

Microsoft

微软院校认证课程系列教材

实用软件工程方法

微软公司 著



高等教育出版社

HIGHER EDUCATION PRESS

本书的著作权归微软公司所有。未经微软公司书面许可, 本书的任何部分不得以任何形式或任何手段复制或传播。著作权人保留所有权利。

图书在版编目 (CIP) 数据

实用软件工程方法 / 微软公司著. —北京: 高等教育出版社, 2005.8
ISBN 7-04-015831-0

I. 实... II. 微... III. 软件工程-教材
IV. TP311.5

中国版本图书馆 CIP 数据核字 (2005) 第 088807 号

策划编辑 尹 洪 责任编辑 严 亮 赵 萍 封面设计 张 楠
责任绘图 吴文信 版式设计 史新薇 责任校对 胡晓琪 责任印制 宋克学

出 版 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

印 刷 北京中科印刷有限公司

开 本 787×1092 1/16
印 张 18.75
字 数 450 000

版 次 2005 年 8 月第 1 版
印 次 2005 年 8 月第 1 次印刷
定 价 77.00 元 (含光盘)

版权所有 侵权必究

物料号 15831-00

编审委员	刘志鹏	朱之文	田本和	王军伟	郑祖宪
	马肖风	王 林	王建国	罗晓中	
组织策划	田本和	尹 洪	蒋 斌	周雨阳	蔡 锴
	李朝晖	姬 琳			
技术编审	麦中凡	蒋 斌	陶 伟	蔡 锴	李 岷
	虞谷晔	夏俊平	李朝晖	喻艺丹	张广军

Microsoft Official Curriculum 最终用户许可协议

重要须知——请认真阅读——您一旦打开“许可使用内容”包装的密封或以其他方式使用此处的“许可使用内容”，即表示您同意接受本《协议》各项条款的约束：本 Microsoft Official Curriculum 可能包含 Microsoft 或其供应商提供的软件或其他材料（总称“许可使用内容”），其使用应遵守以下各项 Microsoft 提示条款。每个软件程序都受一份最终用户许可协议（《协议》）的约束，而该《协议》是您（个人或单一实体）（“最终用户”）和 Microsoft Corporation（“Microsoft”）之间就允许使用软件及相关介质或印刷材料、“联机”或电子文档和基于 Internet 的服务达成的一份法律协议。本《协议》的修正条款或补充条款可能随软件一起提供。您一旦安装、复制或以其他方式使用“许可使用内容”，即表示您同意接受本《协议》各项条款的约束。如果您不同意，请（a）不要打开“许可使用内容”包装的密封；（b）不要使用软件、文档或其他材料，并且（c）退还“许可使用内容”。

“许可使用内容”随带软件的特别提示

许可证的授予。为与本“许可使用内容”一起使用而提供的任何软件（“软件”）都是 Microsoft Corporation 和（或）其供应商享有著作权的作品。“软件”只授予使用许可，而非出售。任何特定“软件”的使用都应遵守以下各《许可协议》中的一份《许可协议》：

（1）一般使用许可。Microsoft 授予最终用户一份有限的、非独家拥有的、免版权费的许可证，许可其在一台由一位单一用户随时使用或访问的单一计算机上安装和使用“软件”的一份副本，并且最终用户：（a）不得修改“软件”，但下文有明确规定时例外；（b）不得发行“软件”或其任何组成部分；（c）不得出借、出租、租赁、出售、分许可、转让“软件”或将“软件”随附的任何印刷材料用于提供商业运营服务；（d）不得在收费的公立或私立课程中使用“软件”；（e）不得对“软件”进行反向工程、反编译或反汇编；尽管有此项限制，但如果适用法律明确允许上述活动并且仅在适用法律明示允许上述活动的范围内，则例外；并且（f）不得转让“软件”的各项权利，除非本《协议》明确规定。

Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》具体说明。

（2）替代使用许可。上述规定的一般使用许可将被任何具体“软件”随附或包括的《许可协议》（如果有）的各项条款取代或替代。除非最终用户首先同意《许可协议》的各项条款，否则将无法安装附带或包括该《许可协议》的“软件”。Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》明确说明。

（3）样本代码使用许可。如果将特定代码或一个样本应用程序作为“许可使用内容”中包括的实验室练习的部分提供（“样本代码”），则这类“样本代码”以“现有状况”被提供，并且没有任何类型的保证。Microsoft 授予您一份有限的、非独家拥有的、免版权费的许可证，

许可您为了个人使用的目的而安装、使用、修改和复制“样本代码”，条件是您不得：（a）发行“样本代码”或其任何组成部分；（b）出借、出租、租赁、出售、分许可或转让“样本代码”；（c）在收费的公立或私立课程中使用“样本代码”；并且（或者）（d）转让“样本代码”的任何权利。如果您修改“样本代码”，您应该根据 Microsoft 的请求，自付费用于因您或代表您对“样本代码”做出的任何修改而使 Microsoft 和 Microsoft 的分公司、关联公司、董事、高级主管、员工、代理商和独立供应商面临的任何索赔或诉讼提供辩护，并且您须赔偿 Microsoft 因这类索赔而招致的任何费用、损害赔偿和手续费方面的合理开支（其中包括但不限于律师费和其他专业人士收取的费用），并使其免受任何损害。Microsoft 应：（a）以书面形式就任何这类索赔或诉讼向您提供合理的及时提示，并且允许您通过 Microsoft 和您双方都接受的律师对这类索赔或诉讼进行答辩和辩护；（b）在您支付费用的情况下向您提供信息、协助和授权，以帮助您为这类索赔或诉讼进行辩护。您不对 Microsoft 在未经您书面允许的情况下做出的任何和解负责，但您不得以不合理的方式拒绝给予这样的允许。

其他许可限制。安装“软件”仅供最终用户根据适用的《许可协议》使用，并且除非以其他方式在另外一份协议中达成一致意见，否则得不到 Microsoft 或其供应商提供的技术或其他支持服务。法律明确规定：禁止在违反《许可协议》的情况下对“软件”进行任何复制或再发行。明确禁止为进一步复制或再发行软件而将“软件”复制到任何服务器或地点。

美国政府许可使用权利。根据 1995 年 12 月 1 日当天或之后签发的请求而提供给美国政府的所有软件，均根据本协议其他部分规定的商业许可使用权利和限制予以提供。根据 1995 年 12 月 1 日之前签发的请求而提供给美国政府的所有软件，视情况根据 FAR, 48 CFR 52.227-14 (1987 年 6 月) 或 DFAR, 48CFR252.227-7013 (1988 年 10 月) 中规定的“限制权利”予以提供。

免责条款。“软件”仅根据《许可协议》的各项条款对“软件”提供保证（如果提供保证的话）。除非在《许可协议》中提供保证，否则 Microsoft Corporation 和（或）其供应商就“软件”不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

“许可使用内容”随带文档和（或）其他材料的具体说明

允许从“许可使用内容”（“文档”）中打印文档（如实验室说明等），条件是：（a）将这类文档用于您的个人培训，并且不得再出版或在任何网络计算机上张贴或以任何介质形式广播这类文档，并且（b）不得对任何文档做出任何修改。

明确禁止对任何介质上包含的作为“许可使用内容”组成部分的录像、录音、图形和（或）任何其他材料（“其它材料”）进行任何复制或再发行。

“许可使用内容”的各组成部分均受商业包装法律和其他法律的保护，并且不得全部或部分予以复制或模仿。除非 Microsoft 明示允许，否则不得复制或转发“许可使用内容”中的任何徽标、图形、声音或图像。

无保证。Microsoft 和（或）其供应商不对“许可使用内容”中不论为任何目的而可能包含的文档或其他材料中的信息、音像或任何其他内容是否合适提供任何保证，无论该类文档、信息、音像或任何其他内容是何目的。所有这类文档和其他材料均以“现有状况”提供，没有

任何类型的保证。Microsoft 和（或）其供应商特此就文档和其他材料不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

有关第三方站点链接的说明

至第三方站点的链接。您可以使用“许可使用内容”链接至第三方站点。第三方站点不由 Microsoft 控制，并且 Microsoft 不对任何第三方站点的内容、第三方站点包含的任何链接或第三方站点的任何更改或更新负责。Microsoft 不对从任何第三方站点收到的网站广播或任何其他形式的传输负责。Microsoft 仅为了您的方便向您提供这些至第三方站点的链接，并且包括任何链接并不暗示 Microsoft 认可相应的第三方站点。

有关全部“许可使用内容”的说明

“许可使用内容”中包括的“软件”、文档和其他材料可能包含不准确的技术内容或印刷错误。可能定期对内容进行修订。Microsoft 可随时在不提供通知的情况下对“许可使用内容”中规定的产品和（或）程序进行改进和（或）更改。

免责条款。除非另行说明，否则本《协议》提及的公司、产品、人物、特性和（或）数据均属虚构，并且无意以任何方式代表任何真实的个人、公司、产品或活动。

保留权利和所有权。Microsoft 保留未在本《协议》中明示授予您的一切权利。“许可使用内容”受著作权和其他知识产权法律及条约的保护。Microsoft 或其供应商拥有“许可使用内容”和其中组件的所有权、著作权和其他知识产权。

同意使用数据。您同意：Microsoft 及其关联公司可以收集和使用作为提供给您的产品支持服务的一部分而收集的与“许可使用内容”相关的技术信息（如果有）。Microsoft 可以将此信息仅用于改进我们的产品或为您提供定制的服务或技术，并且不会以能识别您身份的方式披露此信息。

额外软件/服务。除非我们随下列更新、增补、补充组件或基于 Internet 的服务组件一起提供其他应适用的条款，否则本《协议》适用于 Microsoft 在您获得“许可使用内容”的初始副本之日后可能提供给您的或为您准备的“许可使用内容”的更新、增补、补充组件或基于 Internet 的服务组件。就通过使用“许可使用内容”而提供给您的或为您准备的任何基于 Internet 的服务而言，Microsoft 保留停止这类服务的权利。

出口限制。您承认“软件”受美国出口法律管辖。您同意遵守所有适用于“软件”的国际法和国内法，其中包括美国出口管理条例以及由美国和其他国家（地区）政府颁发的最终用户、最终使用和目的地方面的限制。要了解详情，请访问 <http://www.microsoft.com/exporting/> 网站。

许可使用内容的转让。“许可使用内容”的原始最终用户可以将本《协议》和“许可使用内容”永久性地一次直接转让给另外一位最终用户，条件是该原始用户不得保留“许可使用内容”的任何副本，并且必须转让“许可使用内容”的所有部分（包括全部组件、介质及印刷材料、任何升级版本、各《许可协议》和（如果适用）正版标签）。这种转让不得为非直接转让，如以寄售方式转让。在转让之前，接收“许可使用内容”的最终用户必须同意遵守《协议》的

各项条款。如果“许可使用内容”是一个升级版本，任何转让都必须包括“许可使用内容”的所有先前版本。

终止。如果您未遵守本《协议》的各项条款和条件，在不损害其他权利的情况下，Microsoft 可终止本《协议》。如此类情况发生，您必须销毁“许可使用内容”的所有副本及其全部组成部分。

适用法律。本《协议》受中华人民共和国法律管辖。

责任限制。在适用法律所允许的最大范围内，无论损害赔偿是否在履行合约、出现疏忽或发生其他侵权行为时发生，Microsoft 和（或）其供应商绝不就因“许可使用内容”的任何组成部分或所有组成部分的使用或性能、因提供或未能提供服务、或因可从“许可使用内容”得到的信息而引起的或有关的任何特殊的、间接的、或特定的损害赔偿或任何损害赔偿（包括但不限于因营业中断，因使用、数据或利润的丧失，或因任何其他金钱上的损失而造成的损害赔偿）承担赔偿责任。在任何情况下，Microsoft 的全部责任以及您获得的惟一赔偿将限于为“许可使用内容”实际支付的款额或五美元（U.S.\$5.00）以两者中的较高款额为准；但是，如果您已经签订了一份 Microsoft 服务协议，Microsoft 对这类服务的全部责任将遵守该协议各项条款的规定。由于某些国家和地区不允许排除或限制责任，上述限制条款可能不适用于您。

全部协议；规定可分割性。本《协议》（包括随“许可使用内容”提供的本《协议》的任何补充条款或修正条款）是您与 Microsoft 之间就“许可使用内容”和支持服务（如果有）达成的全部协议，并且取代“许可使用内容”或本《协议》中所包含的任何其他标的之所有先前或同时存在的口头或书面的通信、建议和声明。如果任何 Microsoft 支持服务的政策或计划的条款与本《协议》的条款有冲突，以本《协议》的条款为准。如果本《协议》的任何条款被认定为作废、无效、不能执行或非法，其他条款应继续完全有效。

如果您对本《协议》有任何疑问，或者如果您由于某种原因希望与 Microsoft 联系，请使用“许可使用内容”中附带的地址信息与微软（中国）有限公司联系，或在<http://www.microsoft.com> 网站访问 Microsoft。

准则和定义

“许可使用内容”是一种专门设计的培训工具，供 Microsoft Certified Technical Education Center (Microsoft CTEC)、Microsoft Certified Partner (MCP)、Microsoft 认证培训讲师 (MCT)、IT Academy 计划成员和 Microsoft 可能随时以书面形式指定的其他机构使用。“许可使用内容”旨在使 Microsoft 的技术培训渠道能够向计算机专业人士提供系统、支持和开发培训课程。为了取得最佳成果，“许可使用内容”应该由 Microsoft 认证培训讲师 (MCT) 在课堂环境或在线学习环境中讲授。

Microsoft Official Curriculum (MOC)：由 Microsoft 开发的系列课程材料，用于提供 Microsoft 产品和技术的培训和解决方案。

Microsoft 认证培训讲师 (MCT)：具备必要的教学和技术能力并且由 Microsoft 认证为能够通过 Microsoft CTEC 讲授 Microsoft Official Curriculum 的个人。

Microsoft Certified Technical Education Center (Microsoft CTEC)：已经符合 Microsoft 对

指定其为下列场所的资格要求：（a）一处 Microsoft Certified Partner（MCP）营业点、和（b）一处提供 Microsoft CTEC 服务的任何场所。这些培训中心使用 MCT 向学生提供 MOC 课程培训。

Microsoft Certified Partner：已经符合被指定为 Microsoft Certified Partner 的资格要求的任何场所。

IT Academy 计划成员：已经符合被指定为 IT Academy 计划成员的资格要求的任何院校。

前 言

随着信息技术的发展，软件已经渗透到社会的每个角落，成为保障现代生活和社会正常运转的基本支撑之一。

软件的开发与交付是一个复杂的过程，一些困难始终困扰着软件开发组织。虽然企业在软件开发方面投入了大量的资金和人力，仍然无法获得他们所要求的系统。无论是在过去，还是在今天，都有许多这样失败的例子：软件无法按时交付，系统太昂贵，系统经常出故障，或者根本不满足客户的业务需求。

虽然软件开发仍然面临着诸多挑战，但是为了提高软件开发效率、降低成本、保证质量，软件开发组织仍然在不懈地努力。目前推出的符合行业标准的方法有 ISO9001、CMM (Capability Maturity Model)、极限编程 (eXtreme Programming, XP)、RUP (Rational Unified Process)、PMBOK (Project Management Body of Knowledge) 等。但是这些方法要么过于详细、面面俱到、文档太多，要么难度过大，要求从业人员水平很高，实施起来总是不能得心应手、切中要害。

软件开发组织需要一种经过实践证明的、经济的、易实施的软件开发方法，来指导软件开发和管理。微软解决方案框架 (Microsoft Solution Framework, MSF) 就是一种这样的实用方法。微软 MSF 是一种框架性方法，它吸取了微软多年的经验，抓住了软件开发中必然遇到的最核心问题，给出了精炼的模型和准则，指导软件开发组织以经济、易行、有效的方式进行软件开发，对各种类型的软件开发组织均有可借鉴之处。

微软 MSF 最早是在 1994 年提出的，当时它是微软咨询服务 (Microsoft Consulting Service, MCS) 的组成部分，用于指导客户采用 IT 解决方案去成功解决他们的业务问题。MSF 早期内容主要来自微软的产品开发组和 IT 部门的一些好的经验。后来，MSF 吸取了更广泛领域的经验，这些经验来自咨询专家、客户、全球范围的合作伙伴等。微软分析这些经验中可重复的成功因素，集成为 MSF 的概念、模型、原则，并运用于 MCS、IT 部门、合作伙伴和客户的实践之中。微软解决方案框架 MSF 凝聚了微软 20 多年的软件开发和管理的经验。

微软 MSF 认识到技术并不是解决方案成功的唯一决定因素。人员、过程、风险管理、项目管理和就绪管理在一个成功的 IT 项目中同样扮演着重要的角色。一个成功的项目小组应该有合理的组织结构，遵循有效的软件过程展开工作，主动地、不断地进行风险分析和管理，项目管理融入到每个小组成员的职责中，做好项目前的准备工作，调整技术方案，以适应不断变化的业务需求。这几点虽然非常难以达到，而恰恰就是许多 IT 项目失败的根源。

微软 MSF 的主要目标是帮助开发组织能够设计出满足业务过程 and 用户需求的解决方案；加速开发和缩短部署周期；降低技术的使用成本；提高计划事件成功的可能性；提高对未预测到事件处理的灵活性；设计出的解决方案具有可伸缩性和可靠性；改善核心 IT 的竞争力；取得短期结果的同时，管理好一个长期战略计划；有效地管理项目中的风险。

很多组织已经用 MSF 提高了它们解决方案的成功率和质量，而不需要采用任何行业标准

或者方法。但是作为一个框架, MSF 已经准备好为先前已经采纳行业标准和方法(例如: CMMI、极限编程、PMBOK、ISO9000 等)的组织提供对这些标准和方法的支持和提升, 并能够与之共存。MSF 的基础原理提供了一个很好的指示, 用来表明特定的方法是否能够与 MSF 兼容。那些具有类似原理的标准和方法一般都能很好地共存并相互补充。

软件工程方法一般是多学科的系统方法。由于信息技术发展迅速, 新术语、新概念、新知识层出不穷, 系统方法也在不断修正、完善和变更。过快的发展导致了不同时代的系统方法同时并存, 加上应用的多样性, 很难说原来的系统方法就一定不能用。这对于没有工程实践的读者, 学习起来是比较困难的。限于篇幅的限制, 本书不能详细介绍每个概念、术语、方法、规范和标准, 只能以 MSF 为主线, 补充总结性的预备知识, 以增强对 MSF 的理解。有关这些预备知识的详细论述, 可以参考相关书籍。

编者

2005 年 7 月

目 录

第 1 章 导言	1	3.1.1 项目类型	32
1.1 无所不在的软件	1	3.1.2 项目干系人	33
1.2 当今软件系统案例	1	3.1.3 项目获取过程	34
1.3 应用软件开发	2	3.2 软件度量与量度	35
1.3.1 系统与环境	2	3.2.1 产品度量	36
1.3.2 应用软件的范围	2	3.2.2 过程度量	38
1.3.3 应用软件的开活动	3	3.3 项目管理	39
1.4 软件工程	8	3.3.1 项目管理框架	40
1.4.1 工程评估与风险	9	3.3.2 各种管理简述	41
1.4.2 软件工程的特征	9	3.3.3 软件配置管理	46
1.4.3 软件工程定义	10	3.3.4 软件质量管理	49
1.5 本书的结构	10	3.4 本章小结	51
1.6 问题和讨论	11	3.5 问题和讨论	51
第 2 章 应用软件开发	12	第 4 章 实用软件工程方法 MSF	52
2.1 软件工程过程	12	4.1 MSF 起因与来源	52
2.1.1 软件生命周期过程	13	4.2 MSF 的基本观点	54
2.1.2 软件过程质量	14	4.3 MSF 概念	56
2.1.3 CMMI	17	4.3.1 两种模型	56
2.2 团队组织	20	4.3.2 三种管理准则	57
2.3 建模和实现的方法、技术	22	4.3.3 MSF 的基本原则	58
2.3.1 模型的表示方法	23	4.3.4 关键概念	58
2.3.2 建模(软件设计)的策略和方法	24	4.3.5 经实践检验正确的经验	59
2.3.3 软件实现方法和技术	25	4.4 本章小结	59
2.4 软件工具	26	4.5 问题和讨论	60
2.4.1 阶段相关工具	26	第 5 章 MSF 基础模型	61
2.4.2 非阶段性工具	27	5.1 MSF 小组模型	61
2.5 工作产品	28	5.1.1 小组成功的六个目标	61
2.6 基础设施	29	5.1.2 小组模型	62
2.7 本章小结	30	5.1.3 小组模型依据的 MSF 基本原则	64
2.8 问题和讨论	31	5.1.4 建组要注意的关键概念	66
第 3 章 软件项目	32	5.1.5 成功小组的经验	68
3.1 项目获取	32	5.1.6 角色的职能	70

5.1.7 小组模型的伸缩	80	7.2 MSF 就绪管理	150
5.1.8 小结	84	7.2.1 什么是就绪和就绪管理	150
5.2 MSF 过程模型	84	7.2.2 就绪过程概述	151
5.2.1 MSF 过程模型的特点	84	7.2.3 事前的就绪管理	152
5.2.2 MSF 过程模型依据的基本原则	90	7.2.4 遍及 IT 生命周期的就绪	152
5.2.3 过程模型的关键概念	91	7.2.5 就绪管理依据的 MSF 基本原则	153
5.2.4 过程模型的阶段和里程碑	93	7.2.6 MSF 就绪管理的关键概念与 成功经验	153
5.2.5 MSF 过程模型的成功经验	104	7.2.7 就绪过程步骤	154
5.3 本章小结	106	7.2.8 就绪与 MSF 小组模型和过程模型	160
5.4 问题和讨论	107	7.2.9 MSF 角色的技能要求	161
第 6 章 MSF 风险管理	108	7.2.10 创建就绪计划	163
6.1 风险管理基础	108	7.3 本章小结	163
6.1.1 风险源	108	7.4 问题和讨论	164
6.1.2 风险影响	109	第 8 章 MSF 企业体系结构	165
6.1.3 项目风险与管理	109	8.1 什么是体系结构和企业体系结构	165
6.2 风险管理依据的 MSF 基本原则	109	8.2 企业体系结构的需求	166
6.3 风险管理的关键概念	110	8.2.1 不良企业体系结构的后果	167
6.4 风险管理过程	112	8.2.2 好的企业体系结构的结果	167
6.4.1 风险标识	113	8.2.3 需要设计企业体系结构的信号	168
6.4.2 分析和确定风险优先级	117	8.2.4 开发企业体系结构的通常障碍	168
6.4.3 风险计划和调度	123	8.3 企业体系结构模型	169
6.4.4 风险跟踪和报告	127	8.4 MSF 企业体系结构的规划和分析	170
6.4.5 风险控制	128	8.4.1 MSF 企业体系结构的规划	170
6.4.6 从风险中学习	130	8.4.2 MSF 企业体系结构的分析	170
6.5 有关风险管理的几个问题	132	8.5 MSF 企业体系结构过程	172
6.5.1 写好风险管理计划	132	8.6 企业体系结构项目的过程	174
6.5.2 在项目的生存周期中集成风险管理	133	8.6.1 构思阶段	175
6.5.3 在企业中的风险管理	133	8.6.2 计划阶段	175
6.6 本章小结	135	8.6.3 开发阶段	176
6.7 问题和讨论	135	8.6.4 稳定阶段	177
第 7 章 MSF 项目管理和就绪管理	136	8.7 企业体系结构项目的风险分析	177
7.1 MSF 项目管理准则	136	8.7.1 传统的企业体系结构方法的风险	177
7.1.1 MSF 项目管理域	136	8.7.2 MSF 企业体系结构的风险	178
7.1.2 MSF 项目管理的特点	137	8.8 企业体系结构项目的小组模型	178
7.1.3 MSF 项目管理依据的 MSF 基本原则	141	8.9 企业体系结构的最佳实践	179
7.1.4 有关项目管理的几点建议	142	8.10 本章小结	180
7.1.5 制作进度表的建议	149	8.11 问题和讨论	181

第 9 章 项目构思阶段	182	10.6 计划阶段小组角色的职责	218
9.1 构思阶段的活动	182	10.7 本章小结	219
9.1.1 协调与干系人的关系	182	10.8 问题和讨论	220
9.1.2 项目团队的组建和准备	183	第 11 章 项目开发阶段	221
9.1.3 定义解决方案	185	11.1 开发阶段的活动	221
9.1.4 确定解决方案范围	187	11.1.1 开发技术基础架构	221
9.1.5 建立配置和变更管理	189	11.1.2 解决方案技术基础架构的验证	222
9.2 构思阶段的成功的标准	189	11.1.3 内部发布	223
9.3 构思阶段的文档	190	11.1.4 每日构造	224
9.3.1 远景/范围文档	190	11.1.5 代码审核	225
9.3.2 初始风险评估文档	193	11.1.6 构造用户体验交付物	227
9.3.3 项目的组织结构文档	193	11.1.7 构造运营文档	227
9.3.4 远景/范围认可里程碑及中间 里程碑	194	11.1.8 测试解决方案	227
9.4 构思阶段小组角色的职责	195	11.1.9 缺陷管理	230
9.5 本章小结	196	11.2 零缺点理念	233
9.6 问题和讨论	196	11.2.1 为什么要树立零缺点理念	233
第 10 章 项目计划阶段	197	11.2.2 零缺点理念的指导原则	234
10.1 计划阶段的活动	198	11.2.3 实施零缺点理念的若干措施	234
10.1.1 技术验证	198	11.3 开发阶段成功的标准	235
10.1.2 解决方案的设计	198	11.4 范围完成里程碑的交付物	235
10.1.3 创建主项目计划	205	11.5 范围完成里程碑和中间里程碑	236
10.1.4 创建主项目进度	208	11.5.1 范围完成里程碑	236
10.1.5 建立开发和测试环境	210	11.5.2 MSF 建议的中间里程碑	236
10.2 调度原理	210	11.6 开发阶段小组角色的职责	237
10.2.1 估算工作量由底向上	211	11.7 本章小结	238
10.2.2 上市日期坚定不移的意识	211	11.8 问题和讨论	238
10.2.3 风险驱动的调度	211	第 12 章 项目稳定阶段	239
10.2.4 为不定的未来调度	211	12.1 稳定阶段的活动	240
10.3 计划阶段的成功标准	213	12.1.1 稳定阶段的测试	240
10.4 计划阶段的文档	213	12.1.2 缺陷消除过程	240
10.4.1 功能规范文档	214	12.1.3 用户验收测试	241
10.4.2 主项目计划	216	12.1.4 指定发布候选版本	241
10.4.3 主项目进度表	216	12.1.5 实施投产前测试	241
10.4.4 主风险评估文档	217	12.1.6 试运行	243
10.5 计划认可里程碑及中间里程碑	217	12.2 稳定阶段的成功标准	245
10.5.1 项目计划认可里程碑	217	12.3 发布就绪里程碑的工作产品	245
10.5.2 计划阶段的中间里程碑	217	12.4 发布里程碑和中间里程碑	245
		12.4.1 发布就绪里程碑	245

12.4.2 建议的中间里程碑.....	246	13.5 部署阶段小组角色的职责.....	257
12.5 稳定阶段小组角色的职责.....	247	13.6 本章小结.....	258
12.6 本章小结.....	247	13.7 问题和讨论.....	258
12.7 问题和讨论.....	248	第 14 章 个案研究——微软如何使用	
第 13 章 项目部署阶段.....	249	MSF 做软件开发.....	259
13.1 概念与原理.....	250	14.1 SharePoint Team Services 简介.....	259
13.1.1 基础架构部署.....	250	14.1.1 使用团队 Web 站点.....	259
13.1.2 基础架构部署的变更管理.....	250	14.1.2 产品功能特性.....	260
13.1.3 变更的来源.....	250	14.2 SharePoint 团队.....	262
13.1.4 管理变更的策略.....	250	14.2.1 微软的惯例和原则.....	262
13.1.5 在部署期间变更控制.....	251	14.2.2 团队的组成和汇报结构.....	263
13.2 部署阶段的主要活动.....	251	14.2.3 成员的职责.....	263
13.2.1 部署核心组件.....	251	14.3 产品周期.....	264
13.2.2 部署各个站点.....	252	14.3.1 产品构思阶段和计划阶段.....	264
13.2.3 部署的解决方案稳定.....	255	14.3.2 产品开发阶段和稳定阶段.....	265
13.2.4 转移到运营和支持.....	255	14.3.3 第二次发布的构思和计划阶段.....	268
13.2.5 项目完成.....	255	14.3.4 第二次发布的开发和稳定阶段.....	268
13.3 部署阶段的成功标准.....	256	14.3.5 发布 Beta1 和 Beta2.....	268
13.3.1 部署阶段的工作产品.....	256	14.3.6 最终发布版本.....	268
13.4 部署阶段部署完成里程碑和中间 里程碑.....	257	14.4 产品开发总结.....	269
13.4.1 项目部署阶段部署完成里程碑.....	257	14.5 人员管理.....	270
13.4.2 项目部署阶段的中间里程碑.....	257	14.6 本章小结.....	275
		词汇表.....	276

第1章 导 言

1.1 无所不在的软件

随着信息技术的发展,软件已经渗透到社会的每个角落,成为保障现代生活和社会正常运转的基本支撑之一。

在日常生活中,我们使用的手机、电冰箱、微波炉、洗衣机、电视机、DVD 播放机、电子游戏机等,里面都有软件。在社会活动中,在银行存钱、取钱需要软件的支持,购买飞机票需要软件支持,酒店住宿需要软件支持,饭店就餐打印菜单需要软件支持,超市购物打印清单和付款需要软件支持,缴纳水费、电费、手机费需要软件支持,用烤箱烤个面包也需要软件支持,等等。软件已经进入我们每个现代人的生活。

软件不但在为每个人的日常生活服务,而且已成为社会正常运转的基本保证。设想一下,银行的软件发生故障,把数据库中的数据全部删除,几十秒内可以让几百万人身无分文,其后果不亚于一场社会革命,即便是有数据备份,也会引起社会动荡;如果空中交通指挥系统的软件发生了故障,天上的飞机就无法降落,地上的飞机无法起飞;电信公司的电话交换机的软件出了故障,我们就无法和外界进行联络;等等。软件的可靠性和正确性关系到整个社会的正常运转。

软件对于企业也越来越重要,它已成为企业的核心竞争力之一。例如,银行 A 可以为一位客户开 1 万个账号,每个账号存 100 元钱,存期为 10 年,利息为 3%,一旦第 2 年,这位客户需要用 10 万元钱,那么它动用的 1 千个账号的钱的利息变成 0.5%,而其他账号利率不变;而银行 B 的软件无法提供这种服务,那么客户就有可能把钱存入银行 A,银行 A 因为其软件能够提供更好的功能,从而获得了竞争优势。再例如,一个超市的货物的摆放也很有学问,为什么把尿布和啤酒放在一起呢?因为超市的软件统计出买尿布的人买啤酒的概率很大,所以把它们放在一起。这个超市因为有更好的软件,销售量超过了它的竞争对手。

正是由于软件的需求越来越多、越来越重要,软件从业人员应该学习好的开发方法,来提高软件开发的效率和质量,使软件成为推动社会进步的重要力量。

1.2 当今软件系统案例

有手机的人,几乎人人会发短信息。2003 年春节,从除夕到初七,中国移动的短信息量达到 60 亿条,中国联通的短信息量达到 10.35 亿条。这样,2003 年春节期间,全国的短信息量突破 70 亿条,创下新的历史纪录。

实际上,短信息的发送和接收是通过复杂的软件系统处理完成的。电信运营商对短信息处理系统有着众多的要求。例如,在功能方面,要求自己用户的手机之间可以相互收发短信息,

自己用户的手机可以向其他运营商用户的手机收发短信息，可以通过互联网络向手机发送短信息，手机可以向电子邮件信箱发短信息，可以为未连线的目标用户存储短信息，可以把短信息转换成语音，可以允许网络内容提供商（例如新浪、搜狐）的系统接入，可以进行短信息记费，等等；在性能方面，要求每秒处理 3 000 条短信息，发送的短信息 30s 之内到达目标等；还有系统的可靠性、安全性、可扩展性、可扩充性、可维护性等方面的要求。

这就是和日常生活有关的应用软件系统，它的结构复杂，各模块之间通过各种协议进行数据通信，需要满足客户种类繁多的需求，需要有极高的性能和可靠性。

1.3 应用软件开发

前面描述了一个应用系统的需求，开发这样的应用软件并不是件简单的事。本书就是讨论开发这类软件的解决方案。为此，先廓清一些基本概念和术语。本节针对开发中必需的活动（即最主要的工作是什么）进行介绍。

1.3.1 系统与环境

开发软件的目的是为了建造应用（application）程序。一个应用程序能运行起来并实施其功能，则说明它不单单是一个软件，而是一个系统（system）。应用程序只是应用系统中的一个组件（component），其他组件还包括：其他软件（系统软件、基础设施、工具软件），硬件（计算机、外设、通信设备）；数据（持久的、临时获取和输出的），文档（说明性的、归档性的、使用指南、培训材料）；人（使用者、操作者、维护人员）。这是在“基于计算机的应用系统”这个名目下的真实系统。

所谓“系统”，就是在某个名目下它所涉及的所有组件的集合，且成分间有相互关联的关系。系统之内称为成分或元素，系统之外则称为环境（environment）或上下文（context），系统内外的联系则称为接口（interface）。

系统的概念我们并不陌生，我们也常用到“文教系统”、“公交系统”、“军工系统”、“后勤系统”等。由于软件系统之间关系比较复杂，必须明确界定各系统的范围（scope），以便开发工作始终聚焦在正确的目标上。

应用软件只是应用系统的一个组件。如果单单开发、研究、讨论它，也可以说成是“应用软件系统”，它和其他成分的联系都叫做接口。同样，“操作系统”是保证应用程序能在计算机上运行的软件系统，“数据库系统”是持久存储数据的保证，是数据使用能够共享、一致的软件系统。这些在相应的课程中已经很熟悉了。

1.3.2 应用软件的范围

开发应用软件是为了获取支持某种业务的自动化系统。业务领域没有解决问题的软件，才去开发。然而，在计算机应用的早期，计算平台只提供操作系统和少量库函数、数据包以及必备的工具，如编译器/解释器、连接/加载程序、简单的调试程序等。所有与应用有关的程序都要开发，从用户界面、应用逻辑、数据存储，直到能与操作系统接口。所以给人的错觉是：计

计算机应用就是编程，程序员必须熟悉许多技术上的实现细节，如读/写文件、分配/释放内存、数据格式转换、外部通信等，并把它们编写到应用程序中。

如今，软件制作商做出了功能日益强大的操作系统，使得使用、操纵程序或数据极为简便，程序员一般只需给出命令和选择参数，不用编程即可实现。而且还将原来应用程序中具有共性的功能做成基础设施（*infrastructure*）软件，如各种中间件（远程过程调用、消息队列、事务监控、对象代理、数据库等）。此外，还有大量自动生成工具，如用户界面、文档、数据库访问等。真正需要编程开发的应用软件越来越专门化（针对应用特殊的问题），开发的含义也发生了变化。如果用工具自动生成，开发的含义也就不是编程了，如在用户界面工具上开发某个应用的用户界面。程序员只需设计式样，选定成分，给出参数，一句程序也不用编。应用面比较通用的软件或市售商用软件（*Commercial Off-The Shelf, COTS*），例如 *Microsoft Word*、*Oracle* 等，买来安装上就可以使用。

随着软件技术的发展，基于组件（*component-based*）的应用正在蓬勃发展。各行各业组件制造商大量制造组件，并提供服务，这使得应用软件的开发更加简化。它只面向求解的问题，设计出解决问题的方案，即用接口描述应用的体系结构，开发工作就完成了，每个接口的功能实现可以去租赁组件服务，使用完了付费即可。这虽然是一个十分美好的前景，但目前还不能成为应用开发的主流，还存在经济、法律、安全等有待规范的许多问题。然而，“只用服务不做传统意义的编程”的趋势是肯定的。

1.3.3 应用软件开发活动

应用软件开发活动其实和任何软件开发活动是一样的，都是创建能在机器上运行的本机目标代码系统，使软件运行后表现的行为达到预期的要求，而开发者使用的是某种高级程序设计语言，经编译、连接，装入内存并运行，这是众所周知的。正是由于高级编程语言是人和机器都能接受的表示法（*notation*），所以针对某个问题的所有源代码开发完成后才算有了软件。测试、修改、保存、移植都以源代码为准（除了单板机测控领域仍用汇编语言之外）。因此程序编码或软件构造是软件开发中的一个重要活动。

（1）软件构造

用高级编程语言编写源代码也并不容易，因为每种高级语言都有自己的术语，其基本元素和构造规则（语法与语义）总会趋向支持某种范型（*paradigm*）的程序结构。例如，*Pascal*、*C* 语言，支持结构化过程式编程；*Ada* 支持对象/包式范型；*Smalltalk*、*C++* 支持面向对象范型；*Java*、*.NET* 支持与平台无关的组件范型。

所谓范型就是程序结构规范的形式，它较易影响程序员的思维。例如，对于某一个给定的问题，*C* 语言程序员可能首先想到的是它有哪些功能要求可编成函数（即模块）、每个模块有哪些输入/输出以及实现模块要用什么算法，并且为此设计局部的和全局的数据结构，编写声明，以变量表达式写出语句，以控制语句描述算法，最后写一个主程序来调用各函数，再查看一下总的输入/输出安排在哪里，这就形成了层层调用的模块式范型的源程序。

但是，对于面向对象程序员而言，他可能首先想到的这个问题中有哪些对象、每个对象要有什么属性和方法才能与其他对象交互，以满足这个问题对对象集的行为要求，也就是完成的