HZ BOOKS
华章教育

高等院校精品课程系列教材

软件工程教程

孙 涌◎主编
陈建明 王 辉◎参编

精品课程

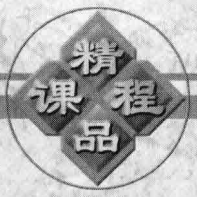
An Introduction to Software Engineering

机械工业出版社
China Machine Press

高等院校精品课程系列教材

软件工程教程

孙 涌◎主编
陈建明 王 辉◎参编



An Introduction to Software Engineering

机械工业出版社
China Machine Press

本书全面系统地介绍了软件工程的有关概念、原则、方法和工具。全书共15章，内容包括：软件工程中面向过程、面向对象的开发方法，技术度量，质量保证，软件项目计划与管理，用统一建模语言UML开发软件的方法等。另外，还对设计模式、敏捷软件开发、Web工程等软件工程相关领域进行了介绍和讨论。本书既注重科学性和系统性，又注重实用性和新颖性。

本书可作为大学计算机及相关专业本（专）科学生的教材或教学参考书，也可作为研究生的参考教材。

图书在版编目（CIP）数据

软件工程教程 / 孙涌主编. —北京：机械工业出版社，2010.4
（高等院校精品课程系列教材）

ISBN 978-7-111-30002-1

Ⅰ. 软…　Ⅱ. 孙…　Ⅲ. 软件工程—高等学校—教材　Ⅳ. TP311.5

中国版本图书馆CIP数据核字（2010）第037498号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：陈佳媛
北京诚信伟业印刷有限公司印刷
2010年4月第1版第1次印刷
184mm × 260mm · 21.75印张
标准书号：ISBN 978-7-111-30002-1
定价：36.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

出版者的话

机械工业出版社华章公司秉承“全球采集内容，服务中国教育”的理念，经过十余年的不懈努力，引进、翻译、出版了大量在计算机科学界、电子科学界享有盛名的专家名著与名校教材，其中包括Donald E. Knuth、Alfred V. Aho、Jim Gray、Jeffrey D. Ullman、R. Jacob Baker等大师名家的一批经典作品，这些作品对国内计算机教育事业的发展起到了一定的推动作用。今天，全国高等学校精品课程建设工作的蓬勃开展为我们更好地服务于计算机教育、电子信息科学教育提供了良好的契机，我们将以严谨的治学态度及全面服务的专业出版精神，在国内广大院校老师们的支持与帮助下，陆续推出具有国内一流教学水平的“高等院校精品课程系列教材”。

精品课程是具有一流教师队伍、一流教学内容、一流教学方法、一流教材、一流教学管理等特点的示范性课程，是教育部实施的“高等学校教学质量与教学改革工程”的重要组成部分，是教育部深化教学改革，以教育信息化带动教育现代化的一项重要举措。自2003年精品课程建设项目持续推进以来，国内高校中的优秀教师纷纷在总结本校富有历史传统而又特色突出的课程教学方法与经验成绩的基础上，充分运用现代网络传播技术将优质的教学资源上网共享，使国内其他高校在实施同类课程教学的过程中能够借鉴、使用这些优质的教学资源，在更大范围内提高高等学校的教学和人才培养质量，提升我国高等教育的综合实力和国际竞争能力。经过几年的共同努力，已经建立起了较为齐全的各门类及各专业的校、省、国家三级精品课程体系，期间先后有总计750门课程通过了专家评审，获得了“国家精品课程”称号。

这些各个层次的“精品课程”建设过程都比较充分地体现了教育部所要求的七个重点，即：具有科学的建设规划，配备高水平的教学队伍，不断进行教学内容和课程体系的改革，使用先进的教学方法和手段，注重建设系列化的优秀教材，高度重视理论与实践两个环节，切实激励各方人员共同参与。也正因为这样的多方面积极参与，使得我国的高等教育在近年来由精英教育转向大众教育的跨越式发展中取得了教学质量上的突破与飞跃。精品课教材作为精品课程的要件之一，比以往教材更加具有实践检验性，教学辅助资源经过不断地更新与补充更加丰富，是精品课教学团队智慧的共同体现。

“师者，所以传道授业解惑也”。教材是体现教学内容和教学要求的

知识载体，是教师进行教学活动的基本工具，是提高教学质量的重要保证。精品课程教学团队中优秀的老师们集多年治学经验撰写出版相关教材，也是精品课程建设的一个重要方面。华章作为专业的出版团队，长久以来以“传承专业知识精华，服务中国教育事业”为使命，遵循“分享、专业、创新”的价值观，实践着“国际视野、专业出版、教育为本、科学管理”的出版方针，愿与高等院校的老师共同携手，为中国的高等教育事业走向国际化而努力。

为更好地服务于精品课程配套教材的出版，华章不仅密切关注高校的优秀课程建设，而且还将利用自身的优势帮助教师完善课程设置、提供教辅资料、准备晋级申报、推广教学经验。具体详情可访问专门网站http：//www.hzbook.com/jpkc.aspx，并可在线填写出版申请，欢迎您对我们的工作给予帮助和指导。

投稿专线：010-88379604

投稿E-mail：hzjsj@hzbook.com

华章教育

华章科技图书出版中心

前 言

自20世纪60年代末期创立以来，软件工程伴随着计算机软、硬件的快速发展，经历了从结构化到面向对象的一系列发展过程，并且已经形成了若干工具、分支学科，目前软件工程仍然是一个异常活跃的研究领域。人们已经认识到，如果有哪个项目不遵循软件工程原则，必定会受到实践的惩罚。当然，软件工程学的研究范围非常广泛，包括技术方法、工具和管理等许多方面，新的技术方法和工具还在不断涌现。

本书集合作者多年从事本科生和研究生软件工程课程教学经验，在参考了大量国内外教材与专著的基础上，结合当前大学软件工程课程教学的实际要求和将来从事实际软件开发的相关情况而编写。本教程作为苏州大学首批精品课程建设计划之一，是计算机专业的一门专业课。本教材虽然不可能包含软件工程的全部内容，但却是本着易懂、实用的原则并结合多年从事软件工程教学、科研实践编写而成。另外考虑到软件工程的发展，编入了一定量的现代软件工程的概念、方法及技术。

在写作过程中，本书重点关注了以下几个方面：

- 以实用为主的同时，适当加强理论的叙述，在相关章节中增加了一些形式化的内容，便于更深入地学习软件工程相关理论或指导实际的软件项目开发。
- 按照一般软件产品（项目）开发过程的顺序进行软件工程内容叙述，便于进行相关实验或实践。
- 面向对象技术在突出统一建模基本概念、方法的同时，强调建模与高级语言的结合与融合，避免了过程建模与具体实现脱节的问题。这一点对于我国的学生尤为重要，其理由是我国当前绝大部分学生认为只有编写代码才能进行软件开发。
- 对一些比较成熟的最新技术进行了介绍，如设计模式、Web工程等。通过这部分内容的学习，既能够与当前实际开发紧密结合，也可为将来继续深化、研究起到一个入门的作用。

本书共15章。第1章概括介绍了软件工程学的基本原理、概念和方法。第2章到第6章主要介绍了面向过程软件工程生命周期顺序的前几个阶段的任务、过程、方法和工具。第7章到第9章比较完整地介绍了面向对象的开发方法。第10章到第13章介绍了软件工程中面向过程及面向对象的软件测试、维护、质量保证及相关的过程、方法和工具。第15章较详细

地介绍了目前软件工程的一些新技术，包括敏捷软件开发、设计模式、软件复用、Web工程等。

由于软件工程这门课程涉及面广，内容丰富，发展迅速，所以我们在取材方面，既考虑保持传统的内容，又充分将软件工程中的新技术、新发展融入其中。同时，我们也兼顾到目前高校学生的实际情况，力争做到取材合理、内容新颖、理论为主、结合实际、重点突出、实用性强。

根据多年从事软件开发和软件工程课程教学的经验和体会，作者认为：软件工程这门课程的特点在于：它看似简单，大部分内容是叙述性的，但要真正掌握好并运用好，绝非易事！特别是软件工程的思想、方法、理论、技术等贯穿整个软件产品（项目）开发的始终，这是任何一门课程所无法替代的。它既有宏观的一面，又有微观、具体的一面，同时还与诸多学科相关联。在此，希望广大学生与读者，能够在学习本书内容的同时，将相关知识与一个实际项目结合，哪怕是非常小的项目，只有这样，才有可能真正学好软件工程。

本书由孙涌主编，陈建明、王辉参编。全书完稿后，由孙涌进行统稿和整理工作。

在本书的编写过程中，感谢王璁伟、房鹏、王晋、姜晓猛、沈文超、葛小培、陈祥荣、耿胜恩等同学为之付出的辛勤劳动，同时还要感谢机械工业出版社的编辑们对本书出版给予的支持。

由于时间仓促，加之作者水平有限，书中难免存在不足和疏漏之处，敬请广大读者不吝赐教。

编　者

2010年4月

教学建议

教学内容	学习要点及教学要求	课时安排	
		计算机专业54学时	计算机相关专业36学时
第1章 软件工程概述	• 了解软件的特点，对此有一个基本认识。这是任何从事和将要从事软件行业的人必须要面对的问题 • 了解软件开发的生命周期，软件的开发到目前为止还基本沿用该过程 • 对软件工程的一般性认识	2	2
第2章 软件生命周期过程模型	• 了解一般科学问题求解的思路与方法 • 掌握其中常用的几种方法：分解与合并、抽象与规范、建立模型等 • 掌握常见的几种软件过程模型，如瀑布模型、原型化模型、螺旋模型、V模型等	2	2
第3章 需求分析	• 了解需求分析的目的与任务 • 熟悉需求分析的主要方法 • 掌握需求的层次 • 学习需求获取的技术 • 了解需求开发与管理 • 掌握需求规格说明书的主要结构与描述 • 掌握对需求描述的几个重要方法：数据流图、数据字典、数据加工、E-R关系模型 • 对形式化方法有一定的了解	6	6
第4章 概要设计	• 了解概要设计的一般概念、方法与作用 • 熟悉模块独立性及其衡量方法 • 学习结构化设计的一般方法，能够处理简单的从需求结果向概要设计的过度 • 掌握数据设计的一般方法 • 对设计中的一些问题进行思考，如界面设计中，关于喜好、文化差异等	4	4
第5章 详细设计	• 了解详细设计的任务 • 掌握详细设计的基本方法。如详细设计表示法、结构化程序设计、面向数据结构的设计 • 学习详细设计文档的格式、写作方法 • 了解软件复审的作用和重要性	4	4
第6章 编码与语言选择	• 了解对于程序的要求 • 学习语言的分类及特点 • 学习编码的风格 • 培养良好的编码习惯	2	2（选讲）
第7章 面向对象方法	• 了解面向对象的基本思想、基本概念和基本原理 • 理解传统方法和面向对象方法的不同点，理解面向对象技术的优势所在 • 理解面向对象分析的概念 • 掌握面向对象开发的一般方法 • 学习如何进行面向对象分析 • 掌握OO分析模型的类属成分 • 掌握面向对象设计模型的构成元素 • 熟悉面向对象设计 • 了解面向对象设计中的Yourdon方法、Booch方法	4	4

（续）

教学内容	学习要点及教学要求	课时安排	
		计算机专业54学时	计算机相关专业36学时
第8章 统一建模语言	• 了解统一建模语言的发展历史 • 熟悉统一建模语言视图以及类、构件、部署和协作图中的图标 • 学习各种视图及其之间的关系 • 掌握静态视图、用例图、交互视图、状态图、活动视图等基本视图的画法和使用 • 熟悉UML与Java的对应关系 • 通过统一建模语言的综合应用的学习，能够在使用统一建模语言方面得到初步的训练	4	4
第9章 统一软件过程	• 了解统一软件过程基本思路及内容 • 学习统一软件开发过程及迭代和递增、核心工作流的基本内容 • 学习统一软件过程并熟悉其中的各个环节的主要内容	4	4（选讲）
第10章 软件测试	• 了解软件测试的目标、软件测试的原则、软件测试的方法、软件测试与软件开发各阶段的关系、测试信息流、错误分类 • 学习单元测试、集成测试、确认测试相关知识和技巧 • 掌握逻辑覆盖、等价划分、边界值分析、错误推测、实用测试策略 • 学习一般的纠错方法 • 了解面向对象的测试策略 • 熟悉OO软件的测试用例设计 • 对GUI测试、测试文档和帮助设施、实时系统测试有一定的了解	4	4
第11章 软件维护	• 了解软件维护的基本内容和特点 • 掌握一般的软件维护方法 • 学习基本的软件维护过程及技术 • 对软件维护过程中的副作用有充分的认识 • 了解软件再工程的主要内容	4	4
第12章 软件质量及其管理	• 了解软件质量的概念及属性 • 学习软件质量保证概述、软件质量保证计划、软件质量成本、软件质量控制相关知识 • 掌握软件质量度量方法 • 了解质量度量模型和三种度量模型 • 学习软件质量评价 • 学习质量体系的建立和实施 • 学习软件能力成熟度模型的基本内容和含义	4	4（选讲）
第13章 软件项目管理	• 了解项目管理的基本概念和内容 • 熟悉一般软件项目可行性研究的内容 • 能进行一些简单软件项目估算工作 • 学习项目进度安排、如何进行人员组织等知识 • 了解软件风险管理和软件配置管理	4	4（选讲）
第14章 CASE环境与工具	• 了解软件开发环境的特点、理想环境的模型、CASE环境简介 • 熟悉CASE环境的组成与结构 • 掌握常用CASE工具的基本使用方法 • 进行适量CASE软件工程实践	2	2
第15章 软件工程新技术概述	• 一般了解敏捷的概念、敏捷过程的含义、敏捷过程模型 • 通过对设计模式的讲解，能够掌握几个基本的设计模式 • 了解Web工程相关内容	4	4（选讲）

目 录

第 1 章

软件工程概述

计算机软件是整个计算机系统中具体实现各种功能和操作的核心部分。软件工程采用工程的概念、原理、技术和方法来开发、维护计算机软件，将工程管理技术成功的经验和思想与具体的软件开发过程、研究技术相结合，形成一整套适合于计算机软件开发的方法、规范和技术。因此，软件工程这门课程，对于从事软件开发研究的专业人员，特别是高层次的管理、分析、开发人员，显得比以往更加重要。本章主要介绍软件工程的发展背景、软件工程学的范畴、软件工程的基本目标等。

1.1 软件发展和软件危机

1.1.1 软件的定义和发展

自从世界出现第一台计算机到目前，软件的开发与研究经历了几十年的发展。对于计算机软件，有多种不同的定义，但一般可以理解为“软件是能够完成预定功能和性能的可执行的计算机程序和使程序正常执行所需要的数据，加上描述程序的操作和使用的文档”。

根据发展的历程，可以分为4个阶段，如图1-1所示。

1）程序设计阶段，约为20世纪五六十年代。

2）程序系统阶段，约为20世纪六七十年代。

3）软件工程阶段，约为20世纪70年代以后。

4）面向对象软件工程阶段，约为20世纪80年代以后。

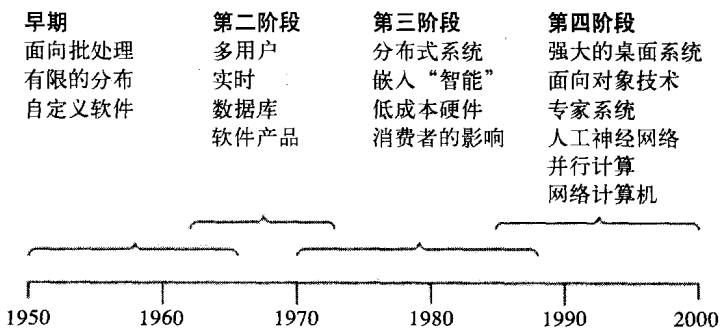


图1-1 软件的发展

软件发展过程中最根本的变化体现在以下几个方面：

1）人们改变了对软件的看法。在20世纪五六十年代时，程序设计曾经被看做一种任人发挥创造才能的技术领域。当时人们认为，程序运行后只要能在计算机上得出正确的结果，程序的写法可以不受任何约束。随着计算机的使用范围日趋广泛，人们要求这些程序易懂、易使用，并且容易修改和扩充。于是，程序便从按个人意图创造的“艺术品”转变为能被广大用户接受的工程化产品。

2）软件的需求是软件发展的动力。早期的程序开发只是为了满足开发者自己的需要，这种自给自足的生产方式是其低级阶段的表现。进入软件工程阶段后，软件开发的成果具有社会属性，它要在市场中流通以满足广大用户的需要。

3）软件工作的考虑范围从只顾及程序的编写扩展到涉及整个软件生存周期。

1.1.2 软件危机过程

20世纪60年代后期，随着计算机应用的日益普及，软件数量急剧膨胀，众多因素导致在软件的开发过程中，所开发的软件产品质量低下，众多软件无法满足用户需求，软件的可维护性差，以至于问题不断堆积，形成日益尖锐的矛盾，通常人们将这些现象通称为“软件危机”。为此，NATO的一个研究小组于1967年提出了“软件工程”概念，并于1968年北大西洋公约组织召开的计算机科学家国际会议上得到签署，与会人员得出结论：软件工程应当使用已经建立的工程科学的基本原理和范型来解决“软件危机”。软件工程学由此产生。

1. 软件危机的主要表现

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题绝不仅仅是“不能正常运行的”软件才具有的，实际上几乎所有软件都不同程度地存在这些问题。概括地说，软件危机包含下述两方面的问题：如何开发软件来满足对软件的日益增长的需求；如何维护数量不断膨胀的已有软件。具体地说，软件危机主要有下述一些表现形式：

1）对软件开发成本和进度的估计常常很不准确。实际成本比估计成本有可能高出一个数量级，实际进度比预期进度拖延几个月甚至几年的现象并不罕见。这种现象降低了软件开发组织的信誉。而为了赶进度和节约成本所采取的一些权宜之计又往往降低了软件产品的质量，从而不可避免地会引起用户的不满。

2）用户对“已完成的”软件系统不满意的现象经常发生。软件开发人员常常在对用户要求只有模糊的了解，甚至对所要解决的问题还没有确切认识的情况下，就仓促上阵，匆忙着手编写程序。软件开发人员和用户之间的信息交流往往很不充分，“闭门造车”必然导致最终的产品不符合用户的实际需要。

3）软件产品的质量往往靠不住。软件可靠性和质量保证的确切的定量概念刚刚出现不久，软件质量保证技术（审查、复审和测试）还没有完全地应用到软件开发的全过程中，这些都导致软件产品发生质量问题。

4）软件常常是不可维护的。很多程序中的错误是非常难改正的，实际上不可能使这些程序适应新的硬件环境，也不能根据用户的需要在原有程序中增加一些新的功能。“可重用的软件”还是一个没有完全做到的、正在努力追求的目标，人们仍然在重复开发类似的或基本类似的软件。

5）软件通常没有对应的文档资料。计算机软件不仅仅是程序，还应该有一整套文档资料。

这些文档资料应该是在软件开发过程中产生出来的，而且应该是“最新式的”（即和程序代码完全一致的）。软件开发组织的管理人员可以使用这些文档资料作为“里程碑”，来管理和评价软件开发工程的进展状况；软件开发人员可以利用它们作为通信工具，在软件开发过程中准确地交流信息；对于软件维护人员而言，这些文档资料更是至关重要、必不可少的。缺乏必要的文档资料或者文档资料不合格，必然给软件开发和维护带来许多严重的困难和问题。

6）软件成本在计算机系统总成本中所占的比例逐年上升。由于微电子学技术的进步和生产自动化程度不断提高，硬件成本逐年下降，然而软件开发需要大量人力，软件成本随着通货膨胀以及软件规模的不断扩大和数量的增加而持续上升。美国在1985年软件成本大约已占计算机系统总成本的90%。

7）软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。软件产品“供不应求”的现象使人类不能充分利用现代计算机硬件提供的巨大潜力。

以上列举的仅仅是软件危机的一些明显的表现，与软件开发和维护有关的问题远远不止这些。

2. 产生软件危机的原因

在软件开发和维护的过程中存在这么多严重问题，一方面与软件本身的特点有关，另一方面也和软件开发与维护的方法不正确有关。

软件不同于硬件，它是计算机系统中的逻辑部件，而不是物理部件。在写出程序代码并在计算机上试运行之前，软件开发过程的进展情况较难衡量，软件开发的质量也较难评价，因此管理和控制软件开发过程相当困难。此外，软件在运行过程中不会因为使用时间过长而被“用坏”，如果运行中发现错误，很可能是遇到了一个在开发时引入的在测试阶段没能检测出来的故障。因此，软件维护通常意味着改正或修改原来的设计，这就在客观上使得软件较难维护。

软件不同于一般程序，它的一个显著特点是规模庞大。例如，美国四代宇宙飞船的软件规模呈指数增长，20世纪70年代末穿梭号宇宙飞船的软件包含4000万行目标代码。假设一个人一年可以开发出一万行的程序，为了开发一个4000万行的软件，是否集中4000人的力量一年就可以完成呢？绝对做不到！因为代码长度增加了4000倍，程序复杂程度的增加远远超过4000倍。而且如何保证每个人完成的工作合在一起确实能构成一个高质量的大型软件系统，更是一个极端复杂和困难的问题，不仅涉及许多技术问题，诸如分析方法、设计方法、形式说明方法、版本控制等，更重要的是必须有严格而科学的管理。

软件本身独有的特点确实给开发和维护带来一些客观困难，但是人们在开发和使用计算机系统的长期实践中，也确实积累和总结出了许多成功的经验。如果坚持不懈地使用经过实践考验和证明是正确的方法，许多困难是完全可以克服的。过去确实也有一些成功的范例。但是，目前相当多的软件专业人员对软件开发和维护还有不少错误观念，在实践过程中或多或少地采用了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

与软件开发和维护有关的许多错误认识和做法的形成，可以归因于在计算机系统发展的早期软件开发的个体化特点。错误认识和做法主要表现为忽视软件需求分析的重要性，认为软件开发就是写程序并设法使之运行，轻视软件维护等。

事实上，对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。只有用户才真正了解他们自己的需要，但是许多用户在开始时并不能准确具体地叙述他们的需要，软件开发人员需要做大量深入细致的调查研究工作，反复多次地和

用户交流信息，才能真正全面、准确、具体地了解用户的要求。对问题和目标的正确认识是解决任何问题的前提和出发点，软件开发同样也不例外。急于求成，仓促上阵，对用户要求没有正确认识就匆忙着手编写程序，这就如同不打好地基就盖高楼一样，最终必然倒塌。

做好软件定义阶段的工作，是降低软件开发成本并提高软件质量的关键。如果软件开发人员在定义阶段没有正确全面地理解用户需求，直到测试阶段或软件交付使用后才发现“已完成的”软件不完全符合用户的需要，这时再修改就为时已晚了。

严重的问题是，在软件开发的不同阶段进行修改需要付出的代价是很不相同的，在早期引入变动，涉及的面较少，因而代价也比较低；而在开发的中期软件配置的许多部分已经完成，引入一个变动要对所有已完成的配置部分都做相应的修改，不仅工作量大，而且逻辑上也更复杂，因此付出的代价剧增；在软件“已经完成”时再引入变动，当然需要付出更高的代价。根据美国一些软件公司的统计资料，在后期引入一个变动比在早期引入相同变动所需付出的代价高2～3个数量级。

通过上面的论述不难认识到，轻视维护是一个最大的错误。许多软件产品的使用寿命长达10年甚至20年，在这样漫长的时期中不仅必须改正使用过程中发现的每一个潜伏的错误，而且当环境变化时（如硬件或系统软件更新换代）还必须相应地修改软件以适应新的环境，特别是必须经常改进或扩充原来的软件以满足用户不断变化的需要。所有这些改动都属于维护工作，而且是在软件已经完成之后进行的，因此维护是极端艰巨和复杂的工作，需要花费很大代价。统计数据表明，实际上用于软件维护的费用占软件总费用的55%～70%。软件工程学的一个重要目标就是提高软件的可维护性，减少软件维护的代价。

了解产生软件危机的原因，澄清错误认识，建立起关于软件开发和维护的正确概念，还仅仅是解决软件危机的开始，全面解决软件危机需要一系列综合措施。

3. 缓解软件危机的途径

软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好，管理严密，各类人员协同配合，共同完成的工程项目。必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法，特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。

应该推广使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好、更有效的技术和方法，尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。

应该开发和使用更好的软件工具。正如机械工具可以“放大”人类的体力一样，软件工具可以“放大”人类的智力。在软件开发的每个阶段都有许多烦琐重复的工作需要做，在适当的软件工具辅助下，开发人员可以把这类工作做得既快又好。如果把各个阶段使用的软件工具有机地集合成一个整体，支持软件开发的全过程，则称为软件工程支撑环境。

总之，为了解决软件危机，既要有技术措施（方法和工具），又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

4. 对软件危机的理解

由以上的叙述可以得知，由于历史原因，人们提出将已经建立的工程科学的基本原理和范型用于软件开发，借此来解决“软件危机”问题。但是，从目前软件发展的实际情况看，虽然现代软件的开发方法、技术、手段、工具等有了巨大的发展。但从广义上来说，“软件危机”所指的问题并没有解决，有些方面的问题甚至更为严重和突出。由此，我们有必要首先对“软件危机”的概念进行一定的考察、定义和重新认识。