

现代软件工程

第1册

管理 技术 篇

周之英 编著

科学出版社

现代软件工程

第 1 册 管理技术篇

周之英 编著

科学出版社

2002

内 容 简 介

本书分三册,每册独立成篇,第1册为管理技术篇,第2册为基本方法篇,第3册为新技术篇。第1册从宏观的角度讨论软件工程的管理技术。首先从分析和讨论软件开发技术的发展史开始,建立软件工程技术的发展的观点和变化的观点,以此作为全套书的线索,进而讨论了软件工程的管理技术:软件风险管理、软件生命期管理和软件工程技术的基本原则。本书的三个专题涉及软件工程的标准和软件度量,它们提供了技术管理层决策判断的工具和准则,是软件工程技术方法的重要组成部分。

第1册可作为学习计算机软件工程和信息系统工程的大学生、研究生的教材或参考资料。对从事软件工程管理人员来说,本书内容是提高管理水平的重要参考资料。它也可帮助从事技术工作的人员进一步提高软件工程能力。

图书在版编目(CIP)数据

现代软件工程(第1册):管理技术篇/周之英编著. —北京:科学出版社,1999

ISBN 7-03-007677-X

I. 现… II. 周… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字(1999)第 23974 号

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

1999年9月第 一 版 开本:787×1092 1/16

2002年7月第四次印刷 印张:17 1/2

印数:8 001—12 000 字数:397 000

定价:24.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

重印前言

经过两年多的发展,软件工程的现代特征更为清晰了,此次重印做了以下更动:

1. 把上、中、下三册改为第 1、2、3 册,为后继的“实例分析”等内容留下空间。

2. 更改了一些章节的次序:把软件体系结构从中册移至第 3 册。第 1 版中册的基本方法按开发涉及方法和知识的大致次序安排章节。实际上,体系结构实属正在发展中的新技术,这些新技术大部分可归结于复用的构造;因此,移到第 3 册更为合理。目前的分册反映对软件开发技术作一个大致分类:第 1 册是管理技术基础,第 2 册是软件过程各阶段的各种方法,第 3 册讨论软件产品的构造问题。

3. 除少量的错误纠正和删减外,内容上做了大幅度改进,主要变动涉及以下六个方面:

- 第一方面,尽可能包括软件工程中有重要影响的一些标准修订后的特点:如 SEI 的 CMMI,OMG 的 UML,OSI 和欧共体的一些软件工程标准等。

- 第二方面,增加一些科学基础性的内容:如第 2 册增加专题 D 软件工程的科学理论基础,涉及对模型、形式化、逻辑等的科学认识。这些基础知识对正确认识和应用软件工程技术有重要意义,也是软件工程技术进一步发展的基础。

- 第三方面,增加一些新兴的热点技术,如不确定性、轻量开发技术 XP、整理 Refactoring 等。

- 第四方面,适当增加和补充一些常识性例子和应用实例,以加深理解,并大大地充实第 3 册的设计方案和应用例子的内容。

- 第五方面,取材主要是国外的书籍和资料,但更突出自己对现代软件工程的认识来组织内容和观点,如以增加的一节“什么是现代软件工程”作为全书的开始。

- 第六方面,对章节的次序有所调整,按方法特点集中,如第 2 册中把结构化分析方法、结构化设计方法等集中在一起,把面向对象开发方法的各种技术集中在一起。第 3 册则集中于软件的构造。

本书是清华大学的研究生教材,也希望能对我国从事软件产业人员和软件工程教学人员有所帮助。由于时间、实践水平和范围的限制,一定还有许多不当之处,恳请读者提出批评、指正和建议。

周之英

2002 年 3 月 26 日于清华园

前 言

我们即将迎来的 21 世纪信息社会将高度地依赖于信息系统。事实上,在现代化的社会中已经很难想象没有“计算机”、没有“软件”会是怎样,面对着那无穷无尽的现实的和潜在的计算机应用需求,研究如何更快、更好、更多、更方便地开发出各种不同类型、不同目的的软件,这就是软件开发技术和软件工程技术所要解决的一个问题。

软件开发技术一直是软件工作者的主要研究方向。50 多年来,随着计算机系统的发展,软件开发技术也发生着变化。软件工程首先是为了解决软件危机而提出的。其目的是以成功的、卓越的开发经验来指导,通过类似于工业化的管理,把一般程度的开发人员的水平提高到优秀水平。软件开发技术的巨大成就,已经使软件开发不再是少数逻辑天才或专家的专利,而是广大用户可以参与和直接开发自己的应用项目。因此,软件工程技术 and 设计方法将会受到更多人的关注。

20 世纪 90 年代以来,软件工程不仅从方法论的角度为管理人员和开发人员提供可见的结构和有序的思考方式,而且大量的成功软件总结出的设计经验,使软件开发人员在面对新的项目时,不必从头做起,而可以充分利用设计模式、框架、部件库等。网络计算环境提供了经济全球化的新倾向,硬件技术以每 6 个月为周期的速度在发展,这些新的需求要求软件能具有充分的适应能力,去适应各种不同类型的连通和变动要求。

因此,老的软件工程教学体系已基本不能反映这些新技术和新需求的现状。从 1996 年我开始逐步对清华大学研究生校级学位课“软件工程技术与设计”的教学内容进行更新。教学的基本方针强调软件技术发展的变动性。我认为在急速变化的技术与社会环境中,无法想象还坚持不折不扣地照搬以前的成功经验会在新环境下继续成功,也无法想象不了解过去条件下的成功与失败,就可以迅速地创造出全新的方法与产品。因此,当前“软件工程技术与设计”的新的教学内容是以软件开发技术的发展史为纲,试图说明每种软件工程技术发展的原因、解决的问题及局限性,使学生不仅学到技术知识,更强调能根据具体情况灵活应用,甚至创造性地推动技术的更进一步发展。目前“软件工程技术与设计”课程讲课大纲如下:

- 软件开发技术发展史
- 软件危机及风险研究的重要性
- 软件工程技术方法的基本原则
- 软件过程的 CMM 模型
- 软件过程改进的实例分析
- 软件过程改进现状
- 需求工程的意义及现状
- 重要的需求分析及规格说明技术(功能性为主,结构化方法)
- 重要的需求分析及规格说明技术(控制为主,状态机, Petri Net)
- 重要的需求分析及规格说明技术(形式方法)

- 软件体系结构研究的意义及现状
- 基本软件体系结构
- 软件设计方法的发展概述
- 面向对象设计方法概述
- Use Case OOD
- Design Pattern
- 软件过程标准化问题
- CORBA(软件部件接口标准)
- DCOM/COM(OLE)(软件部件接口标准)
- 软件度量
- 实例分析(实时系统的设计)
- 自选(必须与教师协商)

当然,现在学生上计算机课已经不满足于单单在课堂上学习理论。理论如果不与实际应用相结合,是无法深刻理解和应用的,更难于发扬创造精神。但由于条件的限制,这一点还有待于今后的改进。

本书是我多年从事软件工程教学的总结。全书共分为五部分:第一部分是总论,涉及一些全局性的问题,如软件开发技术发展史、软件风险、软件工程技术的基本原则和软件生命期过程及改进问题。第二部分讨论与需求工程有关的各种问题。第三部分是与设计系统的构造有关的问题。第四部分涉及分布系统,它对网络计算环境有特殊意义。第五部分是三个专题,其中有关标准和软件度量是更高层次上的问题。我们说,大部分软件技术和方法提供了软件的可见性,标准和度量提供了判断上的工具和准则,因此正文部分和专题部分是相辅相成的,但传统上更强调方法论。本书没有涉及编码和测试问题。这虽然也很重要,但前者有其他课程讨论,后者更需要针对性地讨论,所以暂时不纳入本书范围内。

本书分上、中、下三册出版。除了要全面学习软件工程的学生和从事软件工作的人员外,分册出版可以帮助他们(她们)更方便地找到最需要的内容。上册包括第一部分和三个专题。软件开发的管理层次的人员会对这些内容有兴趣。中册是需求工程(第二部分)和设计方法(第三部分的大部分内容)。软件系统的开发技术人员可以在其中找到灵感,获取进行软件需求分析和设计方面的可选用的方法。下册则是现代软件工程成就的集中体现。包括第三部分中代表了优秀的软件设计经验复用的重要途径——设计模式方面的内容和第四部分——分布系统方面的成果。对于在基本软件工程理论和方法方面有相当造诣,需要了解 20 世纪 90 年代以来最新发展的工程技术人员来说,一定会感兴趣。

选修本课程的清华大学计算机系和其他系的近百名研究生和进修教师——姚诚伟、裴亚民、韩松、曾伟林、刘波、肖奔放、张义、宋晖、徐伟华、黄扬清等,在学习本课程后收集了大量资料和报告。在本书的内容中,吸纳了其中一部分内容。没有他们的积极参与和共同讨论,本书是不可能现在就面世的。协助本书编写工作的还有清华大学计算机系软件教研组的谢若阳老师和计算机系的部分大学生。在此一并表示感谢。

由于目前国内有关软件工程技术与设计方面的资料比较缺乏,因此本课程新的教学内容大都参考国外的书籍和资料。当前的技术发展可说是日新月异,而教学任务要求尽快

把教学内容整理出版,以供急需。因时间和水平所限,一定有许多不周到和不准确之处,恳切希望读者提出批评和建议。

清华大学计算机科学与技术系 周之英

1999 年 4 月 5 日于清华园

目 录

重印前言

前言

第一章 软件开发技术发展史	1
1.1 什么是现代软件工程	1
1.2 信息社会与计算机	3
一、信息社会的基本特征;二、计算机的特点与应用;三、计算机硬件的发展——计算机的过去,现在与将来;四、软件、硬件与计算机应用之间的关系	
1.3 软件开发技术发展史	5
一、20 世纪 40~50 年代;二、20 世纪 60 年代;三、20 世纪 70 年代;四、20 世纪 80 年代;五、20 世纪 90 年代;六、小结	
1.4 软件开发发展过程中的大事记.....	17
第二章 软件危机及软件风险研究	25
2.1 软件危机现象.....	25
一、软件价格高;二、难于控制开发进度;三、软件工作量估计困难;四、质量低;五、软件修改,维护困难;六、软件危机	
2.2 软件危机的部分原因分析.....	30
一、规模大;二、复杂性高;三、软件灵活性的分析;四、影响软件生产率与质量的因素十分复杂	
2.3 软件的风险问题.....	37
一、什么是风险;二、风险分析;三、风险管理;四、进行风险控制;五、常用的风险评估方法;六、实例:订购软件产品中的风险管理;七、不确定性理论与风险决策简介;八、小结	
第三章 软件生命期过程	56
3.1 软件的生命周期.....	56
一、引言;二、软件开发模型的发展;三、当前发展情况	
3.2 软件过程改进的 CMM 模型.....	64
一、基本概念;二、软件组织的成熟与不成熟;三、能力成熟程度模型的内容;四、软件过程各成熟级的可视性分析;五、CMM 的内部结构;六、主要实践活动;七、CMM 的应用;八、有关 CMM 的问题	
3.3 CMM 系列	85
一、CMMI;二、人员 CMM;三、软件获取 CMM;四、过程改进的评估方法 SCAMPI(SM);五、团队软件过程 TSP 和个人软件过程 PSP;六、与 CMM 有关的其他产品	
3.4 日本 OMRON 公司的软件过程改进实例	100
一、OMRON 公司软件过程改进概述;二、项目概况;三、调查发现;四、软件过程改进工作方法概况;五、OMRON 公司软件过程改进工作小结;六、结论	

3.5	微软的软件开发过程简介	108
	一、微软的特点和成就;二、“同步稳定”方法;三、开发阶段的划分;四、微软的开发及交付产品的策略;五、评论;六、“同步稳定”方法与传统方法的比较	
3.6	轻量开发方法 XP	113
	一、轻量方法学和 XP 的哲理;二、XP 只有少量的规则和实践;三、从 XP 中我们主要学些什么呢;四、XP 得以成功的领域;五、小结	
3.7	比较冲突与发展	119
	一、比较和冲突;二、分析;三、对 CMM 的改进	
第四章	软件开发的基本原则.....	126
4.1	普遍适用的原则	127
	一、基于常识的比喻;二、软件工程的社会性;三、严格性和形式化原则;四、分隔原则;五、模块化;六、抽象原则;七、预期变动;八、通用/复用性原则;九、演进式原则	
4.2	按专题分类的一些原则	142
	一、有关软件质量的一些原则;二、有关原型系统的一些原则;三、有关文档的一些原则;四、有关新技术与软件工具的一些原则;五、有关需求工程的一些原则;六、有关设计的一些原则;七、有关编码的一些原则;八、有关测试的一些原则;九、有关管理的一些原则	
专题 A	软件工程标准.....	152
A.1	软件工程标准的级别分类	152
	一、国际标准;二、国家标准;三、行业标准;四、企业规范;五、工程规范;六、市场	
A.2	IEEE/EIA P12207 信息技术——软件生存周期过程	155
	一、IEEE 软件工程标准的制订过程;二、ISO/IEC 12207 概述;三、IEEE/EIA P12207	
A.3	有关软件工程标准目录	185
	一、中国国家标准目录(表 A-2);二、中国国家军用标准目录(表 A-3);三、IEEE 标准目录;四、ISO 标准目录	
专题 B	软件度量.....	194
B.1	概述	194
	一、什么是度量;二、软件度量的目标;三、软件度量研究的范畴	
B.2	度量的基本理论	198
	一、直接和间接度量;二、度量理论——表示法;三、软件度量的框架	
B.3	软件基本度量	202
	一、过程度量;二、产品内部属性度量;三、度量产品的外部属性;四、资源度量;五、例子	
B.4	软件度量的一些方法	217
	一、费用和工作量估计;二、生产率度量和模型;三、数据收集;四、质量模型和度量;五、可靠性模型;六、性能评价和模型;七、结构和复杂性度量;八、GQM 法(Goal/Question/Metric)及 GQ(IM 法(indicator));九、统计工具;十、主要度量方法汇总:文档,复杂性和规模;十一、实验设计和分析	
B.5	软件度量的实施	235
	一、确定公司/项目的目标;二、例:确定软件质量度量;三、执行软件质量度量;四、分析软件度量结果;五、确认软件质量度量;六、软件度量实践(PSM);七、结束语	

专题 C 软件过程改进相关的一些标准 252

 C.1 ISO 9001 252

 一、ISO 9001 简介;二、ISO 9001 与 CMM 的比较;三、ISO 9001 与 CMM2、3、4 级差距的分析;四、ISO 9000;2000

 C.2 软件过程评估和改进 BOOTSTRAP 259

 一、BOOTSTRAP 的主要特征;二、参考模型;三、评分;四、Bootstrap 的步骤;五、小结

 C.3 SPICE 263

 一、简介;二、SW-CMM 与 BPG 结构的比较;三、最终目标;评估结果的可比性

参考文献..... 268

第一章 软件开发技术发展史

1.1 什么是现代软件工程

什么是软件工程？希望改变小作坊式的软件开发方式，转向工业化（基于物质世界客观规律，如物理学、力学、机械、建筑学等），制订工程方法规范，达到生产规模大、效率高、可控的质量和进度。软件工程意味着使软件开发从基于个人才能小作坊式（一种易错、低速、不稳定但具有创意能动作用）主观行为转到基于物性客观（可重复）规则的活动：易学而趋向自动化的过程。

软件工程基于项目和产品，追求超越项目、产品、开发组织、开发团队和个人。科学的特征是：可观察性、公开、可重复。软件工程的基础是客观性，代表软件开发的共性，软件工程的客观性反映在软件工程的一切活动环绕着事实进行。软件工程重视文档。文档记录形成开发中事实历史：反映人对项目的理解和项目进展的客观表示，分析和改进的客观基础。

虽然历史似乎会重复，但历史总是向前发展。我们只能根据软件工程的历史去预测未来发展，学习和掌握软件工程使软件工程师具备历史的观点：尊重事实，包括过去的历史经验和现状，正确认识自己和别人的情况。

软件工程的理论、方法、经验和实践、各种规范标准不同于物质世界的理论等对应物。软件开发强烈地依赖具体环境和执行人。由于人和机器不同，是不重复的，不同环境的人的行为活动从本质上难于具有类似“物”的重复性。由于难于刻画具体环境、条件、执行人的才能行为，所以基本上目前软件工程所有理论是不充分的，绝对重复难。不能仅依靠历史去计划未来，在变动的环境中，需要人去识别项目的个性，对差异做能动性调整。

经济全球化进程中的技术和环境的多样性、复杂性、迅速变化和不确定性，使用软件工程的人群的广泛性，要求软件工程具有简单性、实用性、有效性等特性，但它们之间存在极大矛盾。成功的具体项目始终不断出现，反映了采用良好的处理项目和环境的主要特征的技术：针对具体问题强调片面、局部，甚至极端措施，也可能是某种情况下的成功之道。不同项目、环境特征不一定一致，甚至可能有冲突。存在不同的、甚至相互冲突的观点和方法成为现代社会中的客观事实。在当前现实环境中，软件工程想一成不变地用一个方法解决一切的时代已经结束。于是不单纯强调物性规律，突出人的能动因素，承认方法的多样性是“现代软件工程”的特征。全面综合地、机械地推广软件工程技术会有一定难度。

软件工程主体建立在人类思想/行动常识和成功软件项目经验（局部证实）的基础上。在面对复杂、有冲突的问题时，人的主观能动作用尤为重要，需要对冲突和困难进行理

性的思考：包括认识极端方法（强调某种片面局部，或追求全面、细节，难免超越控制能力）的作用，通过合作和交流寻求合理地解决冲突的方法，获得“满意解”。懂得这种处理的局部性、多样性和暂时性的特点，有发展改进的必要。尊重常识，切忌盲目性，养成积极思考的习惯，具备发展（改进和创新）的观点是优秀软件工程师的特点。

过程：从不知道到知道（一些技术），知道以后要真正理解，理解后要实践，开始不一定能正确实践，正确地实践后，可能发现技术不足之处，得到创新的源泉。

开始，不知道事物规律，工作无约束，就是没有规律的无序的工作。认识一些规律，可防止一些错误，可使道路走得正确一些。起初，这样的规律只是一些限制。进一步认识后，可规范成一个过程，等于规定一条道路，工作就更具体了。下一步，通过认识和解决规律的不足之处，发展突破性工作，在新条件下得到新的规律。这是一个循环的、不断的过程。

不知道→知道→理解→实践→正确实践→创新

无规律→规则→过程→成功突破→新条件下新规律……

看看中国古老的太极图所产生新意：现代软件工程反映人（主观）与物/自动机械（客观）的关系（假如太极图中人代表黑的一半，机械代表白的一半）。人的工作总会包含已经认识的、可重复的、机械性的东西，可以做成一些机械。当人所认识的、可重复的、可机械化的部分成为机械，那么代表人的黑的一半核心里面包含了白的、机械性的东西，逐步变化，转换成白的（机械）。人的主要工作都通过机械解决成自动化，则已经转到另外一半了。但是另外一半里，肯定有对已经自动化（机械）不满意的东西，需要人做的工作：黑的小圆点——需要人的创意，所以白的、自动的里面还包含黑的，不能够一片白。条件变化后，当黑的小圆点部分成主导时，转为代表强调人的特征的另一半了。而人把问题解决后，认识提高，又创造出新的自动工具。所以转动地观看太极图，反映了人与机器的关系、人和自动过程在不同的层次上不断演化。

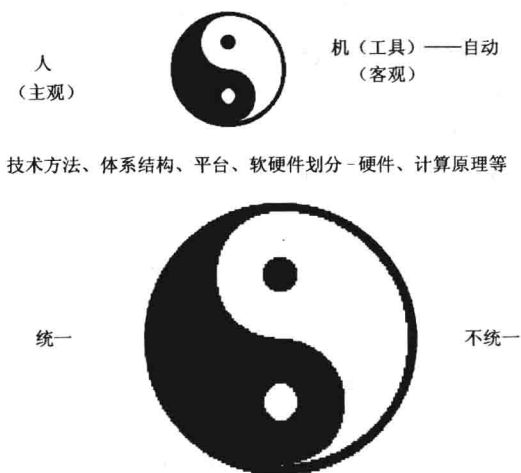


图 1-1

人与机器的关系可用太极图描述，用同样的方法来观看技术方法、体系结构、平台、软硬件划分、硬件、体系结构、计算原则的统一和不统一关系，它也是符合太极图这样的规律，可以说太极图反映了软件工程的过去和未来。

总之，早期的软件开发依赖人的主观行为，传统的软件工程的最高目标等同于“物质”工程，追求物性、固化客观规律。现代软件工程承认多样性和变动性，主、客观的相互影响：相对稳定、演变和突变。在变化的环境中，人与物、人和机器（方法、工具、标准、自动化）的关系在不同层次不断地循环发展着。

关键是观念转变：固化观点转变为“不断有条件地继承和扬弃（已知的、成功合理思想和实践）；不断地认识和演变地处理未知（不确定性）”。

1.2 信息社会与计算机

我们已经迎来了 21 世纪的信息社会。农业社会里，生存资源是土地，个体的农业劳动和手工业劳动是社会生产的基本形式，劳动生产率很低。在工业社会中，生存资源是资本，大规模的机械化与电气化的工业生产是社会生产的主要形式。随着科学技术的进一步发展，特别是“3C”，即 Communication（通信）、Control（控制）和 Computer（计算机）的高度发展，将把人类由工业化社会带入了 21 世纪的信息社会。

一、信息社会的基本特征

（1）战略资源是信息：信息的发掘与增加、信息的管理、开发、流通和利用以及更新将是提高社会劳动生产率的主要手段。

（2）知识爆炸：在信息社会里，知识更新快、老化快，呈现“知识爆炸”的现象。据有关资料估计，在信息社会里，科学技术信息的年增长率将达到 40% 以上，每 5~8 年增长 10 倍。

（3）脑力劳动为主体：在信息社会里，大多数人是从事信息的管理与生产工作，主要进行脑力劳动。

（4）核心技术与计算机有关：在信息社会里，无论个人、团体或企业，无论地区、国家或全球，无论文化、经济或军事，无论生产、管理、教育或休闲的进展，计算机都会或明或暗地起中枢神经作用。

二、计算机的特点与应用

计算机的特点主要有：

- （1）运算速度快。
- （2）精确度高。
- （3）具有“记忆”和逻辑判断能力。
- （4）程序控制自动完成操作。

计算机的应用非常广泛，已从数值计算领域发展到几乎深入到人类生产和生活的一切领域。计算机的应用领域还将随着计算机的进一步发展而扩大。可以预料到 21 世纪时，计算机将从目前基本上还是一种自成体系的设备，发展成一种到处都存在，但又看不见摸不着的“隐藏”在其他设备中的部件，使计算机完全融合在人类生活之中。

三、计算机硬件的发展——计算机的过去，现在与将来

软件离不开计算机这个硬件载体，所以，当我们来谈论计算机软件时，自然离不开计算机硬件本身的发展。

(1) 简单的计算工具及机械。在很久以前，人类进行计算是出自于现实的农业生产和简单的产品交换和分配的需要，为此人们需要记录某些数字和进行简单的计算。在公元前 1000 年前，人们通过在粘土上刻痕来记录数字。到了公元 13 世纪，中国人在算筹的基础上发明了算盘，通过一些串在轴上的算珠可以完成加、减、乘、除运算。可以说，这是世界上最早的较为成熟的计算工具。算盘一直沿用至今。而中国对算法的研究也早有记载，宋代数学家秦九韶算法是利用递归公式来计算多项式的值。

随着人类商业、制造业以及科学实验的发展，计算工作越来越频繁。在 1612—1614 年，苏格兰数学家 John Napier (1550—1617) 发明了一种帮助计算乘法的骨质拼条，它是一种辅助的计算工具。在 1642 年，法国科学家 Blaise Pascal (1623—1662) 发明了齿轮式加减法器，称为 Pascalene。当时他曾制造了 50 台。在此之后，德国的 Gottfried Wilhelm von Leibniz、法国的 Joseph Marie Jacquard 和 Charles de Colmar 都先后对机械式计算机进行了改进。1822 年开始，英国剑桥大学 Charles Babbage 在穿孔卡织布机的启迪下，开始了分析机 (analytical engine) 的设计，首创了包括现代计算机都具有的五大装置：输入、处理、存储、控制和输出，开创了近代机械式计算机研究的先河。1854 年，乔治·布尔出版了名著《思想法则的研究》。该书阐述了一种符号逻辑推理系统。即开创了逻辑代数——布尔代数 (Boolean algebra)。在布尔代数中建立了两种逻辑值“0”、“1”以及三种运算“与 (and)”、“或 (or)”、“非 (not)”，为现代电子计算机的发展奠定了数学基础。以后又相继出现了手摇式机械计算机和电动式机械计算机。

直到 19 世纪 40 年代，所有的机械式或其他形式的计算机都没有硬件和软件之分。所谓算法，或者说是计算原理都是通过机械的制造模型来确定的。如果想修改算法，那就只有重新制造一台新的机器了。好在当时计算机只用来进行固定而简单的计算工作，所以当时虽没有什么软件也可将就了。直到 1945 图灵在他的 ENIAC 报告中描述了存储程序在计算机上的应用。这样，从概念上，将用于算法的程序从硬件中分离出来，“程序”就成为了“软件”的雏形。

(2) 第一台电子计算机的诞生。世界上第一台真正意义上的电子计算机是诞生在 1946 年美国宾夕法尼亚大学的摩尔学院。字长为 12 位，每秒可进行 5000 次加法运算，有 18800 个电子管，1500 个继电器，占地约 150 平方米，重约 30 吨，耗电约达 150 千瓦。从此，计算机技术以人类历史上空前的速度飞速发展。短短的 50 年，计算机发展速度之快，就连人类自己都不敢相信。

(3) 1946—1958 年——电子管计算机时代。主要逻辑元件是电子管。形成存储程序计算机的架构。这类工作模式一直延续至今。这一时期的存储设备已发展为主要使用磁芯及磁鼓。

(4) 1958—1968 年——晶体管计算机时代。主要逻辑元件是半导体晶体管。使计算机的体积大大缩小，出现了以屏幕监视器及键盘输入方式的商用计算机。

(5) 1968—1971 年——中小规模集成电路计算机。集成电路的发展引发了超级计算机的研究。例如, 1968 年由 SEYMOUR CRAY 设计的 CDC7600 超级计算机的性能达到每秒 4000 万次浮点运算。

(6) 1971 年开始——大规模集成电路计算机。第一个微处理机即完全放置于一个芯片上的计算机由 Intel 开发出来了。

(7) 1980 年——个人计算机, 网络开始迅速地扩大到人类生活各个领域。

(8) 1990 年——计算机、网络与通信高度接合所代表的信息高速公路。

(9) 2000 年以后。微处理机将向微体系结构 (micro-architectures) 多处理机芯片 (CMP) 发展。其性能大大突破由数十亿晶体管构成的单一处理机。按照摩尔法则: 微处理器的性能每 18 个月将翻番。在芯片上的晶体管的数目每两年翻番。预计到 2010 年每个芯片上将有 800 亿个晶体管, 时钟频率可达到 2GHz。一个芯片上的功率将达到 180W。芯片要进一步提高性能的主要限制可能将是散热问题。

四、软件、硬件与计算机应用之间的关系

从第一台电子计算机问世以满足计算方面的日益增长的需要之后, 每次硬件技术的突破, 都为软件技术新发展提供了更为广阔的空间, 开拓了新的更广阔应用领域。计算机的应用领域从单纯的科学计算发展到军事、经济、科学、文化直到社会生活的各个方面; 进而要求计算机的运算速度不断提高, 存储容量不断扩大, 体积微型化。而计算机数量的巨增, 使软件系统从简单发展到复杂, 从小型发展到大型, 由封闭的自动化孤岛发展成为一种开放的系统。在开发方面: 从注意技巧发展为注意管理; 由单独设计发展为注意复用, 由少数天才的编程艺术发展为广大用户直接参与开发应用。这种由应用驱动而相互促进、激励的发展过程, 使人类社会文明进入了信息社会新的高度发展的阶段。

1.3 软件开发技术发展史

正如计算机硬件在 50 年内发生了极大变化, 同样, 对计算机软件的开发技术来说, 在观念及目标等方面都发生了很大变化。我们可以大致以 10 年左右来划分软件开发技术的各个阶段。分析与讨论每个时期的特点, 即每个时期的软件开发技术处理的对象、用途、目的、开发方法、认识、发展状况、突破、理论成就及目标。

一、20 世纪 40~50 年代

1. 软件开发技术处理的对象

软件开发技术处理的对象, 以处理机的机器码占据主要地位。机器码一般由指令操作码及数据码组成。即是由“0”及“1”组成的序列。软件开发称为程序编写。由于每个计算机的指令系统都单独设计, 没有规律, “0”与“1”组成序列的含义难于辨别, 难于记忆, 因此编写、阅读、调试程序都非常困难。

2. 用途

当时计算机的用途主要是科学计算或与军事有关的计算问题（如导弹表的计算）。世界上第一个电子计算机在宾夕法尼亚大学摩尔学院运行几个月后，就拆迁到马里兰州阿伯丁武器试验场工作，直到 1955 年退役。

3. 目的

在这时期的计算机没有装入任何软件。即我们现在称之为“裸机”的机器。裸机只能识别二进制代码。程序人员编写机器能识别的机器码程序的目的是“确定计算机硬件动作的序列”。使计算机能自动地执行程序编写人员要求计算机完成的计算任务。

4. 开发方法

由于当时每台计算机都是单独设计，机器的指令系统没有规律，计算机价格昂贵，懂得计算机的人很少，会用计算机的人更少。人们认为，只有逻辑天才才有能力施展高技巧，写出让计算机完成计算任务的程序。普通人是不敢问津的，也根本没有想到需要开发方法。到 20 世纪 50 年代，美国总共也只有 500 名程序员。

5. 认识

认为计算机的用途就是快速计算，用途很窄。当时认为计算机的主要用途是只要计算一些数据表就行，一旦计算完成之后，可将结果印发给全世界使用。所以全世界只需有 5 台计算机就足够了。

6. 发展状况

随着计算机体系结构成熟，商业上可以大批量地生产计算机，价格下降，应用范围很快扩大。特别，为了使大众及商业界能接受及理解计算机，人们在计算机应用方面作了许多尝试，以扩大计算机的影响。如：利用计算机预测美国总统选举结果；在日本，利用计算机进行照相机镜头的设计；甚至，开始设计交通管制系统等。这些软件是用机器语言编写的，程序员将指令和数据直接写到计算机的存储体中。当程序员在原有的程序中加入一条新的指令的时候，程序员必须亲自对整个程序作检查，以确定所有相关的指令和操作经过变动后仍然是正确的。当这些应用变得更为复杂时，编写程序十分困难。渐渐地人们发现机器指令和存储体地址可以用一些便于记忆的符号来代替，为了摆脱用机器码编程的困难，使程序人员熟记计算机的全部指令，出现了用指令符号来编制程序的办法。用指令符号编制的程序称为符号程序，在编制程序时，只要记住用英文名称缩写的指令助记符就可以了。例如，取数用 LDA、加法用 ADD 等。在符号语言的基础上，进一步发展就是出现了汇编语言。用汇编语言编制程序要比用机器的指令代码编写方便得多，大大便于检查和修改错误。而且，指令、原始数据和结果数据的存放单元可以由机器自动分配。

然而，计算机的内部结构是根据指令代码设计的。也就是说，它只能识别和理解用二进制代码表示的指令，不能识别和理解指令助记符。因此，人们用汇编语言编出程序

后，必须将此程序翻译为机器语言程序，称为目的程序（目标程序），机器才能执行。这个翻译工作十分繁琐，几乎完全是机械式的一一对应的翻译。因此，人们开发出专门的程序来完成机械式的翻译。这种程序称“汇编程序”。汇编程序是一种进行“编辑与翻译”的程序，计算机有了它，才允许用户在该计算机上使用汇编语言编制程序，称为汇编语言程序或汇编源程序。汇编程序是当时计算机必不可少的软件。

用汇编语言写的程序和用机器语言写的程序有相同之处，又有不同之处。相同之处在于程序主体部分几乎是一一对应的，不同的是 0、1 数码换成了符号，地址换成了可读的名字，另外还增加了关于工作单元和常数单元的成分。这些不同之处也正是汇编语言的优点，使得用汇编语言编写的程序好写、好读、好改。

由于汇编是依赖于机器的，因此称它为面向机器的语言。使用时必须了解机器的某些细节，如累加器的个数、每条指令的执行速度、内存容量等等。但也正由于它依赖于机器就可以与机器语言程序一样结合机器特点编出短小、高质量、执行速度快的程序。所以，时至今日，汇编语言仍起着重要作用。在一些计算机公司中仍用汇编语言编写系统软件，以保证高质量软件的功效。

机器码书写冗长且易出错，而汇编语言有一定改进，但仍依赖于机器，仍为“面向机器的语言”，使用时需了解机器的某些细节，从而使计算机的应用普及以及普通人学习计算机产生了巨大的困难。随着技术的增长以及程序开发任务的复杂化，单靠机器码与汇编语言已越来越难以适应程序开发的需要。同时，程序内部的控制日趋复杂。到 20 世纪 50 年代后期，人们发现按照某种规则来组织描述程序的助记符将会有助于程序的理解。例如使用一系列数学符号来表示数学运算等措施比单纯的汇编程序要容易理解。这种想法导致了一系列早期的高级语言的出现，其中包括现在我们仍在使用的 Fortran 语言。高级语言的出现和数据类型的使用使得编制复杂的软件成为可能。因此，面向应用的高级语言及相应编译系统的研究成为重要发展方向。

7. 重要技术突破

(1) 对时间-空间关系的认识有了提高，即认识到：可通过使用便宜的存储器来代替计算机硬件的逻辑功能。在电子计算机出现之前，人们设计专用的硬件来解决特定的问题，大部分问题都是通过直接的方式来实现的。后来，发现可以通过组合方式来处理许多问题。因此，经费决定了所能解决问题的复杂程度及规模。当时许多人认为只要有足够的钱来构造硬件，就可解决几乎任何问题。而有了计算机，带来了各种层次的存储设施，如磁芯、磁鼓、磁带、磁盘、纸带、穿孔卡片等等。其价格又便宜，数量又充足。通过把信息放在存储介质（空间资源）上，再多次利用，这样就代替了部分昂贵的硬件逻辑功能。

(2) 迭代-反复使用的子程序的利用。从此，计算机解决问题的方式不仅仅是意味着可以利用各种功能的组合，而且还可以通过反复利用子程序，来解决规模巨大的问题。而如果只依靠过去那种硬件组合的方式，想解决这类问题是无论如何无法承受相应的巨大经济压力的。

(3) 因此，人们思想观念发生了巨大变化。从只要有“足够的资金”就能解决困难而复杂的问题转变为只要能容忍时间上的“等待”就可以解决几乎任何问题。从此，为了