网络分布计算与软件工程

（第二版）

冯玉琳 黄 涛 金蓓弘 编著

科学出版社

网络分布计算与软件工程

(第二版)

冯玉琳　黄　涛　金蓓弘　编著

科学出版社
北京

序

冯玉琳教授常年研究分布式计算和软件工程，带领一支团队做出了突出的创新性贡献，并将其研究成果应用于国民经济重要领域，推动了学科的建设和发展。现在以其研究成果、心得、体会，写成著作与学界共享，这是水到渠成，非常自然的事。

尽管这本书的骨架是围绕软件和软件工程来写的，但是从内容的份量以及从作者多年来的研究实践来说，我觉得更大的特色还是在网络分布式计算方面，作者研究的是面向分布式计算的软件，或者说是实现为软件形式的分布式计算，从书名也能得到这样的印象。所以，我就围绕“分布式计算”来说一说自己的印象和感受。

在我们的印象中，分布式计算与其他一些大规模计算概念，如大规模并行计算、向量计算、网格计算等相比，尽管在原理上有较大差别，但其本意都是为了提高计算效率，解决复杂问题。所以，拿起冯玉琳教授领衔编写的书稿，回想起我们接触这些概念的岁岁月月，不禁浮想联翩。我首先想起 20 世纪 80 年代初期，在由慈云桂院士和陈火旺教授（后为院士）主持的一次国际研讨会上，当时还是青年讲师的范植华正在侃侃而谈，介绍他为银河巨型机研制的 Fortran 向量化软件及其数学原理。Fortran 程序中可并行执行成分的自动识别是发挥这种巨型机优势的关键。他的工作的主要难点是解决循环语句编译的向量化问题。范植华的工作可算是为中国的 Cray 巨型计算机插上了翅膀。不过在参加这个会以后很长时间内，我再没有接触到国内有关并行编译的工作，直到本世纪初在复旦大学上海市智能信息处理重点实验室的学术交流会上，才了解到这里也有一片并行编译的天地，那就是朱传琪教授挂帅的团队。这位曾经全过程参加美国伊里诺大学 CEDAR 大型并行计算机系统研制的归国学者，主持研制了并行化编译系统，通过标准测试程序分析，其性能优于国际上许多同类并行编译系统。他的工作成为一个亮点，与重点实验室多数老师的研究方向不一样。

接着，我又想起改革开放不久，在杨芙清院士于北京大学芍园召开的一次座谈会上，上海软件中心的朱三元教授娓娓而谈，向大家介绍国际 OMG 组织制定的分布式对象编程标准 CORBA 和微软刚发布不久的分布式对象编程技术 DCOM，这是我第一次近距离接触现代的分布式程序设计。听后大家感觉这新的发展形势真是催人警醒。我也想起在南京大学国家重点实验室的学术委员会会议上，观看孙钟秀院士为我们做的分布式程序演示：一辆汽车从一台机器的屏幕上开走，又马上在另一台机器的屏幕上显示出来，引起大家的浓厚兴趣。孙院士在分布式操作系统和分布式程序设计语言方面的研究很早就广为人知。在差不多同一时间，中科院数学所同样经历了一次分布式的洗礼。周龙骧教授派人远涉重洋，到德国斯图加特大学 Neuhold 教授的实验室去参加当时颇为先进的分布式数据库管理系统 POREL 的研制和移植工作。POREL 系统后来被带回国内，移植到数学所的局域网上，这可能是国内的第一个分布式数据库管理系统。我记得曾参加过中科院软件所组织的一次验收会，验收项目就是冯玉琳教授主持开发的网络软件基础架构平台，当时国内第一家能同时提供面向对象、面向消息和面向事务等多种应用模式的软

件基础设施，国家对这个项目给予了很大支持，这也是中科院为国家做的一项重要贡献。本书第8章对该项目的特点和优势有详细说明。

一个重要的转折点可能出现在21世纪初。在中科院计算所组织的一次研讨会上，我听到了夏培肃院士神情严肃地介绍美国的高性能计算(HPC)计划，并且告诫听众说，如不迎头赶上，我们将要被抛在时代的脚步后面。此后不久，在北京我有机会参加了中科院计算所主持的《中国国家网格软件研究与开发》"863"重大专项的验收。他们开发的GOS软件已经在我国十多个领域推广应用。眼见一项大型工程的成果投入使用，我们中国也有了全国性的计算网格，可算是对夏先生呼吁的一个回应。近年来，分布式计算软件有了长足的进步，并且从实验室进入了产业，发挥着重大作用，同时也提出了众多挑战，其中一个重大挑战是：分布式数据在海量数据的计算环境下如何能够高效运行？对此，我因参加了复旦大学周傲英教授主持的多次数据管理研讨会而获益匪浅。在会上了解到，面向大型问题和海量分布数据的Map Reduce计算框架已经相当成熟，不但形成专利，而且被众多行业采用，成为大规模数据分布式处理的事实标准。基于Map Reduce的Hadoop计算平台也被广泛用于云计算等重要领域，说明分布式技术的前进脚步正在加快。至于应用，前不久，我在北京香山、上海华东师范大学等多个学术会议上，听到何积丰院士畅谈物联网在我国的前景以及即将启动的物联网"973"项目，深切体会到海量分布式计算已经在我国的社会和经济建设中落地生根，其开花结果之时指日可待。最近，更惊人的新闻来自《参考消息》，据报道，国外已经开始把卫星联网实现了"分布式卫星计算"，分布式理念已经深入太空。想起天天有"分布式"在头上盯着我们，真是"别有一种滋味在心头"。所以说，我们不能把分布式计算仅仅看成是一个抽象概念，它就在我们身旁，就在我们周围，它渗透到我们的日常生活中，它关系到我们社会的兴旺和国家的强盛。凡是知道计算机重要性的人就应该知道分布式计算的重要性。

令人高兴的是，我国计算机科学家在分布式计算和分布式软件的创新方面也有自己的贡献。除了冯玉琳教授主持研制的"网驰"开放网络计算环境以外，就我所知还有另外一项重要的工作。21世纪初的某日，在北京大学的某间教室里，来自各协作单位的老师们正在热烈讨论申报"973"的题目，其中包括北京大学的梅宏教授和南京大学的吕建教授。当时有三个关键词：软件工程、因特网和Agent。可是怎样把它们"有机地"(中国人喜欢这样说)联系起来呢？在一阵激烈的头脑风暴之后，一个新词"网构软件"浮出水面。它的思想和传统的软件开发相反：不是从需求分析开始一步一步自上向下进行，而是从网上的"软件素材"开始一步一步由下向上搭建用户所需的软件，实际上是一种动态开放的分布式软件工程。循着这个思路，他们在"973"项目的支持下已经进行了多年的深入研究，可能是出于不想显得太"标新立异"，第一个"973"的题目叫："Internet环境下基于Agent的软件中间件理论和方法研究"，没有直接点出网构软件的名称。由于看到社会上对此概念的逐步增加的认可，在后续的"973"的题目中就公开打出了网构软件的旗号。举一个例子，IBM公司每年向社会发布计算机软件研究新趋势的展望，在不久前的一次会议上，他们基本上接受了网构软件的思想，同意将它加入要发布的报告中，只是因为一个与会者建议用另一个名词来表达同一个思想，才使得报告中没有出现"网构软件"几个字。

中国的计算机科学家和工程师们在分布式计算领域取得的成绩正在IT产业中发挥作用，这一点已经在国际上被观察到。一份来自国外的研究报告说：中国正在分布式计算的市场中发挥领跑作用，大大领先于印度、巴西和东欧。他们分析了中国、印度、巴西和东欧的400家开发企

业后声称：中国已经采用分布式技术开发软件的企业比印度多出约34%，比巴西和东欧要多出80%。这份报告从侧面强化了本书的价值。

我不是分布式计算的专家，也写不出包括深刻的分布式计算发展趋势的序言，只是把我这几十年来看到的、听到的点点滴滴写下来。可以说在分布式计算方面，我是被形势的发展推着走的，被我国计算机界对分布式计算一浪又一浪的研究热潮冲着走的。在孔老夫子家里当小时工，久而久之，总能受到一些“之乎者也”的耳濡目染。我就是这样写下了以上这些话。需要声明，“这些话”绝对不能全面反映国内分布式计算研究的所有方面，我了解的情况太少，太不全面了。通过作序，一方面是回忆历史，盘点一下我认识分布式计算的过程；另一方面，最主要的，也是我接触这本书的开始。冯玉琳教授给我提供了一个学习分布式计算的机会。这是一本非常好的教程和参考书，我不仅向读者积极推荐，而且首先自己就准备好好学一学，给自己补一补缺了多年的课。

陆汝钤
中国科学院数学与系统科学研究院
2011年3月

前　言

本书第一版发行后，人们经常提出这样的问题：此书为什么要将网络分布计算与软件工程这两个主题放在一起？业界关于软件工程的著作不下数十种，同时也不乏分布式计算的专著和译著，但它们都是各自独立地在讲述各自的内容。本书作者在长期追本溯源的学术生涯中深深感悟到世界变了，《人月神话》这一经典著作问世 30 多年来，软件技术已经发生了很多变化。随着大规模网络应用软件的出现和普及，分布式系统已经成为软件发展的主流。这就需要一部从不同视野出发的关于软件工程的著作。本书内容兼顾了过去和现在软件技术的发展，而标题的改变则标志着重点的调整，表示本书是在更广泛的意义上讲述网络分布环境下的软件工程技术问题。

虽然传统软件工程的一般原理和方法仍然是指导软件工程实践的有效良方，但是它已经很难适应由网络化软件所带来的各种变化，例如：

- 软件系统的规模越来越大，复杂性越来越高，软件体系结构已发展到分布式的三层或多层结构。软件系统中各个计算单元分布在不同的站点机上，各自独立地运行在异构的操作系统平台上协同完成计算任务。
- 软件系统要处理大量并发的分布事务，可靠性要求更加突出；要能经受大规模并发用户访问的考验，性能要求更高；软件需求的变化更快，要求系统能灵活地快速响应客户需求的变化。
- 分布式系统的发展催生了组件化软件工程方法。分布式系统是如此复杂，即使是从事应用开发的人员也要对分布式系统的结构特点和运行环境有足够的理解和掌握，并能有效地运用中间件平台提供的各种工具实施基于组件的软件工程开发、部署和测试。

软件工程的发展必须面对和解决出现的这些新问题，本书就是为此目的而撰写的，其中内容的取舍和组织代表了作者的学术观点。本书在现有软件工程概念的基础上，重点介绍网络分布计算原理和分布式系统架构，以及与之相适应的组件化软件工程方法。全书内容深入浅出，分为 4 个部分，共 10 章。第 1 部分是软件工程概论，包括第 1～3 章，介绍软件工程的基本概念和方法，以及软件工程技术发展中的两个重要方面，即软件系统建模和软件体系结构。第 2 部分是网络分布计算的原理，包括第 4～6 章，讲述网络分布计算的基本特征，并从基础模型和通信、进程、并发控制、寻址定位、事务和容错等 6 个方面介绍分布式系统的原理和技术，还介绍了与分布式系统设计有关的几类常用算法。第 3 部分包括第 7、8 章，结合典型系统案例，分别介绍了各种不同结构特征的分布式系统，以及中科院软件所研究开发的网络软件基础架构平台。第 4 部分介绍适用于网络分布计算环境的新范型和新方法，包括第 9、10 章。第 9 章讲解组件化软件工程开发，阐述基于组件的软件开发方法，以及软件模式和软件框架的应用；第 10 章是面向服务的计算，主要介绍 Web 服务计算，还对服务计算技术的最新发展进行了展望。

本书内容的组织结构如图 1 所示，可为阅读本书提供一个推荐的导航路径。

本书专门为计算机科学的大学高年级学生或研究生课程而编写，可作为软件工程课的高级教程；适当增加系统设计的练习，亦可作为分布式系统的专业课教程。本书要求读者具有面向对象编程、操作系统及计算机网络的基本知识，也可作为从事软件研究和应用开发的广大软件技术人员基础性的专业参考书。

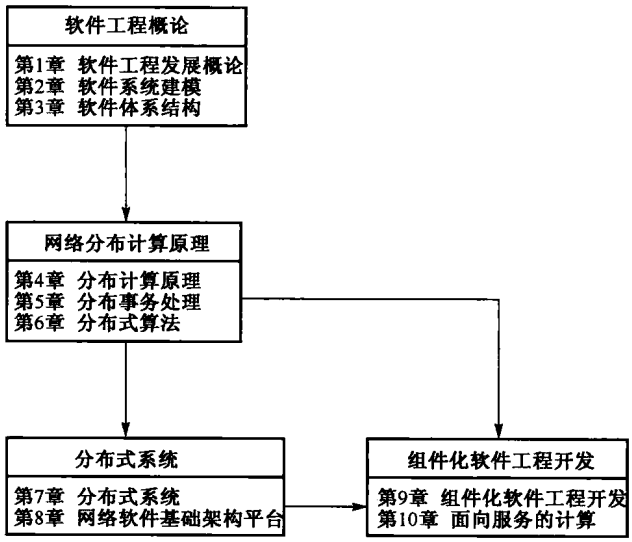

图 1　全书组织结构

本书对第一版进行了重大修订，在保留原书的组织结构基础上，除了对第一版中的错误进行校正外，各章内容都有不同程度的修改。此外，还增加了第 8 章和第 10 章，以反映分布式系统的最新研究成果和发展方向。第一版的第 8 章在本书中为第 9 章，并已作了较大的修改。本书部分章节内容来自作者所在课题组的研究成果，并直接使用了丁柯、范国闯、张波、张文博、张昕、宋靖宇、万淑超、丁晓宁、李磊和刘国梁等博士论文的部分内容。全书最后列出了供延伸阅读和参考的文献。作者在此对书中引用和参考的所有文献的原作者表示感谢。

本书第 1、4、5、6、8、10 章由冯玉琳撰写，第 2、3 章由金蓓弘撰写，第 7、9 章由黄涛撰写。全书由冯玉琳修改和定稿。在本书写作过程中，得到了中国科学院软件研究所软件工程技术研发中心的大力支持。钟华和魏峻参与了第 8 章的组织和讨论，并提供了许多第一手资料。叶丹、王伟、高楚舒、王焘、伍晓泉和黄翔等对本书初稿进行了认真的审校；刘玲玲和明路承担了本书的录入和绘图工作。没有他们的努力，本书的顺利完稿是不可能的。最后，还要感谢科学出版社的编辑，他们为本书的成功出版付出了辛勤劳动。

本书的创作和修订得到了国家自然科学基金和国家基础研究发展计划项目的资助。

软件技术发展日新月异，本书虽经修订再版，仍难免出现疏漏错误，恳请读者不吝批评指正。

作　者
中国科学院软件研究所
2010 年 8 月于北京

目　录

第 1 章　软件工程发展概论

软件工程是计算机学科的一个独立分支，要求软件开发必须按照工程化的原理和方法来组织和实施。由于软件产品本身的特殊性，特别是对于新出现的大规模网络应用软件，软件体系结构和运行环境发生了很大变化，传统的软件工程方法和技术已不能适应由于网络化所带来的新需求。本章在简要总结软件工程基本概念和一般原理的基础上，重点讲述网络环境下软件工程所面临的新问题，以及传统软件工程的观念应作什么样的改变，以适应这种变化了的新形势。本章内容包含著作[冯玉琳，1992，1998]中的部分正文和图片。

1.1　软件工程的目标

软件工程是研究大规模软件制造的方法、工具和过程的工程科学。由于软件规模的庞大，因此，软件工程的成败并非主要取决于个人的聪明才智，而是要求参与工程的每一位开发者都能按照软件工程的规范和过程协同工作。本节在分析软件工程面临问题的基础上，讲述软件工程的基本概念，包括软件工程的基本要素、软件工程的目标及软件工程生命周期等。

1.1.1　软件工程要素

数学定理的发明和推导体现了数学家的高度智慧，其中严密的数学推理和奥妙常为人们所叹服。计算机软件也是人们高度知识型劳动的成果，它蕴涵的逻辑和计算体现了人们对外部世界的理解和认识。对庞大的复杂软件系统的分析和设计，必须遵循一定的科学原理和方法，并要求参与者充分发挥其智慧。

软件工程是研究大规模软件制造的方法、工具和过程的工程科学。软件工程指导计算机软件的开发、运行和维护，采用工程化的理念、方法和技术来开发和使用软件，并对软件开发过程进行有效的管理。

首先，软件工程是针对大规模软件制造的。何谓软件规模？一个最基本的度量参数是源代码行(LOC)。在学校内用于教学程序设计或软件工程课程的实例，代码一般都在 5000 行之内，这只能算是小程序设计。按照通常的软件规模大小的尺度，1～5 万行代码的软件是小规模软件，5～10 万行的软件是中规模软件，10～100 万行的软件是大规模软件，100 万行以上是超大规模软件。随着软件应用复杂性的增加，软件规模越来越大，度量软件规模的尺度也随之扩大。例如，超大规模软件的定义范围，代码已从 100 万行增加到 1000 万行。

数学证明主要依靠数学家个人智慧的发挥，而软件规模如此庞大，只靠几个人的力量显然是不可能完成的。一种大规模的复杂软件，可能需要数百人，甚至数千人在几年的时间内协同工作才能完成。假设一个人一年可以开发出一万行的源代码，按照同样的工作效率，一种 100 万行源代码规模的软件，是否集中 100 人的力量工作一年就可以完成呢？答案是否定的。因为软件工作量中的人数与时间不是简单累加计算的，当软件规模增加 100 倍，软件复杂程度的增加倍数将远远超过 100 倍。软件内部是互相关联的，软件人员需要互相交流，以保证很多人完成的任务集合在一起能够成为一种高质量的大型软件系统，这确实是一个极为复杂而困难的问题。这

不仅涉及许多技术问题，如分析方法、设计技术、错误检测、版本控制等，而且还必须有严格和科学的过程管理。

考虑到研制软件同研制机器或建造楼房的过程有许多相似之处，所以可以参照机械工程或电气工程中的一些概念来进行软件研制，用“工程化”的思想作为指导来解决软件研制过程中面临的困难和混乱。然而，软件工程作为一门新的工程学科，还遵循着自身特有的工程原理和方法。软件是抽象的逻辑性的产品，不是实物性的产品。研制和维护软件的过程非常难以控制。虽然软件工程已发展了几十年的时间，但到目前为止，软件工程仍然不能像机械工程或电气工程那样建立起统一的设计标准和足够的可靠性保证，这就决定了软件的研制和开发较之其他工程学科的项目要困难得多。

软件工程的内容包括三个要素，即方法、工具和过程。

- 软件工程方法为软件开发提供“如何开发”的原理和技术。它包含了多方面的任务，如项目计划与评估、系统需求分析、软件体系结构、算法设计、编码、测试和维护等。软件的开发和维护是一种复杂的活动，必须用软件工程方法作为指导才能进行。不同的软件方法会导致不同的软件过程。
- 软件工具为软件工程方法提供自动或半自动的支撑。方法和工具往往是同一问题的两个不同方面：方法是工具研制的先导，工具则是方法的实在体现。将各种软件工具集成，连同软件运行基础设施一起形成软件工程环境。
- 软件过程是将软件工程方法和工具综合运用以科学地进行软件的开发和实施，包括软件方法使用的选择、要求交付的文档资料、为保证质量所需要的过程管理，以及软件开发各阶段要求完成的工作情况等。

软件工程的目标就在于研究科学的软件工程原理和方法，并开发与之相适应的软件工具，从技术上和管理上保证软件工程项目实施的成功，力求用较少的投资在规定的工程期限内完成高质量的软件。

1.1.2 软件工程面临的问题

从20世纪60年代末开始，人们就在讲“软件危机”，而后一直都在为解决软件危机面临的困难问题而进行坚持不懈的努力。软件工程作为一个学科方向，越来越受到人们的重视。《人月神话》[Brooks,1995]出版30多年来，很多技术已发生了重要变化。随着大规模网络应用软件的出现和普及，软件的体系结构和运行环境发生了很大变化，软件系统的规模越来越大，复杂性越来越高，软件体系结构已发展到分布式的3层或多层结构。软件系统中各个计算单元分布在不同的站点机，各自独立地运行在异构的操作系统平台上，协同完成计算任务。分布式系统已经成为软件发展的主流。传统的软件工程方法和技术因难以适应网络化所带来的新需求而面临严重的挑战。

软件可靠性是衡量软件质量的一个重要指标。软件可靠性是指软件系统能否在指定的环境条件下正确运行并实现所期望的结果。软件错误的后果十分严重，如医疗软件的错误可能危及病人生命；银行软件的错误会造成金融混乱；航天发射软件中的错误会使火箭试验失败。通常有40%左右的软件开发代价要花在软件编码完成之后的测试和排错上，但即便如此，也不能保证经测试的软件就没有错误。例如，在网络上运行软件时，软件的不确定性增加，分布式软件系统的测试更是一种复杂的工作，因此软件系统花在测试上的代价非常之大。

许多网络应用软件还要经受大规模并发用户的访问的考验。虽然在严格的软件测试过程中，软件运行已经达到了满意的可靠程度，可是在网络运行的实际环境下，一旦“瞬间”同时访问

的大量并发用户的请求超过预期，软件系统就可能“崩溃”。近些年来，一些新开发的网络售票系统、证券交易系统和身份确认系统等都曾出现过这样的问题，造成了严重的不良后果。在 Internet 网络环境下，软件需求的变化性加大，要适应各种不同的客户需求；软件的复杂性加大，要处理大量并发分布的事务。这样的软件更容易出错，因此软件的可靠性和性能要求也就更高了。

投入了巨大的人力和资源开发出来的软件，其产品质量往往不甚理想，不能达到预期的目标，这在软件行业是一种普遍的现象。因为程序人员几乎总是习惯从技术的角度去理解用户的需求，使得软件设计人员对客户需求的理解与用户的想法常常有很大距离。通常，用户对自己所需软件的功能和性能在事前是难以确切地表述清楚的，从软件项目一开始，就可能存在软件开发人员与用户理解的概念差异。软件设计带有太多的灵活性和随意性，加之用户需求经常会发生变化，这使软件质量控制成为一个很难解决的课题。对于大规模网络应用软件工程开发，在满足用户需求、控制开发进度和保证软件质量等方面依然有许多问题需要解决。

1.1.3 软件生命期模型

目前，划分软件生命周期的方法有许多种，软件规模、软件类型、软件开发时使用的方法及开发环境等，都会影响软件生命周期的划分。在划分软件生命期阶段时，应遵循的一条基本原则，就是使各阶段的任务彼此间尽可能相对独立，同一阶段各项任务的性质尽可能相同。为了用工程化的思想有效地管理软件生命期活动的全过程，一般可以将软件生命期划分为如下六个阶段。

1. 软件计划

软件计划的任务是确定软件开发的总目标，确定工程的可行性，理解工作范围和所花的代价。软件计划应以可行性研究报告为基础，由软件人员和用户共同确立软件的功能和限制条件，制定软件计划任务书。

2. 软件需求分析

软件需求分析的目标是在系统模型分析的基础上，建立软件需求规格说明书。通常，用户知道必须做什么，但是并不能完整准确地表达出他们的要求，更不知道如何用计算机解决他们的问题；软件人员知道怎样用软件实现人们的要求，但对特定用户的具体要求并不能完全掌握。因此，在需求分析阶段，系统分析人员必须与用户密切配合，充分交流信息，建立经过用户确认的系统模型，作为以后软件设计和实现的基础。系统模型是要求开发的目标软件系统的抽象表示，不同的系统模型抽象会导致不同风格的软件需求规格说明。

软件需求分析是软件工程开发中的重要一步，也是决定性的一步。通过软件需求分析，才能把软件功能和性能的总体概念抽象描述为具体的规格说明，成为软件人员与用户对要求开发的目标系统理解一致的共同基础。

3. 软件设计

软件设计就是从软件需求规格说明出发，形成软件的具体设计方案的过程。软件设计要求确定软件系统的体系结构、给出系统中各软件模块的相互关系、数据库的运用，以及每个模块的功能说明；还应考虑到在软件需求规格说明中对实现环境的要求，如网络环境下运行支撑的要求等。

软件设计又分为总体设计和详细设计两步。总体设计给出系统的整体结构，如软件系统由

哪些模块组成及模块间的关系。详细设计给出各个模块的具体描述，其中包含必要的细节，程序员根据它们可以独立地写出实际的程序代码。

4. 软件编码

根据软件设计的结果，选择合适的程序语言为每一个要求编码的软件模块编写程序。编写的程序应该是结构良好且易于理解。有些软件模块可能不需要编程，仅利用现成的可复用组件，并对照设计要求进一步检查确认和客户化即可。

5. 软件测试

在一个软件系统的整个开发过程中，会出现一系列“信息转移”。信息转移是发生错误的根源。如在需求分析阶段，系统分析员错误地理解了用户的需求，发生了用户到系统分析员之间的信息转移错误；系统分析员在书写需求规格说明书时不能正确表达自己的思维，发生了系统分析员到文件的信息转移错误。在软件开发过程中，软件人员和用户都要参与，人的活动和交互不可能完美无缺，人犯错误的机会也很多。所以，软件开发过程中总是不可避免地会出现错误。软件测试是对软件需求分析、设计和编码的最后复审，是保证软件质量的关键一环。

软件测试包括单元测试和集成测试。单元测试是根据详细设计说明，对软件模块进行详细的测试。集成测试是在单元测试的基础上将各单元模块装在一起进行整体测试。有时，开发的软件系统是更大的一个计算机系统的组成部分，在这种情况下，集成测试还包括所开发的软件与其他软件系统放在一起进行系统有效性测试。经过一系列的测试和排错，最后得到可交付运行使用的软件。

6. 软件维护

经过测试的软件仍然可能有错，加之用户需求和系统运行环境也有可能发生变化，因此交付运行的软件仍然需要继续完善、修改和扩充。这就是软件维护。通常有四类维护活动，即改正性维护，诊断和改正在使用过程中新发现的软件错误；适应性维护，修改软件以适应环境的变化；完善性维护，根据用户的要求改进或扩充软件，使之更加完善；预防性维护，修改软件为将来的维护活动预先做准备。

软件是程序及软件开发、使用和维护所需要的所有文档。根据这样的定义，软件不再仅仅是程序，研制软件也不仅仅是编写程序。按照软件工程化研制的要求，软件生命周期每一阶段完成确定的任务后，都要产生一定格式的文档。表 1.1 列出了软件生命期每一阶段的基本任务及工作结果。

表 1.1　软件生命期阶段任务划分

阶　段	基本任务	工作结果
软件计划	理解工作范围	可行性研究报告、计划任务书
需求分析	定义用户要求	需求规格说明书
软件设计	确立软件结构	设计说明书
软件编码	编写程序	程序
软件测试	发现和排除错误	可运行的系统
软件维护	运行和管理	改进的系统

软件生命期划分为上述六个阶段，这就为工程化的软件研制提供了可遵循的途径。但必须指出，实际的软件系统研制工作，不可能是直线进行的，常常存在着反复，有时需要从后面的阶段回复到前面。例如，在设计阶段发现需求规格说明书有不完整或者不精确之处，就需要回到需求分析阶段进行“再分析”；测试阶段发现了模块内部或者系统的错误，有时甚至要回溯到设计阶段对原来的设计进行修正。

软件生命周期前五个阶段，即计划、分析、设计、编码和测试，通常称之为软件的开发期。最后一个阶段称之为软件的维护期。在软件的整个生命周期中，维护的周期最长，工作量也很大。

仅就开发阶段而言，以上所讲的是开发周期五阶段论，重点强调软件工程的规范和管理，强调自顶向下的软件过程。但这种阶段划分的灵活性较差，回溯和再分析的代价较大。在实际的软件开发中，许多软件工程师倾向于用更具弹性的、快捷的、廉价的方式去开发所需的软件，特别是在软件规模较小和用户需求比较模糊的时候，开发周期演变成为轻量级系统建模、代码设计和迭代测试等三阶段的反复循环，直至完成最终目标系统。这种开发周期三阶段论划分重点强调持久的开发和效率。

1.2 软件开发方法

软件开发过程是软件开发人员以制造软件产品为目的、在软件工具支持下完成软件开发期各阶段任务的一系列相关过程。由于目标软件产品的性质及所采用的软件开发方法的不同，软件过程也常常很不相同。本节介绍几种常用的软件过程模型，尽管它们存在很大的差异，但其基本特征元素都是共同的，这些基本特征元素有四种。

- 软件规格说明：描述软件功能及其行为。
- 软件设计和实现：研制符合规格说明要求的软件。
- 软件验证：确保软件符合客户需求。
- 软件演化：适应变化的客户需求。

软件开发方法是软件开发过程所遵循的规则和步骤。一个好的软件开发方法应能覆盖软件开发活动的全过程，并且方便在开发活动各阶段之间的过渡和演化。本节重点介绍三种代表性的软件开发方法，即结构化方法、面向对象方法及敏捷软件开发方法，并在讲述软件复用概念的同时，介绍了基于组件的开发方法。有关组件化软件工程的详细内容将在第 9 章再行详述。

1.2.1 软件开发过程

软件开发过程可以分为顺序的开发过程和反复的开发过程。在顺序的开发过程中，一旦某项任务完成，过程路径便不再返回到这个任务或在它之前完成的任务了。在反复的开发过程中，过程路径可以回到以前完成的任务，进行适当改变或调整，并使这种变动的效果在过程路径中向前传递。第三种可能的方式是结合顺序和反复模型的增量式方式。反复的和增量式的过程模型结合起来，又形成一种螺旋式的过程模型。

软件过程模型是对一个软件过程的抽象描述。每个过程模型从一个特定的角度描述了一个软件过程。软件过程模型不是为开发人员提供一个面面俱到的软件过程规范，而是围绕软件生命周期中最核心的问题展开的，模型的准则是清晰、敏捷和宏观上严格、微观上灵活。

1. 顺序的过程模型(图 1.1)

软件工程管理人员通常最偏爱这种开发过程模型。顺序过程模型对过程的控制较强。这种