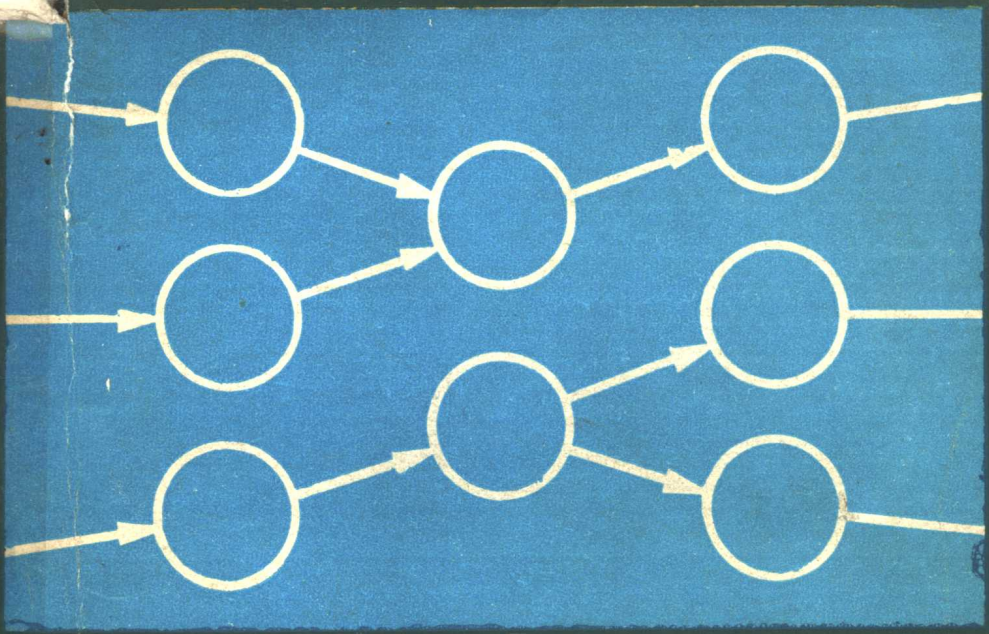

冯玉琳　赵保华　编

软件工程

方法·工具和实践

中国科学技术大学出版社

软件工程

—方法·工具和实践

冯玉琳　赵保华　编

中国科学技术大学出版社

1988·合肥

内 容 简 介

本书系据作者近几年在中国科学技术大学校内外多次讲授的讲义整理而成，是软件工程的一部简明教程，目的在于使读者对软件工程学的全貌有个概要的了解，从而有可能将个人小程序的设计经验扩展到大程序设计方面。

本书内容丰富，并附有大量实例，重点介绍软件工程的方法和工具，强调实践，适合于大专院校计算机软件专业作为软件工程课的教材或教学参考书，亦可供从事计算机专业的工程技术人员阅读和自学之用。

软 件 工 程

——方法·工具和实践

冯玉琳 赵保华 编

责任编辑 伍传平 封面设计：罗 洪

*

中国科学技术大学出版社出版

（安徽省合肥市金寨路96号）

安徽省合肥永青印刷厂印刷

安徽省新华书店发行 各地新华书店经售

*

开本：850×1168/32 印张：11.75 字数：300千

1988年2月第1版 1988年2月第1次印刷

印数：1— 000册

ISBN 7-312-00065-7/TP·1

书号：15474·4 定价：2.40元

前　言

从计算机程序设计的领域独立出“软件工程”这个名词，标志着人们试图用“工程化”的思想来指导并解决软件研制中面临的种种困难和混乱。软件工程的目标在于研究一套科学的工程方法，并与此相适应，发展一套方便的工具系统，力求用较少的投资获得高质量的软件。

本书是软件工程的一本简明教程，目的在于使学生对软件工程科学的全貌有个概要的了解，从而使学生的注意力从个人小程序的独立活动转移到或扩展到大程序设计方面。

与其它一些课程的特点不同，软件工程课是一门“设计”课程，强调实际训练。课堂教学只讲授软件工程学中的一些基本概念、方法和原则，更重要的是，学生必须参加一个从计划、分析、设计到编码、测试的软件开发的全过程，以便从中得到实际的从事软件工程的训练和经验。课程将提出一些设计题目供学生选用。

全书分为三部分，十三章。

第一部分为1—2章，介绍软件和软件工程的基本概念。

第二部分为3—9章，按照软件生命期的观点依次介绍各个阶段，包括软件计划、软件需求分析、软件设计、软件编码、软件测试和软件维护。由于软件工程的方法很多，这里只重点介绍结构化的分析和设计方法，包括DFA，SD，SP，Jackson方法等。

第三部分是10—13章，讨论软件工程的外在支持，包括人的因素方面，例如软件管理；物的因素方面，例如软件工具和环境，强调软件开发工具和环境在软件工程中的重要地位，并对这方面当前发展动态作了一般介绍，特别是对以UNIX为背景的支

持环境系统作了一些典型分析。

小项目软件有“小”的特点，专辟一章讲述，并给出了一个小编译系统的设计例子。

本书结束语，再次强调了软件工程实践的重要性。有一个课程实习项目提要列在附录C中，供读者选择参考用。

本书的主要特点是简明，内容与教学环节紧密配合，重点介绍软件工程的方法和工具，并强调学生的实践训练，适合于大专院校计算机软件专业作为软件工程课的教材或教学参考书。

为顾及一般读者，本书叙述力求通俗易懂，重在思想方法介绍。略去部分章节后可供具有程序设计初步经验的广大工程技术人员自学之用。

本书是根据我校近几年使用的软件工程讲义整理而成。因时间仓促，其中难免出现有疏忽谬误之处，敬请读者指正。

在本书的编写过程中，许多同志提出了不少宝贵意见，初稿形成时曾与中国科学院软件研究所仲萃豪、丁茂顺等同志，电子工业部华东计算技术研究所蔡林希同志以及中国科学技术大学计算机系唐策善同志进行过许多有益的讨论。安徽大学李庆同志曾参加本教程教学并很好地指导了课程设计实习。中国计算机函授学院钱胜洲、张宁等同志为本书做了大量实际工作。本书的附录D：课程实习项目示例是中国科学技术大学硕士研究生桂自强同志编写的。对以上这些同志的热情帮助和辛勤劳动，谨致深切谢意。

编　者

1986年

目　　录

第一章 绪 论

§1.1 什么是软件工程

软件工程是研究“大”程序设计的方法、工具和管理的工程科学。

首先，软件工程的要点是“大”程序设计。假设一个人在一个月内能够完成1000行代码的设计，按照如此的工作效率，一个10,000行代码的程序是否一个人花十个月或者十个人花一个月就能完成呢？回答是否定的。当程序的规模从1000行增加到10,000行时，原代码行增加了十倍，但是，程序复杂性程度的增加却远不止十倍。

当许多人共同设计一个由许多模块组成的大型系统时，在许多年的时间内会出现许多不同的版本，面对着如此的许多人、许多模块、许多年和许多版本……，如此之“多”的活动，如何进行设计、开发和维护，这里不仅涉及到技术的方面，如设计方法、版本控制等，而且关系到行政管理的方面，如价格评估，人员组织等。

六十年代末期国际上出现了“软件危机”，其主要表现是：软件质量差，可靠性难以保证；软件成本增长难于控制，极少有在预定的成本预算内完成的；软件开发进度难于控制，周期拖得很长；软件的维护很困难，维护人员和费用不断增加。例如：IBM公司开发的OS／360系统，耗资几千万美元，花费了五千多人年，拖延了几年才交付使用。交付使用后不断发现错误。有的软件甚至耗费了大量的人力财力，结果半途而废。

考虑到研制一个软件系统同研制一台机器或建造一座楼房有许多共同之处，因此可以参照机械工程、建筑工程中的一些技术来进行软件的研制，人们试图用“工程化”的思想作指导来解决软件研制中面临的困难和混乱，从根本上解决软件危机。1968年北大西洋公约组织的学术会议第一次使用了“软件工程”这个名词。

正象所有的工程科学一样，软件工程自身遵循着一套科学的设计原理和方法，以这些方法为前导，人们发展一系列的软件工具系统来帮助工程软件的开发。但是软件工程又与其他各种工程科学很不相同。软件是抽象的、逻辑性的产品，不是实物性的。研制和维护软件本质上是一个“思考”的过程，很难对它进行控制。软件的设计基本上无统一的设计标准，无准确的数量分析，无足够的可靠性保证以及无有效的维护手段，这就决定了软件的研制和开发较其它工程项目要困难得多。

综言之，软件工程的目标在于研究一套科学的工程方法，并与此相适应，发展一套方便的工具系统，力求用较少的投资获得高质量的软件。

§1.2 软件工程面临的问题

摆在软件工程科学面前有许多方面的棘手问题，以下就软件价格、软件可靠性、软件维护、软件生产率和软件再应用等方面作一些简单分析。

1.2.1 软件价格

由于新的电子器件的出现，计算机硬件的功能和质量在不断提高，而成本却大幅度下降。计算机硬、软件投资所占比例中，软件投资在迅速上升。据1972年U·S·Air Force发布，美国空军计算机投资中软件所占比例，1955年低于20%，1970年剧增到

60%，预计1985年为85%，软件投资比重上升的曲线如图1.1所示。

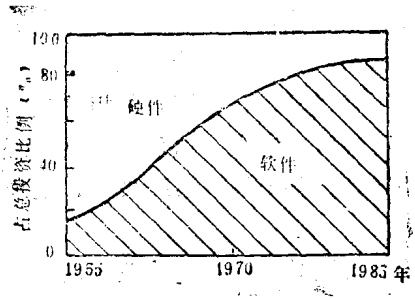

图1.1 软硬件投资比例

再看纽约Wall Street Joural, Jan.23, 1979年公布的Carter 1980年财政年度预算，总计有570亿美元用于发展计算机系统，其中320亿美元，即56%用于发展计算机软件。

随着计算机应用愈来愈广泛，新的应用领域要求开发新的软件系统。另一方面，当新一代的计算机取代现有的旧的计算机时，软件系统也要随之更新。而程序设计是高度密集型技术，所以，软件价格上升的势头将会继续下去。

1.2.2 软件可靠性

指软件系统能否在既定的环境条件下运行并实现所期望的结果。通常大约有40%左右的软件开发的代价要花在软件设计之后的测试和排错上，即使如此，也不能保证经测试的软件就没有错误。现代的程序设计技术，如结构程序设计，能有助于减少程序设计中的错误；但是不幸的是，我们很难保证一个大的软件系统确实是无任何错误的。

大的软件系统的测试是一件很复杂很费时的事情，程序的任何改变都可能会涉及到许多人，这就使得软件系统花在测试阶段的代价非常之大。为了提高软件的可靠性，就要付出相应的代

价，重要的是在可靠性和代价之间作出权衡。

除了软件的正确性以外，在更广泛的意义下，软件的可靠性还应该包括安全性和健壮性。

对于合理的一组输入，系统会给出正确的结果；而对于用户的有意或者无意的不合理输入，系统应能拒绝这种输入，并指出输入的不合理性，提醒用户注意。这是软件系统的安全性。对于不合理的输入束手无策，甚至导致系统失败，这样不安全的系统是不能使用的。

健壮性是指软件系统对环境变化的适应性。当软件系统所处的环境发生变化时，例如存贮溢出，硬件故障等意外事件发生时，系统都能按照某种预定的方式作适当处理，有效地控制事故的蔓延，不致丢失重要的信息，从而避免灾难性的后果。

1.2.3 软件维护

软件维护是指修正已经运行了的软件系统所需要做的工作。几乎所有的软件系统在运行了若干年以后，仍然需要作这样或者那样的修改和补充。一个软件开发机构，要把很大的精力花在维护已有的软件上。随着更多软件的生产，用之于软件维护的力量还会愈益增加，这样势必会束缚开发组织无法去生产新的软件。

为什么维护需要花费如此大的精力呢？这是因为已经运行的程序还总是在变化，故障要排除，改进要加进去，而且优化也要做。不仅当前的版本要改变，仍在使用的去年的版本也要修改，可能即将投入使用的明年的版本也要修改，除了解决原有的问题需要精力外，改变本身又会带来新的问题，也要花精力去解决。

因此，如何减少软件维护的总工作量，也就成为软件工程的主要目标之一。

1.2.4 软件生产率

计算机的广泛使用使得软件的需求量迅速上升。世界各国普

遍感到软件人力资源的缺乏，这种趋势仍在继续增长下去。但是，目前的软件生产仍然基本上处于手工状态，生产率很是低下，而生产出来的软件的质量又很不理想，不适应市场对软件的大量需求。如何以较少的投资，获得“高产优质”的软件，这是软件工程的主要目标之一。其实现途径有二，即良好的软件工程设计方法和方便的软件设计工具环境。

软件工程设计方法告诉人们“什么时候做什么？什么时候怎样做？”。由于软件研制过程毕竟是相当复杂的，涉及的因素很多，所以这些软件方法又有不同程度的灵活性与试探性。一般说来，软件方法规定了

△ 明确的工作步骤

△ 具体的文档格式

△ 确定的评价标准

近几十年来，软件工程已研究发展了各种各样的软件工程设计方法，这些方法的实用范围是各不相同的，有的方法适用于数据处理系统，而有的方法适用于实时控制系统；各种软件方法的风格也迥然不同。有的方法仅仅是一组指导性的原则，而有的方法则有较具体的处理规则，有的方法建立在严密的数学基础之上，而有的方法则是实际经验的总结。本课程只能选择一些比较实用的、有代表性的方法进行讨论。

方法和工具两者之间有着密切的联系，相辅相成。在软件开发维护的不同阶段，都应该有这样或者那样的作为软件工具的程序系统，帮助人们自动地完成许多数据分析和处理工作。这样的工具环境愈完善，软件的设计和维护也就愈方便。这是当前提高软件生产率的一个很重要的方面。人们希望研究出一套系统的软件方法，一组配套成龙的软件工具，从而为软件人员提供一个能复盖整个软件生命期的良好的工作环境，使软件生产率必将大为提高。

1.2.5 软件再应用

软件再应用也是提高软件生产率、降低软件成本的一个重要方面。当人们一次又一次地去设计一些互相类同的程序时，很多的劳动都花费在一些基本重复的工作上，软件再应用的技术就是旨在减少这种重复。

可再应用的软件单位可以是程序代码，如子程序；也可以是软件的逻辑结构，如软件模块的详细设计表示。

软件再应用方式大体上可分为两种。最简单易行的一种方式是直接组合可再应用的软件单位。这当然会有许多使用上的限制，用得最多的是各种程序库。另一种方式是按模式再应用，有如下几种情形。

1) *面向问题编程* 用面向问题的语言，或者非常高级程序语言（very high-level languages）编写程序，这样的程序实质上是问题的一种高级描述（specifications）。再由此出发，可以得到各种不同高级语言的机器实现。

2) *程序变换* 可以设计这样的源码变换系统，使之能把一种语言的源程序变换成另一种语言的程序。鉴于FORTRAN语言具有各种丰富的应用程序库，只要设计成功从FORTRAN语言到另一种语言，比如说PASCAL语言的变换系统，就立即可以得到PASCAL语言的同样丰富的程序库。

3) *程序系统生成* 许多程序系统虽然表面上看来很不相同，但是它们的基本结构模式却是一样的。例如，对于不同语言的结构化编辑，除了语言的结构框架或者模板不同之外，编辑系统其他部分的设计基本上都是类同的；对于高级程序语言编译中的语法分析和代码生成也是如此。我们可以设计出一个关于一定的结构模式的系统，给定一些特定的具体说明和要求，这样的系统就能产生出特定的软件来。通常所说的许多程序自动生成系统，就是这种情形的软件再应用的例子。