能的需求越来越高，软件的规模也越来越大，这对于软件开发人员的工作能力也提出了更高的要求。

（4）计算机现在已经被广泛地应用到社会生活和生产的各行各业，常常涉及一些行业的专业知识，这对于只熟悉计算机软件开发的软件工程师提出了更高的要求。并且每个行业有每

个行业的标准，而软件的生产却无法规定一个统一的标准。

(5) 由于软件是一个逻辑产品，在生产过程中存在着“不可见性”，所以这也大大增加了软件生产过程中组织管理的难度。

2. 软件危机

正因为软件具有上述这些特点，长期以来在软件的生产过程当中，一直缺乏一种高效的开发方法，导致了软件生产的效率低下，难以在预定时间内完成，最终交付的软件产品在质量上难以保证并且难以维护。这种现象在 20 世纪 60 年代被称为“软件危机”。它的具体表现如下：

- (1) 已完成的软件产品不能满足用户的需求；
- (2) 开发的进度难以保证，交付的时间一拖再拖；
- (3) 软件开发的成本难以准确地估算，开发过程控制的困难造成了开发成本超出了预算；
- (4) 软件产品的质量难以保证，屡屡出现运算结果不正确，频繁出错，甚至死机的情况；
- (5) 软件通常没有相应的文档资料，或者是文档与软件产品不符，造成软件难以维护。

虽然软件的特点是导致软件危机的客观因素，但在软件开发和维护过程中使用的方法不正确却是导致软件危机的主观因素。主要表现为：忽视软件开发前期的调研和分析工作，没有统一的、规范的方法，文档资料不齐全，忽视开发人员之间的交流沟通，忽视测试工作，轻视软件的维护。

3. 软件工程的定义

1968 年在德国举行的学术会议上，北大西洋公约组织正式提出了软件工程这一术语。它的目的是采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理方法和先进的软件开发技术结合起来，运用到软件开发和维护过程中，以解决软件危机。

1993 年 IEEE 在软件工程技术汇编中对软件工程是这样定义的：软件工程是：一是将系统化的、规范的、量化的方法应用于软件的开发、运行和维护过程，也就是说将工程化应用于软件的开发和管理之中；二是对一中所选方法的研究。

《计算机科学技术百科全书》中的定义：软件工程是应用计算机科学、数学及管理科学等原理，开发软件的工程。软件工程借鉴传统工程的原则、方法，以提高质量，降低成本。其中，计算机科学、数学用于构建模型与算法，工程科学用于制定规范、设计范型、评估成本及确定权衡，管理科学用于计划、资源、质量、成本等管理。

目前比较认可的一种定义认为：软件工程是研究和应用如何以系统性的、规范化的、可量化的过程化方法去开发和维护软件，以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

软件工程研究的主要内容是软件开发技术和软件开发管理两个方面。软件开发技术方面主要研究软件开发方法、软件开发过程、软件开发工具和开发环境。软件开发管理方面主要研究软件工程管理学、软件工程经济学、软件工程心理学。

2.1.2 软件工程的七条基本原理

自从 1968 年提出“软件工程”这一术语以来，研究软件工程的专家学者们陆续提出了 100 多条关于软件工程的准则或信条。美国著名的软件工程专家巴利·玻姆（Barry Boehm）综合这些专家的意见，并总结了美国天合公司（TRW）多年的开发软件的经验，于 1983 年提出了

软件工程的7条基本原理。玻姆认为,这7条原理是确保软件产品质量和开发效率的原理的最小集合。它们是相互独立的,是缺一不可的最小集合;同时,它们又是相当完备的。人们当然不能用数学方法严格证明它们是一个完备的集合,但是可以证明,在此之前已经提出的100多条软件工程准则都可以有这七条原理的任意组合蕴含或派生。下面简要介绍软件工程的7条原理。

(1) 用分阶段的生命周期计划严格管理。这一条是吸取前人的教训而提出来的。统计表明,50%以上的失败项目是由于计划不周而造成的。在软件开发与维护的漫长生命周期中,需要完成许多性质各异的工作。这条原理意味着,应该把软件生命周期分成若干阶段,并制定出切实可行的计划,然后严格按照计划对软件的开发和维护进行管理。玻姆认为,在整个软件生命周期中应指定并严格执行6类计划:项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行维护计划。

(2) 坚持进行阶段评审。统计结果显示:大部分错误是在编码之前造成的,大约占63%。错误发现得越晚,改正它要付出的代价就越大,要差2~3个数量级。因此,软件的质量保证工作不能等到编码结束之后再进行,应坚持进行严格的阶段评审,以便尽早发现错误。

(3) 实行严格的产品控制。开发人员最痛恨的事情之一就是改动需求。但是实践告诉我们,需求的改动往往是不可避免的。这就要求我们要采用科学的产品控制技术来顺应这种要求,也就是要采用变动控制,又叫基准配置管理。当需求变动时,其他各个阶段的文档或代码随之相应变动,以保证软件的一致性。

(4) 采纳现代程序设计技术。从20世纪六七十年代的结构化软件开发技术,到最近的面向对象技术,从第一、第二代语言,到第四代语言,人们已经充分认识到:好的方法比花费气力更重要。采用先进的技术既可以提高软件开发的效率,又可以减少软件维护的成本。

(5) 结果应能清楚地审查。软件是一种看不见、摸不着的逻辑产品。软件开发小组的工作进展情况可见性差,难于评价和管理。为更好地进行管理,应根据软件开发的总目标及完成期限,尽量明确地规定开发小组的责任和产品标准,从而使所得到的标准能清楚地审查。

(6) 开发小组的人员应少而精。开发人员的素质和数量是影响软件质量和开发效率的重要因素,应该少而精。这一条基于两点原因:高素质开发人员的效率比低素质开发人员的效率要高几倍到几十倍,开发工作中犯的错误也要少得多;当开发小组为N人时,可能的通信信道为 $N(N-1)/2$,可见随着人数N的增大,通信开销将急剧增大。

(7) 承认不断改进软件工程实践的必要性。遵从上述6条基本原理,就能够较好地实现软件的工程化生产。但是,它们只是对现有的经验的总结和归纳,并不能保证赶上技术不断前进发展的步伐。因此,玻姆提出应把承认不断改进软件工程实践的必要性作为软件工程的第7条原理。根据这条原理,不仅要积极采纳新的软件开发技术,还要注意不断总结经验,收集进度和消耗等数据,进行出错类型和问题报告统计。这些数据既可以用来评估新的软件技术的效果,也可以用来指明必须着重注意的问题和应该优先进行研究的工具和技术。

2.2 系统开发的方法

为了解决软件危机,软件工程的研究人员不断总结经验,探索新的软件开发方法。目前主要的开发方法有:生命周期法、原型法、面向对象的开发方法、计算机辅助设计开发方法等。

2.2.1 生命周期法

任何系统都有一个产生、发展、成熟、消亡（更新）的过程，整个过程称为系统的生命周期。信息系统也不例外，也存在着从提出开发的要求开始，然后进行规划、开发、使用，直到软件被淘汰抛弃的过程。所谓生命周期法就是严格按照系统生命周期的各个过程和步骤去开发系统。生命周期法的基本思想是将软件工程学和系统工程的理论和方法引入到计算机系统的开发中，按照用户至上的原则，采用结构化、模块化自顶向下对系统进行分析与设计。具体来说，它将整个信息系统开发过程划分为系统定义（问题定义、可行性研究、需求分析）、系统设计（总体设计、详细设计、编码和单元测试、综合测试）、系统实施与维护（系统实施、系统运行与维护）3个时期9个阶段。图2-1为实际的生命周期法模型，因为像瀑布一样，所以叫瀑布模型。

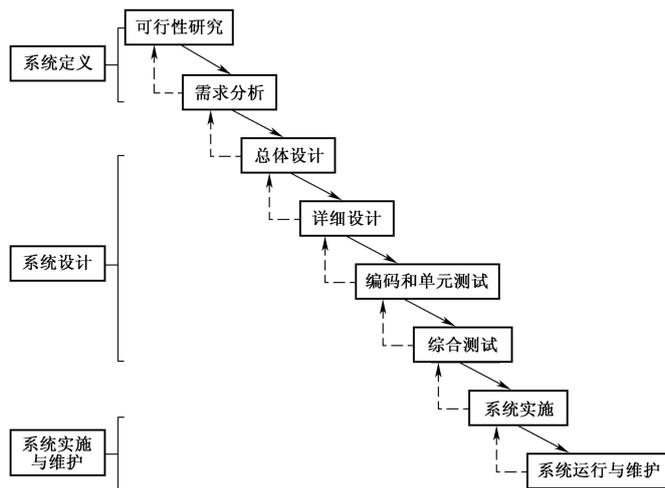


图 2-1 实际的瀑布模型

这9个阶段构成信息系统的生命周期。其中，主要阶段的任务如下：

(1) 可行性研究。可行性研究是在系统开发项目确定之前，对系统开发的必要性、可能性以及可能的候选方案，从整个系统生命周期的角度进行分析和评价。可行性研究包括明确任务、调查环境、提出方案、分析可行性4个方面。

(2) 需求分析。需求分析的目的是解决“做什么”的问题。它是在可行性分析的基础上，针对现行系统进行全面的调查分析，从而提出新系统的逻辑模型。

(3) 系统设计。系统设计由总体设计和详细设计两个阶段组成。是为实现系统分析提出的系统逻辑模型所做的各种技术考虑与设计，它根据新系统的逻辑模型建立系统的物理模型，确定系统的实施方案，从而解决系统“怎么做”的问题。

(4) 系统实施与维护。系统实施是真正解决“具体做”的问题，它是新系统付诸实现的实施阶段。系统实施阶段是具体实现系统设计阶段的新系统的物理模型。

系统实施主要包括软硬件准备、程序设计、数据收集与准备、人员培训、系统测试、系统转换（新旧系统转换）和系统评价等部分。

系统运行与维护是系统工程生命周期的最后一个阶段。新系统要具有长久的生命力，必须不断地完善，以适应各种变化。

在 2-1 图中，实线代表前后阶段的关系，完成前一阶段并评审合格后，开始下一个阶段的工作。虚线表示当在完成本阶段工作的时候，发现了上一个阶段工作的疏漏或者是错误，要及时返回到上一个阶段去改正，而改正的过程是重新进行上一个阶段工作的过程，同样要经过评审合格以后才能进行下一个阶段的工作。

生命周期法的主要优点是：在整个开发过程中阶段和步骤清楚，每一阶段和步骤均有明确的成果，这些成果以可行性分析报告、系统分析说明书、系统设计说明书等文档形式表现出来，并作为下一阶段工作流的依据，所以生命周期法也被叫做文档驱动的开发方法。在每一阶段和步骤中，整个项目可以划分为许多组成部分，各个部分可各自独立地开展工作，且各部分的工作必须按要求到达各阶段或步骤的终点，这有利于整个项目的管理与控制。

但是，在实践过程中，生命周期法也有一些缺陷：

(1) 难以准确定义用户需求。系统的开发过程是一个线性发展的“瀑布模型”，各阶段须严格按顺序进行，并以各阶段提供的文档的正确性和完整性来保证最终应用软件产品的质量，这在许多情况下是难以做到的。用户在初始阶段提出的要求往往不全面也不明确，而在设计过程中，用户可能感到最初的目标达不到要求，需要修改，这给开发工作不仅带来较大的工作量，而且使开发工作存在较大的难度。

(2) 开发周期长，难以适应环境变化。对于一个比较大的系统，开发工作可能需要 2~3 年，在此期间，用户的要求会越来越高，环境的变化可能使原提出的配置、设计要重新考虑。

(3) 系统开发成本高，效率低。系统开发的各个阶段的工作从系统分析、系统设计到系统实施，绝大部分靠人工完成。

2.2.2 原型法

原型法是 20 世纪 80 年代随着计算机软件技术的发展，特别是在关系型数据库系统(Relational Data Base System, RDBS)、第四代程序生成语言(4th Generation Language, 4GL)和各种系统开发生成环境产生的基础上，提出的一种开发方法。它试图改进生命周期法的缺点，它不要求像生命周期法那样对原系统进行深入全面的调查和分析。其基本思想是：在投入大量的人力、物力之前，开发者和用户在系统的重要需求上取得一致意见后，由开发者在短期内开发出一个功能不十分完善、实验性的、简易的应用管理软件的基本框架(称其为原型)，先运行整个原型，通过对其评价，不断修改、改进、扩展，使之逐步完善，直至形成一个相对稳定的系统。它摒弃了那种一步一步周密细致地调查分析，然后逐步整理出文字档案，最后才能让用户看到结果的烦琐做法。

原型法又主要分为两种：一种是丢弃原型，另一种是进化原型。

(1) 丢弃原型。使用这种方法的目的是为了明确需求，首先用最简单的方法以低成本在短时间内开发出一个可工作的系统，这个系统可以实现用户最初要求的功能，而不用考虑开发工具、性能、最终系统运行的环境等。然后通过和用户反复地沟通交流，修改这个原型，使原型能够充分体现用户的最终需求。在明确了需求之后，这个原型就会被丢弃。这个原型只不过是完成了生命周期法中需求分析阶段的任务，以后软件开发将按生命周期法在需求分析阶段以后的步骤来继续进行。

(2) 进化原型。这种方法从最初开始，就使用用户要求的开发工具，与用户一起工作，

从最初的需求开始，到完成一个原始的系统，然后逐步改进，最终发展成为符合用户最终需要的系统。采用这种方法，原型没有被丢弃，但在开发前就需要考虑到软件要求的最终环境，从一开始就要对需求有一个比较清晰的认识。

原型法的主要优点表现在：

(1) 原型法的开发过程是一个循环往复的反馈过程，符合用户对计算机应用认识逐步发展、螺旋式上升的规律。开始时，用户和设计者对于系统的功能要求的认识是不完整的、粗糙的。通过建立原型、评价原型、修改原型的循环过程，设计者以原型为媒介，及时取得来自用户的反馈信息，不断发现问题，反复修改，完善系统，确保用户要求得到较好满足。

(2) 原型法很具体，使用户能很快接触和使用系统，容易为不熟悉计算机应用的用户所接受，可提高用户参与系统开发的积极性。这样进行开发，系统也更加贴近实际；易学易用，减少用户的培训时间。

(3) 原型法开发周期短，费用相对少，使用灵活，对于管理体制和组织结构不稳定、有变化的系统比较适合。

但原型法有一定的局限性，这主要表现在：

(1) 对于一个大型的系统，如果不经系统分析来进行整体性划分，想要直接用屏幕来一个一个模拟是很困难的，所以不适合大规模系统的开发。

(2) 对于大量运算的、逻辑性较强的程序模块，原型法很难构造出模型来供人评价。

(3) 对于原基础管理不善、信息处理过程混乱的情况，使用有一定的困难。首先是由于工作过程不清，构造原型有一定困难；其次是由于基础管理不好，没有科学合理方法可依，系统开发容易走入机械地模拟原来手工系统的误区。开发过程的管理也存在难度，缺乏规范化的文档资料。

2.2.3 螺旋模型

软件是逻辑产品，它的开发是有风险的。生命周期法和原型法都没有对软件开发过程中的风险进行分析，将风险分析引入到开发中，并将生命周期法和原型法结合起来，构成的模型就是螺旋模型。螺旋模型将瀑布模型与演化模型结合起来，并且加入两种模型均忽略了的风险分析。螺旋模型沿着螺线旋转，如图 2-2 所示，在笛卡尔坐标的四个象限上分别表达了以下四个方面的活动。

(1) 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件。

(2) 风险分析：分析所选方案，考虑如何识别和消除风险。

(3) 实施工程：实施软件开发。

(4) 客户评估：评价开发工作，提出修正建议。

沿螺线自内向外每旋转一圈便开发出更为完善的一个新的软件版本。

2.2.4 面向对象的开发方法

传统的开发方法在一定程度上解决了软件危机中的问题，但传统的开发方法开发出来的软件存在着软件重用性差、软件的可维护性差和所开发出的软件系统往往不能真正满足用户的需要等问题。面向对象的技术起初是用在程序设计中的，专指在程序设计中采用封装、继承、抽象等设计方法。面向对象的思想已经涉及软件开发的各个方面。如面向对象的系统分析（Object Oriented Analysis, OOA），面向对象的系统设计（Object Oriented Design, OOD）、以

及我们经常说的面向对象的编程实现（Object Oriented Programming, OOP）。

面向对象技术就是以对象的观点来解决现实世界中的问题。从普通人认识世界的观点出发，把事物归类、综合，提取出共性并加以描述。在面向对象的系统中，世界被看成是独立对象的集合，对象之间通过消息相互通信，对象具有“智能化”的结构，它将数据和消息“封装”在一起，对一个对象的访问完全通过其外部的接口来进行，内部的实现细节、数据结构对外是不可见的。

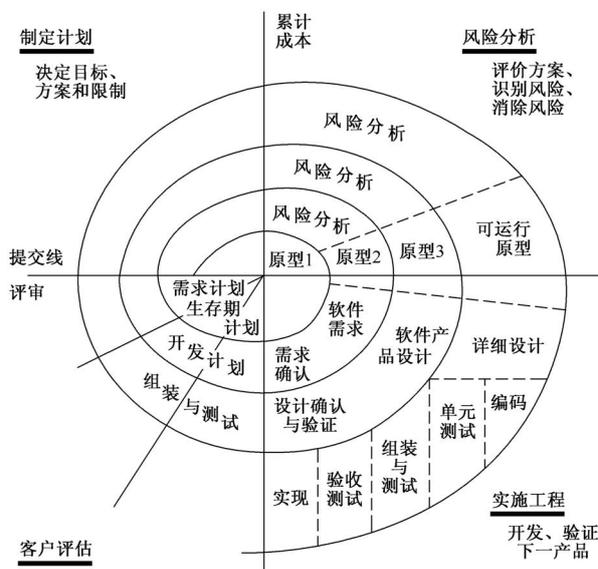


图 2-2 螺旋模型

面向对象方法是一种分析方法、设计方法和编程方法。它是一种围绕真实世界的概念来组织模型的全新的思考问题的方式，其基本思想是：对问题空间进行自然分割，以更接近人类思维的方式，建立问题域模型，以便对客观实体进行结构模拟和行为模拟，从而使所设计出的软件尽可能直接地描述现实世界，构造出模块化的、可重用的、维护性好的软件，并能够控制软件的复杂性和降低开发维护费用。在面向对象方法中，对象作为描述实体的统一概念，把数据和对数据的操作融为一体，通过方法、消息、类、继承、封装和实例化等机制构造软件系统，并为软件重用提供强有力的支持。

(1) 面向对象的系统分析。面向对象的系统分析（Object Oriented Analysis, OOA）是采用从特殊到一般的归纳方法，对现实世界中的实体进行分类，区分对象及其属性，整理对象及其组成部分，划分成不同的对象类，从而得到现实系统中对象及其关系，进而分析并掌握系统运行的规律。OOA 的重点是使用面向对象的观点解决现实世界模型的建立问题。OOA 是利用问题领域中找出的类和对象的观点来研究系统实际需求的一种分析方法。

(2) 面向对象的系统设计。面向对象的系统设计（Object Oriented Design, OOD）是采用从一般到特殊的演译方法，基于对现实系统的认识，指出其中不合理的环节，对相应的类与类之间的关系进行改造，设计出更合理的系统。

OOD 是一种设计方法，它包含两个重要的方面：

①面向对象的分解过程。

②使用面向对象表示方法描述所设计系统的逻辑模型（类和对象结构）和物理模型（模块和过程体系结构），以及描述系统的静态和动态模型。

OOD 提供的面向对象分解是使得面向对象设计方法在本质上区别于结构化设计方法的重要方面。

（3）面向对象的编程。面向对象的编程（Object Oriented Programming, OOP）是一种系统实施方法，在这种方法中，程序由一组相互协作的对象组成，其中每个对象是某个类的实例，这些产生对象的类都是由继承关系形成的类层次结构中的一个成员。

通俗地理解 OOP 是指在软件开发中，把 OOD 的结果代码化的方法，即编程方法，其中程序所处理的是一组相互关联协作的对象，而这些对象又是按一定关系组织在一起的许多类的实例，OOP 需要有相应的开发环境（如 C++）的支持。OOP 有三个重要方面：

- ①OOP 中使用的是对象而不是算法作为其基本的系统逻辑组块；
- ②每个对象是某个类的实例；
- ③类之间通过继承关系连接。

这三个方面缺少任何一个都不能称为面向对象的编程。

2.2.5 计算机辅助设计开发方法

计算机辅助软件工程（Computer Aided Software Engineering, CASE）原来是指用来支持管理信息系统开发的，由各种计算机辅助软件和工具组成的一个大型综合性软件开发环境，随着各种工具及软件技术的发展、完善和不断集成，逐步由单纯的辅助开发工具环境转化为一种相对独立的方法。

计算机辅助软件工程的目标在于使整个软件生命周期各阶段开发过程自动化，就像第四代语言使软件系统编程自动化一样。CASE 技术是系统开发工具与方法的结合，它的目标是为了实现一种较完善的技术，为系统开发人员提供一组优化的、集成的且能大量节省人力的系统开发工具，它着眼于系统分析和设计以及程序实现和维护等各个环节的自动化，并使之成为一个整体。CASE 是基于现在的软件工程方法来进行的。

CASE 方法解决系统开发问题的基本思想是：结合系统开发的各种具体方法，在完成对目标系统的规划和详细调查后，如果系统开发过程中的每步都相对独立且一定程度上彼此形成对应关系，则整个系统开发就可以应用专门的软件开发工具和集成开发环境来实现。

CASE 方法的基本思路决定了 CASE 环境具有如下的特点：

（1）在实际开发一个系统中，CASE 环境的应用必须依赖于一种具体的开发方法。例如结构化方法、原型法、面向对象方法等，而一套大型完备的 CASE 产品，能为用户提供支持上述各种方法的开发环境。

（2）CASE 是一种辅助的开发方法。这种辅助主要体现在它能帮助开发者方便、快捷地产生出系统开发过程中各类图表、程序和说明性文档。

（3）由于 CASE 环境的出现从根本上改变了我们开发系统的物质基础，从而使得利用 CASE 开发一个系统时，在考虑问题的角度、开发过程的做法以及实现系统的措施等方面都与传统方法有所不同，常常有人将它称之为 CASE 方法。

典型的 CASE 工具主要包括以下几类。

（1）图形工具。用图形和模型的方式表示信息系统所使用的各种技术，例如绘制流程图、结构图或者与其他设计方法有关的图表工具。

- (2) 原型设计工具。用于快速生成用户界面、报表等。
- (3) 代码生成器。从原型系统的工具中自动产生可执行的程序源代码。
- (4) 测试工具。用于测试各类错误，包括对程序的结构、生成的源代码、系统集成的测试。
- (5) 文档资料生成工具。用于产生结构化方法所需的各种技术文档和用户系统文档。

其他的工具还有：语句校对程序、信息知识库、开发方法和项目管理工具等。

CASE 开发方法的优点在于，它是高度自动化的系统开发方法，只要在分析和设计阶段严格按照 CASE 方法规定的处理过程，就能够将分析、设计的结果让计算机辅助工具自动来完成。另外，CASE 方法的开发方法以及开发过程的规范性、可靠性、开发效率都很好。

CASE 开发方法的缺点在于开发者在系统开发中需要熟练掌握 CASE 工具，另外目前也缺乏全面完善的 CASE 工具。

2.3 系统开发的模式

信息系统开发的模式基于信息系统的体系结构，在信息系统发展的过程中，信息系统经历了单机模式、中央集中模式、客户机/服务器（Client/Server，C/S）模式、浏览器/服务器（Browser/Server，B/S）模式 4 种模式的发展。

2.3.1 单机模式

单机模式信息系统是早期最简单的信息系统，目前仍有着一定范围的应用，一般适用于系统规模较小、数据量不大的情况。在单机模式中，客户端应用程序和数据库服务器安装在同一台计算机上，数据库一般使用本地数据（如 Microsoft Access、Visual FoxPro）。应用程序通过本地化的数据引擎来访问本地数据库，其他客户端无法通过网络访问到本地数据库。单机模式的信息系统开发简单、费用便宜、开发周期短，但不利于多用户共享系统数据，不支持多用户并发控制，数据处理也不能够满足较大系统的要求。

2.3.2 中央集中模式

中央集中模式是数据库系统的早期产品，是一种采用大型服务器主机和终端结合的系统，这种模式将操作系统、数据库管理系统和访问数据库的应用程序都放在作为核心的主机上运行，终端和主机之间以专线的方式连接，终端没有任何处理能力，只用来显示主机发来的数据以及供用户输入数据。这种模式的最大优点是管理员能以中央控制方式充分掌握系统的安全性，但对主机的性能要求比较高，并且主机的价格昂贵，还要有专人来管理和维护。随着计算机网络技术和个人计算机的性能大幅度提高而价格又大幅度下降，这种模式逐渐被 C/S 模式所替代。

2.3.3 客户机/服务器模式

客户机/服务器模式又称 C/S 结构，是随着计算机网络技术的发展而发展起来的。C/S 结构的关键在于功能的分布，一些功能放在前端机（即客户机）上执行，另一些功能放在后端机（即服务器）上执行。C/S 模式简单地讲就是基于局域网的信息系统模式。

服务器负责管理数据库的访问，为多个客户程序管理数据，并对数据库进行检索和排序，此外还要对客户机/服务器网络结构中的数据库安全层加锁，进行保护。客户机负责与用户的交

互，收集用户信息，通过网络向服务器请求对诸如数据库、电子表格或文档等信息的处理工作。

最简单的 C/S 模式数据库应用由两部分组成，即客户应用程序和数据库服务器程序。两者可分别称为前台程序与后台程序。运行数据库服务器程序的机器，称为应用服务器，服务器程序启动后，就随时等待响应客户程序发来的请求；客户程序在客户使用的计算机上运行，客户使用的计算机称之为客户机。当需要对数据库中的数据访问时，客户程序就自动寻找服务器程序，并向其发出请求，服务器程序根据预定的规则进行应答，送回结果。应用的形式如图 2-3 所示。

C/S 结构的优点是能充分发挥客户端 PC 的处理能力，很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。

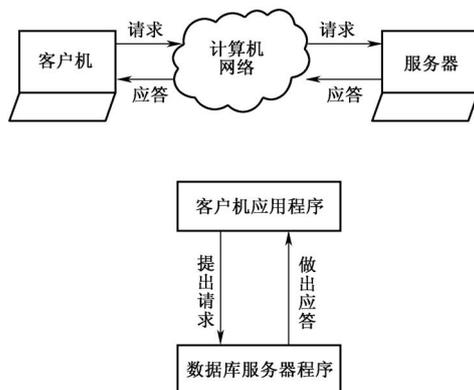


图 2-3 C/S 模式

C/S 结构的缺点主要有以下几个：

一般只适用于局域网。而随着互联网的飞速发展，移动办公和分布式办公越来越普及，这需要系统具有扩展性。这种方式的远程访问需要专门的技术，同时要对系统进行专门的设计以便处理分布式的数据。

客户端需要安装专用的客户端软件。首先涉及安装的工作量，其次任何一台电脑出问题，如病毒、硬件损坏，都需要进行安装或维护。特别是当客户端数量很多，甚至是客户端在异地的时候，工作量非常大。还有，系统软件维护和升级时，每一台客户机都需要重新安装，其维护和升级成本非常高。

对客户端的操作系统一般也会有限制。可能适应于 Windows 98，但不能用于 Windows 2000 或 Windows XP。或者不适用于微软新的操作系统等，更不用说 Linux、UNIX 等操作系统了。

2.3.4 浏览器/服务器模式

浏览器/服务器模式又称 B/S 结构。它是随着 Internet 技术的兴起，对 C/S 模式应用的扩展。在这种结构下，用户工作界面是通过 IE 浏览器来实现的。用户在客户端通过 Web 浏览器向 Web 服务器提出请求，由 Web 服务器对数据库进行操作，并将结果传回客户端。用户端只需要安装如 Netscape 和 IE 一类的浏览器，而应用程序、数据库都安装在服务器一端。

在 B/S 体系结构系统中，用户通过浏览器向分布在网络上的许多服务器发出请求，服务

器对浏览器的请求进行处理,将用户所需的信息返回到浏览器。B/S 结构简化了客户机的工作,客户机上只需配置少量的客户端软件即可。但是服务器将担负更多的工作,对数据库的访问和应用程序的执行都将在服务器上完成。即当浏览器发出请求后,其数据请求、加工、返回结果以及动态网页生成等工作全部由 Web 服务器完成。如图 2-4 所示。

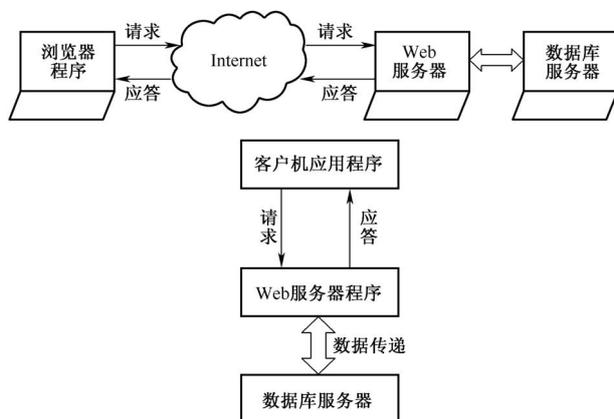


图 2-4 B/S 模式

B/S 结构的优点:

- (1) 具有分布性特点,可以随时随地进行查询、浏览等业务处理;
- (2) 业务扩展简单方便,通过增加网页即可增加服务器功能;
- (3) 维护简单方便,只需要改变网页,即可实现所有用户的同步更新;
- (4) 开发简单,共享性强。

B/S 结构的缺点:

- (1) 个性化特点明显降低,无法实现具有个性化的功能要求;
- (2) 操作是以鼠标为最基本的操作方式,无法满足快速操作的要求;
- (3) 页面动态刷新,响应速度明显降低;
- (4) 功能弱化,难以实现传统模式下的特殊功能要求;
- (5) 对企业外网环境依赖性太强,由于各种原因引起企业外网中断都会造成系统瘫痪。

2.3.5 C/S 模式和 B/S 模式的比较

下面从几个方面来对 C/S 模式和 B/S 模式进行比较。

(1) 开发和维护成本。C/S 开发和维护成本较高。对不同客户端需要开发不同的程序,且应用程序的安装、修改和升级,均需要在所有的客户机上进行。而 B/S,客户端只需有通用的浏览器,所有的维护与升级工作都是在服务器上进行,无需对客户端进行任何改变,因而大大降低了开发和维护的成本。

(2) 客户端负载。C/S 的客户端具有显示与处理数据的功能,负载重。随着应用系统的功能越来越复杂,客户端的应用程序也变得越来越庞大。B/S 的客户端把事务处理逻辑部分给了服务器,客户端只需进行显示即可,俗称为“瘦”客户机。

(3) 可移植性。C/S 移植困难,因为不同开发工具开发的应用程序,一般情况下互不兼容,

难以移植到其他平台上运行。对于 B/S，在客户端安装的是通用浏览器，不存在移植性问题。

(4) 用户界面。C/S 用户的界面是由客户端所安装的软件决定，因此用户界面各不相同；而 B/S 通过通用的浏览器访问应用程序，其浏览器的界面统一，使用时类似于浏览网页。

(5) 安全性。C/S 适用于专人使用的系统，可以通过严格的管理派发软件，适用于安全性要求较高的专用应用软件；而 B/S 适用于交互性要求较多、使用人数较多、安全性要求不是很高的应用环境。

综上所述，这两种开发模式都是网络环境下的开发模式，B/S 相对于 C/S 具有更多的优势，现如今大量的应用开始应用 B/S 模式，许多软件公司争相开发 B/S 版本的软件。由于 Internet 逐步走进人们的日常生活当中，电子商务进一步应用的需求，客户简便化的使用要求等对加速推广使用 B/S 模式起到了推波助澜的作用。

2.4 系统开发的工具

信息系统开发中一个重要的问题就是对于开发工具的选择。目前常用的信息系统主要是基于 C/S 模式和 B/S 模式的，所以开发工具也主要包括两部分：一个是前端开发工具的选择，另一个是后端数据库管理软件的选择。另外在开发过程当中还会使用到一些其他软件。

2.4.1 前端开发工具

前端开发工具主要在前台给用户提供用户界面，并且要和后台数据库进行数据交换。主要有如下几种。

1. PowerBuilder

PowerBuilder 是 Powersoft 公司于 1991 年发布的软件，最早版本为 PowerBuilder 1.0。1994 年，Powersoft 公司收购了 Waicom 公司，同时推出了 InfoMaker 产品；1995 年也是非常具有历史性的一年，Sybase 公司和 Powersoft 公司合并，成为世界第六大独立软件商，而且发布的 System 11 是符合工业标准的高性能、高可用性、易扩展的新一代 SQL Server 产品；1996 年，Sybase 公司收购了 Visual Components 公司，同时推出了最经典的一个 PowerBuilder 版本——PowerBuilder 5.0；接着，Sybase 公司又于 1997 年推出了 Adaptive Server Enterprise 11.5、Jaguar CTS 1.1、PowerBuilder 6.0、Power++ 2.0、PowerJ 2.0、PowerSite、PowerDesigner 6.0。但是 PowerBuilder 6.0 由于存在不少 Bug，最终升级为 PowerBuilder 6.5 这个相对成熟的版本。直到现在，很多 PowerBuilder 程序员都一直在使用这个版本。1999—2000 年推出了 PowerBuilder 7.0，它采用了全新的框架结构，开发环境和以前的 PowerBuilder 完全不一样；2001 年推出的 PowerBuilder 8.0 和 2003 年推出的 PowerBuilder 9.0 都采用了与 PowerBuilder 7.0 相同的框架结构。

它支持应用系统同时访问多种数据库，其中既包括 Oracle、Sybase 之类的大型数据库，又包括 FoxPro 之类支持 ODBC 接口的小型数据库，PowerBuilder 是完全可视化的数据库开发工具，它提供了大量的控件，大大加快了项目的开发速度，也使开发者更容易掌握数据库的开发。

它使用的编程语言叫做 PowerScript，也是一种高级的、结构化的编程语言。PowerScript 提供了一套完整的嵌入式 SQL 语句，开发人员可以像使用其他语句一样自由地使用 SQL 语言，这样就大大增强了程序操纵和访问数据库的能力。可以说 PowerBuilder 既适合初学者快速学习数据库的开发，又可以有经验的开发人员开发出功能强大的数据库，是一种适用面非常广的开发工具。

PowerBuilder 是一个用来进行客户机/服务器开发的完全的可视化开发环境。使用 PowerBuilder, 可以用一种可视的直观的方式来创建应用程序的用户界面和数据库接口。这是一个总的概念, 实际上是开发人员使用 PowerBuilder 去开发应用程序, 由于所开发的各种应用程序充分利用了图形用户接口 (GUI) 的优点, 所以 PowerBuilder 被认为是一个图形工具。

在客户机/服务器结构的应用中, PowerBuilder 具有描述多个数据库连接与检索的能力。特别是 PowerBuilder 能从大多数流行的 RDBMS 中存取数据, 且不管数据存放在什么地方; 另外, 各种应用程序又可以独立于 RDBMS, 因为 PowerBuilder 可以使用数据库的标准操作语言 SQL (结构化查询语言)。

使用 PowerBuilder, 可以很容易地开发出功能强大的图形界面的访问服务器数据库的应用程序, PowerBuilder 提供了建立符合工业标准的应用程序 (例如订单登记、会计及制造系统) 所需的所有工具。

PowerBuilder 应用程序由窗口组成, 这些窗口包含用户与之交互的控件。开发人员可以使用所有标准控件 (如按钮、复选框、下拉列表框或编辑框) 以及 PowerBuilder 提供的特殊的使应用程序更易于开发和使用的控件。

通常人们把 PowerBuilder 看成是一种开发工具, 实际上它比其他工具强得多, 是一种强有力的开发环境。开发人员不仅能用它来开发用户容易使用的各种应用程序, 还可以通过 PowerBuilder 修改数据库。

由于 PowerBuilder 融入了面向对象的思想 and 可视化技术, 提供了可视化的应用开发环境, 使得用户可以运用它以一种“所见即所得”的方式快速创建应用程序的用户界面, 开发出基于后台数据库的信息系统。

在传统的客户机/服务器的开发模式中, PowerBuilder 具有其他信息系统开发工具无法比拟的优势。它提供了多种接口标准, 支持信息系统同时访问多种数据库, 不仅包括 Oracle、Informix、Sybase、SQL Server 等提供专用接口的大型数据库, 同时也包括 Access、FoxPro 等以 ODBC 为标准接口的小型数据库。另外, 其自带的 Adaptive Server Anywhere (ASA) 是一个功能强大的数据库管理系统, 对小型信息系统而言, ASA 不失为一个理想的选择。

另外, PowerBuilder 提供了数十个窗口、菜单等可视化对象和数据存储、数据管道等不可视对象, 并对这些对象赋予了数以千计的属性、函数、事件, 加上它的图形化界面、事件驱动和面向对象的脚本语言编程方法, 使得 PowerBuilder 成为快速应用开发的最优秀的第 4 代语言之一。PowerBuilder 具有开发快捷、功能强大、数据库访问强、编程代码简便、系统封装性好等特点。

2. Delphi 和 C++ Builder

由 Borland 公司推出的 Delphi 具有可视化的编程环境, 是一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想, 采用了可重复利用的完整的面向对象程序语言 (Object-Oriented Language)。

Delphi 可在 Windows 3.x、Windows 95、Windows NT、Windows XP、Windows Vista 等环境下使用。当前, Delphi 也可以在 Linux 平台上开发应用, 其在 Linux 上的对应产品是 Kylix。

Delphi 拥有一个可视化的集成开发环境 (IDE), 采用面向对象的编程语言 Object Pascal 和基于部件的开发结构框架。Delphi 提供了 500 多个可供使用的构件, 利用这些部件, 开发人

员可以快速地构造出应用系统。开发人员也可以根据自己的需要修改部件或用 Delphi 本身编写自己的部件。

“真正的程序员用 VC，聪明的程序员用 Delphi”，这句话是对 Delphi 最经典、最实在的描述。Delphi 被称为第四代编程语言，它具有简单、高效、功能强大的特点。和 VC 相比，Delphi 更简单、更易于掌握，而在功能上却丝毫不逊色；和 VB 相比，Delphi 则功能更强大、更实用。可以说 Delphi 同时兼备了 VC 功能强大和 VB 简单易学的特点。它一直是程序员至爱的编程工具。

Delphi 具有的特性：基于窗体和面向对象的方法，高速的编译器，强大的数据库支持，与 Windows 编程紧密结合，强大而成熟的组件技术。但最重要的还是 Object Pascal 语言，它才是一切的根本。Object Pascal 语言是在 Pascal 语言的基础上发展起来的，简单易学。

Delphi 提供了各种开发工具，包括集成环境、图像编辑 (Image Editor)，以及各种开发数据库的应用程序，如 Desktop Database Expert 等。除此之外，还允许用户挂接其他的应用程序开发工具，如 Borland 公司的资源编辑器 (Resource Workshop)。

在 Delphi 众多的优势当中，它在数据库方面的特长显得尤为突出：适应于多种数据库结构，从客户机/服务器模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和提供大量的企业组件。

它在本质上是一个代码编辑器而不是一种语言，但是由于 Delphi 几乎是目前市场上唯一一个使用 Pascal 语言并持续推出新版的产品，因此有的时候人们称呼 Delphi 为 Object Pascal 的代名词。因此 Borland 公司已经把 Object Pascal 语言改称为 Delphi 语言。2006，Borland 把开发工具产品线独立出来，成立了一家新的公司 CodeGear。原本的 Delphi 一度被命名为 Turbo Delphi。但是在新的版本中，除了 Delphi for Win32 (原有的 Delphi 版本) 之外，还有一个新的产品被命名为 Delphi for PHP。因此，在目前的 CodeGear 的发展规划中，或许“Delphi”这个名称不再是 Object Pascal 语言的专有 IDE，而是将被扩展为一个通用的 IDE 名称。

C++ Builder 是由 Borland 公司继 Delphi 之后又推出的一款高性能可视化集成开发工具。C++ Builder 具有快速的可视化开发环境：只要简单地把控件 (Component) 拖到窗体 (Form) 上，定义一下它的属性，设置一下它的外观，就可以快速地建立应用程序界面；C++ Builder 内置了 100 多个完全封装了 Windows 公用特性且具有完全可扩展性 (包括全面支持 ActiveX 控件) 的可重用控件；C++ Builder 具有一个专业 C++ 开发环境所能提供的全部功能：快速、高效、灵活的编译器优化，逐步连接，CPU 透视，命令行工具等。它实现了可视化的编程环境与功能强大的编程语言 (C++) 的完美结合。

3. Visual Studio

Visual Studio 是微软公司推出的开发环境，是目前最流行的 Windows 平台应用程序开发环境。目前已经开发到 10.0 版本，也就是 Visual Studio 2010。

Visual Studio 并不是一种开发语言，而是集成了若干个开发工具，可以用来创建 Windows 平台下的 Windows 应用程序和网络应用程序，也可以用来创建网络服务、智能设备应用程序和 Office 插件。

Visual Studio 97 是最早的 Visual Studio 版本，包含面向 Windows 开发使用的 Visual Basic 5.0、Visual C++ 5.0，面向 Java 开发的 Visual J++ 和面向数据库开发的 Visual FoxPro，还包含创建 DHTML (Dynamic HTML) 所需要的 Visual InterDev。其中，Visual Basic 和 Visual FoxPro 使用单独的开发环境，其他的开发语言使用统一的开发环境。

1998年,微软公司发布了 Visual Studio 6.0。所有开发语言的开发环境版本均升至 6.0。

2002年,随着.NET口号的提出与 Windows XP/Office XP 的发布,微软发布了 Visual Studio .NET(内部版本号为 7.0)。在这个版本的 Visual Studio 中,微软剥离了 Visual FoxPro,作为一个单独的开发环境以 Visual FoxPro 7.0 单独销售,同时取消了 Visual InterDev。与此同时,微软引入了建立在 .NET 框架上(版本 1.0)的托管代码机制以及一门新的语言——C#(读作 C Sharp,意为 C++++)。C# 是一门建立在 C++和 Java 基础上的现代语言,是编写 .NET 框架的语言。Visual Basic、Visual C++都被扩展为支持托管代码机制的开发环境,且 Visual Basic .NET 更是从 Visual Basic 脱胎换骨,彻底支持面向对象的编程机制。而 Visual J++ 也变为 Visual J#。后者仅语法与 Java 相同,但是面向的不是 Java 虚拟机,而是 .NET Framework。

2005年,微软发布了 Visual Studio 2005。.NET 字眼从各种语言的名字中被抹去,但是这个版本的 Visual Studio 仍然还是面向 .NET 框架的(版本 2.0)。

4. Java

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。用 Java 实现的 HotJava 浏览器(支持 Java Applet)显示了 Java 的魅力:跨平台、动态的 Web、Internet 计算。从此,Java 被广泛接受并推动了 Web 的迅速发展,常用的浏览器现在均支持 Java Applet。

Java 平台由 Java 虚拟机(Java Virtual Machine)和 Java 应用编程接口(Application Programming Interface, API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个 Java 平台之后,Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序可以只编译一次,就可以在各种系统中运行。Java 应用编程接口已经从 1.1x 版发展到 1.2 版。目前常用的 Java 平台基于 Java 1.5,最近版本为 Java 1.7。

Java 分为三个体系:Java SE(Java 2 Platform Standard Edition,Java 平台标准版),Java EE(Java 2 Platform, Enterprise Edition,Java 平台企业版),Java ME(Java 2 Platform Micro Edition,Java 平台微型版)。

Java SE 以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类,并为 Java EE 提供基础。

Java EE 以前称为 J2EE。企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的,它提供 Web 服务、组件模型、管理和通信 API,可以用来实现企业级的面向服务体系结构(Service Oriented Architecture, SOA)和 Web 2.0 应用程序。

Java ME 以前称为 J2ME。Java ME 为在移动设备和嵌入式设备(比如手机、PDA、电视机顶盒和打印机)上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连网和离线应用程序的支持。基于 Java ME 规范的应用程序只需编写一次,就可以用于许多设备,而且可以利用每个设备的本机功能。

2009年4月20日,Oracle(甲骨文)宣布收购 Sun,Java 一定有着更为广阔的前景。

2.4.2 后端数据库管理软件

1. Access

Microsoft Office Access (前名 Microsoft Access) 是由微软发布的关系型数据库管理系统。它结合了 Microsoft Jet Database Engine 和 图形用户界面两项特点, 是 Microsoft Office 的成员之一。Microsoft Access 在很多地方得到广泛使用, 如小型企业、大公司的部门和喜爱编程的开发人员专门利用它来制作处理数据的桌面系统。它也常被用来开发简单的 Web 应用程序。这些应用程序都利用 ASP 技术在 Internet Information Services 运行。

Access 数据库由 7 种对象组成, 它们是表、查询、窗体、报表、宏、页和模块。

表 (Table) ——表是数据库的基本对象, 是创建其他 5 种对象的基础。表由记录组成, 记录由字段组成, 表用来存储数据库的数据, 故又称数据表。

查询 (Query) ——查询可以按索引快速查找到需要的记录, 按要求筛选记录并能连接若干个表的字段组成新表。

窗体 (Form) ——窗体提供了一种方便的浏览、输入及更改数据的窗口, 还可以创建子窗体显示相关联的表的内容。窗体也称表单。

报表 (Report) ——报表的功能是将数据库中的数据分类汇总, 然后打印出来, 以便分析。

宏 (Macro) ——宏相当于 DOS 中的批处理, 用来自动执行一系列操作。Access 列出了一些常用的操作供用户选择, 使用起来十分方便。

模块 (Module) ——模块的功能与宏类似, 但它定义的操作比宏更精细和复杂, 用户可以根据自己的需要编写程序。模块使用 Visual Basic 编程。

页 ——是一种特殊的直接连接到数据库中数据的 Web 页。通过数据访问页将数据发布到 Internet 或 Intranet 上, 并可以使用浏览器进行数据的维护和操作。

Access 是小型数据库, 具有存储方式单一、界面友好、易操作等优点, 但它的缺点也比较明显, 数据库过大或者记录数过多的时候, 它的性能会急剧下降。

2. MySQL

MySQL 是一个小型关系型数据库管理系统, 开发者为瑞典 MySQL AB 公司。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一特点, 许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。

与其他的大型数据库例如 Oracle、DB2、SQL Server 等相比, MySQL 自有它的不足之处, 如规模小、功能有限 (MySQL Cluster 的功能和效率都相对比较差) 等, 但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说, MySQL 提供的功能已经绰绰有余, 而且由于 MySQL 是开放源码软件, 因此可以大大降低总体拥有成本。

目前 Internet 上流行的网站构架方式是 LAMP (Linux+Apache+MySQL+PHP), 即使用 Linux 作为操作系统, Apache 作为 Web 服务器, MySQL 作为数据库, PHP 作为服务器端脚本解释器。由于这 4 个软件都是免费或开放源码软件 (FLOSS), 因此使用这种方式不用花一分钱就可以建立起一个稳定、免费的网站系统。

由于 2008 年 MySQL AB 公司被 Sun 公司收购。而 2009 年, Sun 又被 Oracle 收购。所以对于 MySQL 的前途, 没有任何人抱乐观的态度。因为 Oracle 公司本身拥有在数据库领域处于领导地位的 Oracle 数据库。

3. Sybase

Sybase 是美国 Sybase 公司研制的一种关系型数据库系统,是一种典型的 UNIX 或 Windows NT 平台上客户机/服务器环境下的大型数据库系统。 Sybase 提供了一套应用程序编程接口和库,可以与非 Sybase 数据源及服务器集成,允许在多个数据库之间复制数据,适于创建多层应用。系统具有完备的触发器、存储过程、规则以及完整性定义,支持优化查询,具有较好的数据安全性。Sybase 通常与 Sybase SQL Anywhere 用于客户机/服务器环境,前者作为服务器数据库,后者为客户机数据库,采用该公司研制的 PowerBuilder 为开发工具,在我国大中型系统中具有广泛的应用。

4. SQL Server

SQL Server 是一个关系型数据库管理系统。它最初是由 Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发的,于 1988 年推出了第一个 OS/2 版本。在 Windows NT 推出后,Microsoft 与 Sybase 在 SQL Server 的开发上就分道扬镳了,Microsoft 将 SQL Server 移植到 Windows NT 系统上,专注于开发推广 SQL Server 的 Windows NT 版本。Sybase 则较专注于 SQL Server 在 UNIX 操作系统上的应用。

5. Oracle

Oracle 是甲骨文 (oracle bone inscriptions) 的英文翻译的第一个单词, Oracle 的关系型数据库是世界上第一个支持 SQL 语言的数据库。1977 年, Lawrence J.Ellison 领着一些同事成立了 Oracle 公司,他们的成功强力反击了那些说关系型数据库无法成功商业化的说法。

Oracle10g 是业界第一个完整的、智能化的新一代 Internet 基础架构, Oracle10g 电子商务平台实际上是指 Oracle 数据库 10g、Oracle 应用服务器 10g 和 Oracle 开发工具套件 10g 的完整集成。

Oracle 数据库 10g 是第一套具有无限可伸缩性与高可用性,并可在集群环境中运行商业软件的互联网数据库,具有 400 多个领先的数据库功能,在集群技术、高可用性、商业智能、安全性、系统管理等方面都实现了新的突破。作为甲骨文公司长达十年的软件技术研发成果,真正应用集群技术 (Real Application Clusters) 能够提供近乎无限的扩充能力与整体可用性,为用户带来透明的、高速增长 of 的集群功能。

Oracle 应用服务器 10g 是 J2EE 认证的最轻、最快、最具伸缩性的应用服务器,提供了企业门户软件、无线支持、高速缓存、轻量级 J2EE 引擎、商务智能、快速应用开发、应用与业务集成、Web 服务等多种应用开发功能,形成完整的电子商务应用开发和部署环境。使用了 Oracle 应用服务器 10g 的用户可以通过升级软件来取代升级硬件,大大节省了基础设施的成本花费。

Oracle 开发工具套件 10g 是一套完整的集成开发工具,可用于快速开发使用 Java 和 XML 语言的互联网应用和 Web 服务,支持任何语言、任何操作系统、任何开发风格、开发生命周期的任何阶段以及所有最新的互联网标准。

目前 Oracle 最新的版本是 Oracle 11g。

2.4.3 其他工具

(1) 统一建模语言。统一建模语言 (Unified Modeling Language, UML) 确切地说并不是一种工具,而是非专利的第三代建模和规约语言。UML 是在开发阶段说明、可视化、构建和书写一个面向对象软件密集系统的制品的开放方法。UML 展现了一系列最佳工程实践,这些最佳实践在对大规模的复杂系统进行建模方面,特别是在软件架构层次已经被验证有效。

UML 可以贯穿软件开发周期中的每一个阶段。被对象管理组织 (Object Management

Group, OMG) 采纳作为业界的标准。UML 最适于数据建模、业务建模、对象建模和组件建模。

UML 作为一种模型语言, 它使开发人员专注于建立产品的模型和结构, 而不是选用什么程序语言和算法实现。当模型建立之后, 模型可以被 UML 工具转化成指定的程序语言代码。

IBM 的 Rational Rose 和 Microsoft 的 Visio 都是 UML 工具。

(2) Rational Rose。Rational Rose 是 Rational 公司出品的一种面向对象的统一建模语言的可视化建模工具。用于可视化建模和公司级水平软件应用的组件构造。

就像一个戏剧导演设计一个剧本一样, 一个软件设计师使用 Rational Rose, 以演员(数字)、使用拖放式符号的程序表中的有用的案例元素(椭圆)、目标(矩形)和消息/关系(箭头)设计各种类, 来创造(模型)一个应用的框架。当程序表被创建时, Rational Rose 记录下这个程序表然后以设计师选择的 C++、Visual Basic、Java、Oracle 8、CORBA 或者数据定义语言(Data Definition Language) 来产生代码。

Rational Rose 包括了统一建模语言(UML)、OOSE 以及 OMT。其中统一建模语言(UML) 由 Rational 公司 3 位世界级面向对象技术专家 Grady Booch、Ivar Jacobson 和 Jim Rumbaugh 通过对早期面向对象研究和设计方法的进一步扩展而得来的, 它为可视化建模软件奠定了坚实的理论基础。同时这样的渊源也使 Rational Rose 力挫当前市场上很多基于 UML 可视化建模的工具, 例如有 Microsoft 的 Visio 2002, Oracle 的 Designer 2000, 还有 PlayCase、CA BPWin、CA ERWin、Sybase PowerDesigner 等。

Rational Rose 是一套完全的具有能满足所有建模环境(Web 开发、数据建模、Visual Studio 和 C++) 需求的灵活的解决方案。Rose 允许开发人员、项目经理、系统工程师和分析人员在软件开发周期内将需求和系统的体系架构转换成代码, 消除浪费的消耗, 对需求和系统的体系架构进行可视化、理解和精练。通过在软件开发周期内使用同一种建模工具可以确保更快地创建满足客户需求的可扩展的、灵活的并且可靠的应用系统。

(3) RUP。RUP (Rational Unified Process, 统一软件开发过程, 统一软件过程) 是一个面向对象且基于网络的程序开发方法论。

根据 Rational (Rational Rose 和统一建模语言的开发者) 的说法, 好像一个在线的领导者, 可以为所有方面和层次的程序开发提供指导方针、模板以及事例支持。RUP 和类似的产品, 如面向对象的软件过程(OOSP) 以及 OPEN Process 都是理解性的软件工程工具: 把开发中面向过程的方面(如定义的阶段、技术和实践) 和其他开发的组件(如文档、模型、手册以及代码等) 整合在一个统一的框架内。

2.5 系统开发的方式

1. 用户自行开发

用户具有开发系统的基本必要条件, 且技术力量比较雄厚, 可以采取自行开发的方式。这种开发方式的优点是, 开发人员熟悉业务处理过程, 沟通交流容易, 但是, 开发人员应专心从事开发工作, 不受原来工作的影响, 也应克服组织中各方面的干扰因素。

自主开发适合于有较强的信息系统分析与设计队伍和程序设计人员、系统维护人员的组织和单位, 如高等院校、研究所、计算机公司等。独立开发的优点是开发费用少, 实现开发后的系统能够适应本单位的需求且满意度较高, 系统维护方便。缺点是由于不是专业开发队伍,

容易受计算机业务工作的限制，系统优化不够，开发水平较低。

2. 委托开发

用户将信息系统建设的规划、目标等方面的要求明确提出，可以采取招标等方式委托软件公司，通过签定合同的方式来完成开发任务。委托（外包）开发方式适合于使用单位无信息系统分析、设计及软件开发人员或开发队伍力量较弱，但资金较为充足的组织和单位。

委托开发的方式的优点是省时、省事，系统的技术水平较高。这种开发方式的不足是，风险较大，对于开发单位需要进行深入调查，所签定的开发合同的条款需要细致、明确。同时费用高，系统维护需要开发单位的长期支持。此种方式需要使用单位的业务骨干参与系统的论证工作，开发过程中需要开发单位和使用单位双方及时沟通，进行协调和检查。

3. 合作开发

由用户和开发单位共同完成系统开发任务。优点在于双方取长补短，用户在此过程中培养了一支队伍。在双方合作过程中用户应充分明确自身的职责。合作开发方式适合于使用单位有一定的管理信息系统分析、设计及软件开发人员，但开发队伍力量较弱，希望通过管理信息系统的开发完善和提高自己的技术队伍，便于系统维护工作的单位。双方共同开发成果，实际上是一种半委托性质的开发工作。优点是相对于委托开发方式比较节约资金，可以培养、增强使用单位的技术力量，便于系统维护工作，系统的技术水平较高。缺点是双方在合作中沟通易出现问题，需要双方及时达成共识，进行协调和检查。

4. 从市场上购买适合组织需要的软件

目前，软件的开发正在向专业化方向发展，一些专门从事信息系统开发的公司已经开发出一批使用方便、功能强大的专项业务管理信息系统软件。为了避免重复劳动，提高系统开发的经济效益，也可以购买现成的适合于本单位业务管理的信息系统软件，如企业管理信息系统、教育管理信息系统、财务管理系统、进销存管理系统等。这种方法的优点是购买的软件技术资料齐备，维护可靠，节省时间、系统技术水平高。缺点是市场上的软件往往具有通用性，专用性较差，对于组织的特殊情况难以充分考虑，跟本单位的实际工作需要可能有一定的差距，有时可能需要做二次开发工作，这往往会有一定的技术难度，没有有关产品供应商的协助是难以进行的。因此，在选择通用软件时，不可只看开发商的宣传，要经过多方详尽的考查后再作决定。

5. 咨询开发

以组织自己的力量为主，外请专家进行咨询的方式。主要是系统分析员进行咨询指导，如帮助做系统的总体规划和系统分析等，而系统的实施由组织自己进行。这种方式是对用户自行开发的一种补充。

这几种开发方式的选择，都要根据使用单位的技术力量、资金情况、外部环境等各种因素进行综合考虑。不论哪一种开发方式都需要使用单位的领导和业务人员参加，并在管理信息系统的整个开发过程中，培养、锻炼、壮大使用单位的系统开发、设计和维护队伍。

本章小结

信息系统开发遵循着软件工程的理念，软件工程的诞生是因为出现了软件危机。软件危机的主要表现是“已完成”的软件不能满足用户的需求；开发进度不能保障；软件开发成本难以准确估算；软件产品的质量没有保证。软件工程是采用工程的概念、原理、技术和方法来开

发与维护软件,把经过时间考验而证明正确的管理方法和先进软件开发技术结合起来,运用到软件开发和维护过程中,来解决软件危机。软件工程研究的主要内容是软件开发技术和软件开发管理两个方面。软件开发技术方面主要研究软件开发方法和软件开发过程、软件开发工具和环境。软件开发管理方面主要研究软件工程管理学、软件工程经济学、软件工程心理学。

信息系统的主要开发方法有:生命周期法、原型法、面向对象的开发方法、计算机辅助设计开发方法等。软件生命周期是指一个软件从提出开发要求开始到该软件报废为止的整个时期。通常将软件的生命周期划分为问题定义、可行性研究、需求分析、概要设计、详细设计、编码和单元测试、集成和测试、维护几个阶段。

信息系统开发的模式基于信息系统的体系结构,在信息系统发展的过程中,信息系统经历了单机模式、中央集中模式、客户机/服务器模式、浏览器/服务器模式4种模式的发展。而目前随着网络技术的发展,信息系统的开发模式越来越多地采用客户机/服务器模式和浏览器/服务器模式。

信息系统开发的工具主要包括两部分:一个是前端开发工具,另一个是后端数据库管理软件。

本章最后介绍了系统开发的几种常用方式。这几种开发方式的选择,都要根据使用单位的技术力量、资金情况、外部环境等各种因素进行综合考虑。

练习 2

- (1) 什么是软件危机?软件危机表现在哪几个方面?
- (2) 什么是软件工程?软件工程的目的是什么?
- (3) 信息系统的主要开发方法有哪些?各有什么特点?
- (4) 信息系统的开发模式有哪四种?
- (5) 信息系统开发的工具有哪些?
- (6) 简述信息系统常用的开发方式及如何进行选择。