

第一章 软件技术概述

1.1 计算机与计算机系统

人类在认识自然、改造自然的过程中,曾经创造过各种各样的计算工具。随着科学的进步,尤其是电子技术的迅猛发展,在1946年诞生了世界上第一台电子计算机。1946年6月,美国数学家冯·诺依曼(Von Neumann)提出了一个“存储程序”的计算机方案。这个方案确定:计算机由控制器、运算器、存储器、输入设备、输出设备组成;将指令和数据同时存放在存储器中。该方案奠定了现代计算机结构理论,促进了计算机的迅猛发展,被誉为计算机发展史上的里程碑。一直到现在,计算机仍没有突破冯·诺依曼结构的框架。

电子计算机能够快速而准确地进行各种数值运算,又能完成大量的数据处理,也能够在生产现场实现复杂生产过程的自动控制。目前,计算机的应用范围已波及到人类活动的各个领域。

计算机系统包括两大部分:硬件系统和软件系统。

一、硬件系统

计算机的硬件系统包括计算机的全部硬设备和功能部件。如输入输出部件,信息的存储部件、工作控制部件及加工数据的运算部件等。硬件系统是计算机的物质基础。

二、软件系统

计算机的软件系统主要指为使计算机完成某些功能而编写的程序(当然包括文档)集合。软件系统是计算机的灵魂。如果没有计算

机的软件系统去指挥、协调计算机去工作,计算机将是一堆废铁。

1.2 软件的概念及分类

1.2.1 软件的概念

计算机软件的出现是五十年代后期的事情,对于什么是软件即软件概念的定义,经历了一个演变发展的过程。很多人认为,软件就是计算机所使用的各种程序。实际上这种说法并不确切,程序只是软件的一部分。

按照现代软件工程的观点,软件的定义如下:

软件是程序以及开发、使用和维护程序所需的全部文档。

1.2.2 软件的分类

计算机软件系统内容丰富,它是计算机发展中最为活跃的因素。世界各国都投入了大量的人力和物力开发、研制新型软件。计算机软件的技术成就标志着计算机应用和发展的水平。计算机软件一般可分为两大类:一类是系统软件;另一类是应用软件。

一、系统软件

系统软件是管理、监控和维护计算机资源的软件。一般来说,系统软件是指具有下列功能的软件:

1. 高效使用硬件功能;
2. 提供各种应用的通用服务功能;
3. 支持计算机系统的故障预防、异常处理;
4. 保护数据和程序等信息,支持计算机系统正确、安全地运行;
5. 支持计算机与其它计算机或设备的通信处理等。

如果对系统软件进行分类,它主要包括以下几种类型:

- (1)控制软件:主要完成对作业、任务、资源、数据及故障的处理

等进行控制和管理。它是操作系统的主体组成部分。

(2)通信软件:主要完成计算机通信网络各层间的通信规程的处理并进行通信故障检出和网络运行控制。

(3)语言处理软件:它的作用是将用户用程序设计语言书写的源程序翻译成机器语言。主要包括:汇编程序、编译程序和解释程序等。

(4)服务软件:它主要是对程序的执行和大量数据操作提供通用的服务;如:连接程序、编辑程序、分类合并程序等等。

(5)数据库管理软件:主要完成各种数据的统一管理,使其规范化。

(6)软件开发支持软件:支持提高软件开发的效率和质量。其中包括:系统分析工具软件、代码生成支持软件、软件测试支持软件等。

二、应用软件

应用软件是用户为使用计算机解决实际问题所开发的软件的总称。应用软件极为丰富,为了满足各类用户的实际需要,每时每刻都有人在研制、开发和推出一些应用软件。目前微机究竟有多少应用软件,恐怕难以说清。

应用软件可分为两大类:一类是不分业务、行业而可以使用的公共应用软件;另一类是按业务、行业分类的应用软件。

1. 公共应用软件

公共应用软件大致可分为如下几个种类:

(1)数据处理类软件;

(2)进行声音、图形图像、文献等信息处理的软件;

(3)进行各种信息检索处理的软件;

(4)人工智能方面的软件;

(5)计算机辅助设计与制造、计算机辅助教学、计算机辅助分析以及决策支持系统等方面的通用软件等。

2. 按行业、业务分类的应用软件

这种软件专业性很强,一般不能通用,仅可用于某一行业或业务

领域。如：应用于医疗保健、教育、服务、法律、军事等方面的应用软件。

1.3 软件开发技术

1.3.1 软件工程概述

随着计算机应用水平的不断提高，人们逐渐认识到，软件在计算机应用中起着举足轻重的作用。任何部门或单位要想用好计算机都必须把软件问题摆在优先地位考虑。许多业务部门的技术人员和管理人员现已体会到，为使计算机更加有效地解决自己专业领域里的具体问题，就应自己组织力量进行软件开发。这已是计算机应用工作的一个必不可少的组成部分。总之，形势要求我们对软件技术有更进一步的认识。

一、软件工程的概 念

计算机软件是一个逻辑的而不是物理的系统，它具有一些与硬件显著不同的特点。开发软件的主要工作集中在定义、开发和维护等方面。60年代以来计算机语言和编译技术的成果为程序设计工作带来了方便，又为计算机的广泛应用开辟了道路。然而，我们必须看到软件开发技术尽管出现了许多新事物，但和硬件技术的飞速发展相比，它已经明显地落后了。多年来被人们沿用的手工作坊式软件开发方法一直未曾被突破，这种传统的方法使得许多发现了的问题得不到及时解决。在软件产品的质量、成本以及开发时间等方面无法满足人们的需求，极大地影响着软件行业的声誉。一些典型的问题表现在软件产品质量低劣（如工作可靠性差、用户难以使用、难以维护等），研制软件成本太高，不能按时交付使用等。这些现象一般称其为“软件危机”。造成“软件危机”的原因是多方面的，从认识方面看，程序设计曾经被当作是一种任人发挥创造精神的技术学科。不少人以为，写

出的程序只要能在计算机上得出正确结果,程序如何编写不应受到任何约束。尽管程序很难被别人看懂,但只要含有精心设计的程序技巧,就是高水平的程序。殊不知这样可供“孤芳自赏”的程序在测试、修改以及维护中会遇到很多困难,而且常常是重新产生错误的祸根。进一步分析技术和管理上的原因,包括:未能充分理解和正确表达用户的需求;采用了不适当的开发方法;没有或未能认真遵循开发规范;测试方法不当;项目管理不力;不重视资料、文档工作等等。

“软件危机”出现于 60 年代末期,人们已逐渐认识到,那种手工作坊式软件开发方法已经行不通了,必须寻找新的技术来指导软件的开发。1968 年在北大西洋公约组织的一次学术会议上首次提出了“软件工程”这一名称。在关于怎样对待编写程序中的 GOTO 语句的使用问题上展开了激烈的争论,从而导出了结构化程序设计的概念。70 年代中期又提出了软件生命期的概念。80 年代以来,软件工程的思想得到系统的归纳和整理,提出了许多工程化方法。在软件开发环境、软件标准化等方面都取得了显著的成果。总之,为了摆脱软件危机,把软件开发技术向前推进一步,人们认识到出路只有一个,就是软件研制一定要走工程化的道路。

软件工程就是采用工程的概念、原理、技术和方法来开发和维护软件,把经过实践和时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。它是指导计算机软件开发和维护的工程科学。软件工程的指导思想是以处理工程问题一样的方法处理软件生产的全过程,利用较少的投资获得高质量的软件产品。

研究软件工程的学科称为软件工程学。软件工程学既是边缘学科,又是一种综合性学科。它包括:计算机科学、系统工程学、管理学、经济学、人体工程学以及心理学等。

软件开发技术和方法的发展,大体经历了三个时代,即程序设计时代、软件时代、软件工程时代。各时代的特点如表 1-1 所列。

表 1-1 软件工程的发展

项 目 \ 时 代	程序设计时代	软件时代	软件工程时代
产品名称	程 序	软 件	软件商品
生产方式	个 人	课 题 组	软件生产组织
软件质量	取决于个人	取决于课题组	软件生产管理组织 可靠性评价规范
设计对象	以硬件为中心	硬件和软件结合	以软件为中心
开发工具	无	个体式开发工具	有软件开发工具 归系统所有
维护手段	无	不重视	重点考虑维护
设计方法	无	自上而下	有一套设计方法， 以结构化设计为主

二、软件的生命期

如上所述,所谓软件工程是以处理工程问题一样的方法处理软件生产的全过程。世界上任何事物都有它的发生、发展和消亡的过程。任何的工程产品(如机械、化工、纺织、建筑等等)对于新产品的生产都要经过定义、可行性研究、分析、设计、制造、测试、使用维护等几个阶段。

软件的生命期是指从概念的形成直到所开发的软件在充分使用之后完全失去使用价值为止的整个过程。它大体可分为三个时期,即定义、开发和维护时期。每个时期又可分成若干个阶段,即定义、可行性研究、分析、设计、编程、测试、维护等几个阶段。每个阶段都按产品生产的要求,确定其具体内容和任务,并产生一定规格的文档送下一个阶段,下一阶段在前一阶段的基础上继续进行工作。

各阶段的主要任务如下:

1. 软件定义时期

(1) 软件项目规划阶段

其任务是确定开发的总目标,给出所开发软件的功能、性能、可靠性以及用户接口等方面的设想。由软件开发人员和用户合作进行可行性研究,并对可利用的资源、开发成本、效益和开发进度等进行估计,制定实施计划供审查。

(2)分析阶段(即需求分析阶段)

着重解决该软件应做什么的问题,这就必须对软件的需求进行精确定义,对涉及到的数据流进行分析。写出功能说明书(软件需求说明书)及用户手册的初稿。

2. 软件开发时期

软件开发时期集中解决怎么做的问题。这时软件开发人员要考虑软件的总体结构、数据结构和程序结构,以及如何用计算机语言实现等问题。这一时期可分为三个阶段:

(1)软件设计阶段

设计阶段的工作是在分析阶段的成果基础上进行的。其工作比较复杂,在一般情况下可分成两大步:一是总体设计;二是详细设计。总体设计完成软件的模块结构及模块间的接口设计;详细设计完成模块结构的细化,并确定每一模块的内部实现算法。软件设计阶段的工作成果是软件设计说明书。

(2)程序编制阶段

编程阶段的主要工作是以某一特定的程序设计语言表达设计阶段所确定的算法。

(3)软件测试阶段

对软件进行测试,发现并排除存在的错误。

3. 软件维护时期

也可说软件维护阶段。这是软件生命期中最长也是最后的一个阶段。其任务是通过软件的运行,发现和排除软件中存在的错误,并根据运行环境的变化,对软件进行改进、扩充。

图 1-1 表示了三个时期的关系。图 1-2 为软件生命期瀑布模型。软件生命期各阶段都有明确的工作内容,并规定好什么是一个

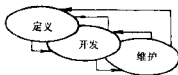


图 1-1 软件的三个时期

阶段任务的完成检验条件。按时间顺序严格执行,前一阶段的任务未完成决不能往下阶段进行,这就是在图 1-2 中描述的软件生命期瀑布模型。由于按此模型进行软件开发的阶段性强,在每一阶段结束前,要

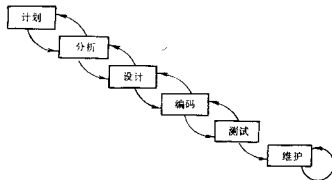


图 1-2 软件生命期瀑布模型

进行阶段评审。如发现问题必须暂停向下进行,而应返回上一阶段找出问题并加以解决。正如图 1-2 所示,事实上软件开发过程中的反复常常是不可避免的。

三、软件文档

对于软件产品应提供的文档,不同部门、不同系统有很多标准和规定,我们不去一一进行说明。

按照军标 GJB438-88《军用软件文档编制规范》列出十三种文档:

- 可行性研究报告
- 项目开发计划
- 软件需求说明书

- 数据要求说明书
- 概要设计说明书
- 详细设计说明书
- 数据库设计说明书
- 用户手册
- 操作手册
- 程序维护手册
- 测试计划
- 测试分析报告
- 安装实施过程等。

当然,对于规模不太大的软件而言,上述的某些文档可以适当合并。

1.3.2 软件开发方法综述

软件开发方法总体上可归为以下两类:结构化方法、原型法。

1.3.2.1 结构化方法

又称为生命周期法或结构化生命周期法。这是目前最常用也是最成熟的软件开发方法。使用此开发方法成功的范例很多。这种方法的优点是有严格的一套开发程序,各开发阶段都要求有完整的文档记录。

结构化方法在开发软件时,需要在分析阶段采用结构化系统分析法,软件设计阶段采用结构化系统设计方法,编程采用结构化程序设计方法。在开发软件系统的整个过程中,严格按照软件生命期的各阶段进行。系统开发生命周期的划分实际上是系统开发的一种方法论,或者说是组织工作的过程。特别是对计算机信息系统这样的复杂系统工程来说将有许多人参加研制,有许多任务要完成。因此,必须有人组织、协调、控制。所谓控制,应包括职责的分工控制、费用预算的控制、计划的控制以及质量控制等。系统开发生命期的每一步都

必须有检查、评价及决策。

下面将结构化方法的主要特点进行简要论述。

一、强调需求定义的重要性

结构化方法(SDLC)表述了随着特定开发阶段的展开,逐步实现系统的开发过程。已完成的阶段可以进行项目的审查、控制、管理等。实际的开发步骤、阶段的标志以及交付的文档,因具体实现技术不同可能有差别,但其指导思想和总体目标是一致的。

如果不管方法的来源如何,所有结构化方法都强调需求定义对于系统或项目的成功绝对重要,一开始对问题就没有一个清晰的理解,那就谈不上会有什么效率和成果。

从实用上讲,一般认为,需求定义必须有下述的一些属性,它们是:

- 完备的:所有需求都必须加以适当的说明。
- 一致的:需求之间应该没有逻辑上的矛盾。
- 非冗余:不应有多余的、含混不清的需求说明。
- 可理解:参加的各方应以一种共同的方式来解释和理解需求。需求是明确可辨的。
- 可测试:需求必须能够验证。
- 可维护:文档的组织应该是可灵活修改和易读的。
- 必要的:所规定的需求必须是用户所需要的。
- 正确的:需求应是准确完整的。

如果需求是不完全、不合乎逻辑的、不贴切或易使人发生误解的,那么,不论以后各阶段的工作质量如何,都必然导致一场灾难。由此可见,需求定义是系统开发成功与否的关键。

许多成本分析表明,随着开发生命周期的进展,改正错误或在改正错误时引入的附加错误的代价是按指数增长的。见图 1-3 所示。

研究表明,百分之六十到百分之八十的错误来源于定义。见图 1-4 所示。

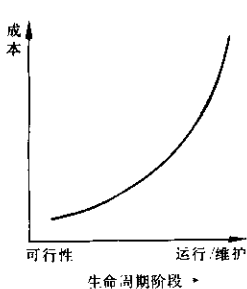


图 1-3 改正—费用曲线

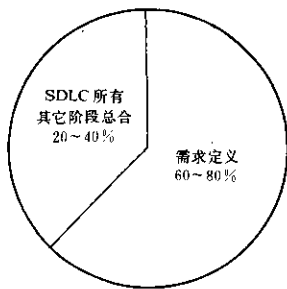


图 1-4 系统错误来源

二、严格的/预先定义方法

结构化方法的需求定义是采用严格的/预先定义的方法。从概念上讲，一个负责定义的小组试图完全彻底地预先指出对应用来说是合理的业务需求，并期待用户进行审查、评价、认可，并在此基础上进行下一阶段的工作。

严格的/预先定义方法是在如下假设的前提下形成的：

1. 所有的需求能被预先定义。
2. 修改定义不完备的系统，代价一定是昂贵的、困难的。
3. 参加项目的研制者之间能够进行清晰而准确的通信和交流。
4. 静态描述/图形模型对应用系统的反映是充分的。
5. 严格方法的生命周期的各阶段都是固有正确的。

三、结构化方法的软件生命期分为下面几个阶段：

- 软件项目规划阶段
- 需求分析阶段
- 设计阶段

- 编码阶段
- 测试阶段
- 维护阶段

1.3.2.2 原型法

原型法是针对结构化方法(或生命周期法)的局限性而发展起来的。它的基本做法是根据用户提出的概略需求,快速地构造一个最终系统的工作模型并逐步发展此模型。

原型法的主要特点如下。

一、强调需求定义的重要性

和结构化方法一样,原型法同样强调需求定义的重要性。

二、原型定义方法

其假设的前提是:

1. 并非所有的需求在系统开发以前都能够准确地说明。
2. 有快速的系统建造工具。

直到最近,大系统的原型化才成为可能。目前,用于完成原型化的较好工具可归纳为:

- 集成数据字典
- 高适应性的数据库管理系统
- 非过程的报告书写器
- 非过程查询语言
- 屏幕生成器
- 起高级语言
- 自动文档编排
- 原型人员工作台等等

如果生命周期可以浓缩到需求分析(定义)阶段,而且能用易适应的软件加以实现的话,人们就不会被改正费用曲线所支配。

3. 项目参加者之间通常都存在通信上的障碍。
4. 需要实际的、可供用户参与的系统模型。
5. 需求一旦确定,就可遵从严格的方法,即一旦通过建立原型并在演示中得到明确的需求定义后,即可运用行之有效的结构化方法来完成系统的开发。
6. 大量的反复是不可避免的、必要的、应该加以鼓励。

三、原型法的基本工作过程

基本工作过程如下:

1. 项目规划
2. 定义基本需求
3. 开发工作模型
4. 演示、求精扩充
5. 修改/增加
6. 工作模型用户接受否? 若被接受,原型完成。否则,转 4。

1.3.2.3 结构化方法和原型法相结合

由于所研制的软件系统的特点和所面临的情况不尽相同,在进行软件开发过程中使用什么方法要根据实际情况进行选择。特别是研制较大型的系统,结构化方法的各种技术比较成熟,其中包括:企业系统规划方法、战略数据规划方法等等。而原型法的各种技术虽然发展很快,但总的来说还没有统一的标准和规范,所以,单独使用原型法开发系统的可能性,在目前来说并不是特别大。在一般情况下,开发软件系统所采用的方法,要么纯采用结构化方法,要么采用结构化方法和原型法相结合。

下面是结构化方法和原型法结合开发软件系统的过程(图 1-5)。

以上对系统的开发方法及其特点进行了概述,至于软件开发的具体技术本书就不详述了,有兴趣的读者可参阅有关软件的专

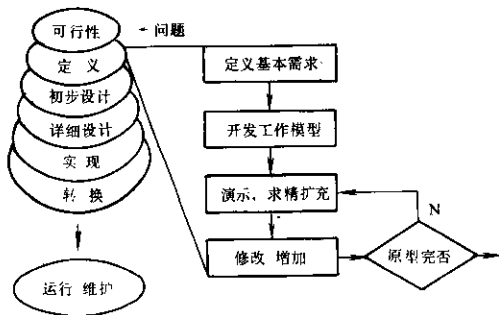


图 1-5 结构化方法和原型法结合

著或专门研究软件开发技术的书籍。

1.4 软件开发环境

所谓软件开发环境是指在基本的系统软件的基础上,为支持软件的开发和维护而提供的一组工具群。软件开发环境的设计目标是提高软件生产率和改善软件质量。

1.4.1 软件工具

所谓软件工具是指能够支持软件开发的使用方便的某种软件群。即软件工具本身也是软件,是能对软件开发和维护带来方便的起到工具作用的软件。

软件工具范围很广,从最早的编辑程序,到今天形成的软件开发环境自动工具,其作用和规模大小虽不相同,但都属于软件工具之列。

一、软件工具的发展过程

在程序设计时代,从画出框图到编码,几乎没有任何工具,这时的程序修改相当困难。到了软件时代,出现了各种编译程序,使程序的编制大大前进了一步。但是,软件时代的软件工具仅以程序设计为重点,有很大的局限性。

进入软件工程时代,软件工具的开发利用正蓬勃展开,如联机编辑工具、文档化工具、代码自动生成工具以及测试工具等。近来软件工具的发展更为迅速,可用设计规格语言来书写软件设计说明书,并以自动生成方式来生产软件;以需求规格语言来书写用户需求规格说明书,并以这种需求规格说明书为输入,通过自动生成工具,从现有的软件资源中选取程序模块,自动结合,生成界面,自动生成用户所需的软件产品。

另一方面,共享软件资源的模块化技术、用户设计评价的模拟技术与会话型的设计技术组合在一起,正向着会话型模拟方向发展。

可以看出,软件工具对于软件开发来说是不可缺少的,是软件工程极为重要的一环。充足的软件工具可具体体现软件工程中先进的开发技术与方法。可以说,没有软件工具就没有高效率的软件生产。

二、软件工具的分类

1. 按开发阶段划分

根据软件生命周期的几个阶段,划分软件工具,可归为如下几类:

- (1)需求定义描述工具
- (2)设计、分析工具
- (3)编码工具
- (4)测试与调试工具
- (5)维护工具

2. 按工具的功能划分

可分为三大类：

(1)开发技术支撑工具

- 系统描述语言
- 环境模拟
- 调试、测试工具
- 性能评价工具
- 维护运行工具

(2)开发管理支撑工具

- 费用管理工具
- 质量管理工具
- 工程管理工具

(3)一般的软件工具

1.4.2 软件开发环境

从软件工程的观点看,软件的开发不仅仅是程序设计,而且是规格的描述与管理、阶段地逐步细化、各阶段正确性的确认、各模块的检查、文档的编写与管理以及编码等等。要提高软件开发的效率,需要有一个把文本编辑、文件系统、编译程序、检查评价工具等有机地组合在一起的软件开发环境,而且这个环境必须具有会话功能。

从80年代开始,会话型软件开发环境出现了一些新的发展趋向,一是分布式个人环境,二是智能环境。

为了使软件开发真正走向工程化,软件开发环境必须实现标准化,这已是大势所趋。

1.5 计算机软件的发展与未来

1.5.1 软件发展历史简述

自1946年世界上第一台电子计算机(ENIAC)问世以来,便对

人类生活产生了巨大的影响。经过短短四十多年,计算机已从第一代发展到了第五代。目前,计算机科学家正在研制第六代计算机。不过,这主要是从计算机硬件的发展来划分的。从计算机软件来看,也取得了巨大的进展。软件的发展大致经历了以下四个阶段。

一、汇编语言阶段

汇编语言阶段是软件发展的初期阶段。它是在机器语言的基础上发展起来的一种较低级的语言。此时的源程序通过汇编程序翻译成机器语言程序。虽说此时编程序仍很困难、复杂,但比起直接使用机器语言编程已是前进了一大步,编程效率得到很大提高。

二、高级语言阶段

高级语言是从50年代以后发展起来的。它不再像汇编语言那样依赖于具体的计算机硬件结构和不同的指令系统,而是规范化的代码。如FORTRAN、ALGOL60、PASCAL等。高级语言源程序只有经过“翻译”,转换成机器语言后机器才能执行。其负责“翻译”的程序称为编译程序。

由于编译程序的出现,计算机工作方式由单用户单作业发展到批作业处理方式。与此相适应,先后出现了为提高批作业效率的硬设备。高级语言的推广使用,极大地提高了编程序效率,同时使软件的易读性、可扩充性、易维护性等有了很大的提高。极大地促进了软件的发展。计算机外围设备的增加,软件的丰富,为操作系统的形成奠定了基础。

三、操作系统的形成

到60年代后期,计算机硬件已由电子管发展到小、中规模集成电路,计算机外围设备不断丰富。要使计算机的所有资源(包括软件和硬件)协调一致、有条不紊地高效率工作,必须有一个总调度来统一管理和协调。这个总调度、总管理就是操作系统。