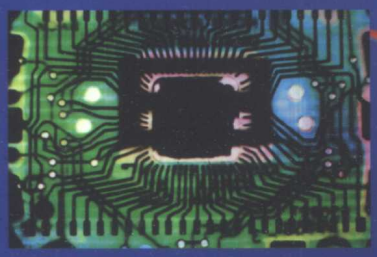
软件工程与 WINDOWS16BIT 人类

吴智涌　著



海天出版社

软件工程与 WINDOWS16BIT 人类

吴智涌　著

海 天 出 版 社

图书在版编目（CIP）数据

软件工程与 WINDOWS16BIT 人类/吴智涌. －深圳：海天出版社，2001. 3
ISBN 7－80654－401－1

Ⅰ.软... Ⅱ.吴... Ⅲ.①磁盘操作系统，DOS ②窗口软件，WINDOWS Ⅳ.TP31

中国版本图书馆 CIP 数据核字(2001)第 05148 号

海天出版社出版发行
（深圳市彩田南路海天大厦 518026）
http://www.htph.com E－mail:yws53@163.net
责任编辑:杨五三 封面设计:李 萌
责任技编:王 颖 责任校对:周 红 赵春燕

海天电子图书开发公司排版制作
深圳市彩帝印刷厂印刷 海天出版社经销
2001 年 3 月第 1 版 2001 年 3 月第 1 次印刷
开本:787mm×1092mm 1/16 印张:14.25
字数:270 千 印数:1－5000 册
定价:28.00 元

谨以此书 献给 WINDOWS16BIT 人类

WINDOWS16BIT 人类的工作模式：

1. 策划：富有创意地策划目标、策划后续活动中技术难点的解决方案。

2. 设计：设计 WINDOWS16BIT 人类高层软件系统。

3. 施工：WINDOWS16BIT 高层软件的执行。

WINDOWS16BIT 人类世界是一个规则第一的世界，在这个世界里，将会摆脱感情的束缚，感情必须服从于规则。在这个世界里，每个人的潜力都能得到最充分的发挥。

WINDOWS16BIT 人类将会流行一种知识接收器，可以随时随地通过无线电波接收所需要的任何知识，这些知识从知识服务器中发出，并且全部是按照 WINDOWS16BIT 人类知识全息机制进行组织的，因此，WINDOWS16BIT 人类将会轻松克服知识的局限，专心致力于策划、设计、施工。

欢迎广大同行提出宝贵意见，让我们一道将我们的共同事业推向前进，意见和问题可发往作者的邮箱：

wuzhiy@cmmail.com

前　言

本书的理论及技术能够成立的前提是：人类为一个分布式处理系统，并且是采用面向对象的技术手段进行设计的，在这套系统中的每个人都是一个对象，每个对象都可以看成是一个高级蛋白质机器人，各对象之间的同步依靠时钟和历法进行控制，整个社会通过各种通讯手段连成了一个网络系统。如今，在这套系统中存在着严重的软件危机，下一步还会出现更严重的软件危机，为了解决这些危机，必须创立人类软件工程学。作者目前将人类软件工程学分为两个层次，第一个层次为 WINDOWS16BIT 人类软件工程学，第二个层次为 WINDOWS32BIT 人类软件工程学。本书的目标是基本完成了第一个层次。

先谈谈对人类软件工程学进行上述分层命名的原因。目前在人类社会应用最广泛的软件系统为微软公司的操作系统，这套操作系统目前经过了 DOS，WINDOWS16BIT，WINDOWS32BIT 等三个发展阶段，根据全息原理，可以把微软公司的操作系统作为人类软件系统的原型系统，与这套原型系统相对应，把目前的人类定义为 DOS 人类，下一步将升级成 WINDOWS16BIT 人类，再下一步将升级成 WINDOWS32BIT 人类，并且，人类软件系统各版本的特征与微软的操作系统也可以建立对应关系，像 WINDOWS16BIT 操作系统相对于 DOS 操作系统来说只不过是一套外挂式图形系统一样，WINDOWS16BIT 人类相对于 DOS 人类也只需要改变高层软件系统，而不需要改变底层基因，也就是说，WINDOWS16BIT 人类是指高层软件系统进行了重新设计的人类。由于 WINDOWS32BIT 操作系统相对于 WINDOWS16BIT 操作系统来说，底层系统有了根本的变化，由 16 位变成了 32 位，上层软件系统也有了较大的变化，因此，未来的 WINDOWS32BIT 人类相对于 WINDOWS16BIT 人类，从底层基因到上层软件系统都会有较大的变化。

WINDOWS16BIT 人类软件工程研究的是 WINDOWS16BIT 人类高层软件系统的设计工程，实质上是一套仿真面向对象软件设计技术的管理技术，这套管理技术的优势是：可以使被管理的系统变成一个可控的、可预测的、可度量的、符合规则的、优化的系统，并且可以使该系统在发展的过程中一直保持着这种状态。WINDOWS16BIT 人类高层软件系统能使人类的行为科学化，其本质就是对集体未来的运行轨迹进行科学而又详细的预测和策划，并写成 WINDOWS16BIT 人类高层软件程序，使集体未来的发展在一种可控的状态下进行，自动排斥各种非科学因素的干扰。使集体严格按预定的步骤高效率地运行。从而使集体有更加强大的生命力。集体未来发展的详细蓝图就存贮在 WINDOWS16BIT 人类高层软件密码之中，而这种密码是由我们人类自身进行设计的，我们应该充分地利用这种设计权来实现我们的目标。

本书分成上下两篇，上篇用工程类知识全息算法将计算机软件工程学的基本理论按相同的模

式层层展开，作为 WINDOWS16BIT 人类软件工程的基础，其中，知识全息算法是本书在 16.3.2 中提出的一个新概念，属于 WINDOWS16BIT 人类世界必须引入的基本机制。下篇在提供了 WINDOWS16BIT 人类专有技术手段的基础上，以 WINDOWS16BIT 人类高层软件系统的开发为例对上篇中的每一个目标和步骤进行举例说明，下篇构成了 WINDOWS16BIT 人类软件工程的实质性内容，也基本完成了一套简单的 WINDOWS16BIT 人类高层软件系统的设计范例。未来的 WINDOWS16BIT 人类高层软件系统将会按照下篇提供的基本模式去设计。

对计算机软件工程学的知识按照工程类知识全息算法进行这样一番处理之后，能使软件开发人员在软件开发过程中最自然而又最充分地运用软件工程相关理论及技术手段。因为软件工程理论及相关的技术手段太多，在实际软件开发过程中，对这些理论和技术手段的运用往往容易挂一漏万，或运用不当，或经常处于犹豫状态，不知从何用起，甚至会认为，不运用软件工程相关理论及技术手段也照样会开发出优秀软件。而本书则用一系列简明的步骤将软件工程学的知识串联起来，在严格的步骤中很自然地把相关的技术手段引出来，读者在开发软件时，只需要分阶段地从前向后一条一条地检索并运用本书中的各步骤，就会使软件开发的全过程都能充分地在软件工程理论和技术手段的指导下进行，从而很自然地使软件开发质量有了较大的提高。

当然，本书不可能穷尽软件工程学中的所有技术手段，因为与软件工程相关的技术手段是在不断发展和变化的，但是软件工程的知识全息算法是可以不变的，也就是说，本书的结构是可以不变的，并且是可以按相同的模式无限扩充的，今后每发现一种新的技术手段都可以简单地添加到上篇相应章节的基本技术手段部分，然后再修改一下相应章节中的相应步骤，以便能通过该步骤把新加的技术手段引出来，本书下篇的相应步骤也要进行同步扩充，今后对于完全过时的技术手段，也可以从书中删除，并且把相应的步骤去掉，也可能仅仅对相关的技术手段和步骤进行更新。本书将尽量采用这种方式与软件工程的发展保持同步。

计算机软件工程学与 WINDOWS16BIT 人类软件工程学是相辅相成、相互完善的关系，一方面，WINDOWS16BIT 人类软件工程学是在计算机软件工程学的基础上扩充而成，另一方面，来自 WINDOWS16BIT 人类软件工程学中的研究成果，又可以使计算机软件工程学进入一个新的境界。例如，WINDOWS16BIT 人类高层软件的设计思想、设计方法、通用类将来会演化成计算机软件工程学的重要组成部分，这个部分从一定的抽象层次上看，相当于计算机软件工程学中的项目管理部分，从更高的层次上看，相当于所有常规软件的顶层模块，因此，要设计优质计算机软件必须先设计 WINDOWS16BIT 人类高层软件系统，还可以把计算机软件系统看成是 WINDOWS16BIT 人类高层软件系统的运行结果。另外，学习 WINDOWS16BIT 人类软件工程还有助于我们更好地理解计算机软件工程中的每一个实用开发步骤。

应用 WINDOWS16BIT 人类软件工程学的具体方法是：以本书下篇所设计的某公司 WINDOWS16BIT 人类高层软件系统作参考，去设计其他具体的 WINDOWS16BIT 人类高层软件系统，其基本思想是，把人看成是一种计算机，设计这种计算机的高层软件，让人在这种高层软件的控制下运行，不同的软件开发团队，其 WINDOWS16BIT 人类高层软件系统是有所不同的，因为任务不同，初始条件不同，对象资源不同，大小环境不同……，本书主张，一个软件开发团队在常规软件项目实施之前，必须先设计出 WINDOWS16BIT 人类高层软件系统，控制着团队内各成员的行为

模式，调度策略及工作步骤，并且在项目实施之前，就能对该项目用科学的手法进行度量，为制订最终的项目进度计划及人事考评提供依据。我们今后将要为人类的每一个集体都开发出这样一套软件系统，例如，可以开发出一套学校 WINDOWS16BIT 人类高层软件系统，工厂 WINDOWS16BIT 人类高层软件系统，商店 WINDOWS16BIT 人类高层软件系统，同软件开发团队类似，不同的学校，不同的工厂，不同的商店，其 WINDOWS16BIT 人类高层软件系统是各不相同的，但也有共同之处，这些共同之处可以作成一个通用的上层类，各具体的集体可以通过继承来获得这个上层类的功能和性能。从而可以通过软件的复用技术使 WINDOWS16BIT 人类高层软件系统的设计越来越简单。本书下篇在 15.2 中设计了一个通用管理类，人类中的任何一个集体都可以通过继承这个通用管理类来形成最适合于本集体的专用管理类。第二十章以此通用管理类为例详细介绍了 WINDOWS16BIT 人类高层软件系统的复用技术。

在设计 WINDOWS16BIT 人类高层软件系统时，要考虑该系统的长远应用、升级及维护问题。下篇在设计通用管理类时，详细设计了其中的每一个成员函数。对其中的每个成员函数都设计了三个应用层次，当应用到第一个层次时，相当于升级到了 WINDOWS1.X 人类，当应用到了第二个层次时，相当于升级到了 WINDOWS2.X 人类，当应用到了第三个层次时，相当于升级到了 WINDOWS3.X 人类。本书不仅仅对 WINDOWS16BIT 人类每一阶段的升级特性进行了规定，而且，还为 WINDOWS16BIT 人类向 WINDOWS32BIT 人类的升级作好了准备，这就是，通用管理类中的基因控制成员函数提供了由 WINDOWS16BIT 人类向 WINDOWS32BIT 人类升级的安全接口。我们在设计大型软件系统时，如果采用了这样的类似思路进行设计，就会使软件的生存期大为加长。

本书还强调测试工作的重要性，人类在设计软件时无论如何小心，都会出现这样或那样的问题，因此，本书提出了一个走读测试的概念，这种测试方法主要是针对详细设计的结果进行测试，也可以针对代码进行测试。在 16.3 中的每一个成员函数都给出了走读测试的范例，供读者参考。尽管有了走读测试，软件生命周期中的测试阶段仍然是必不可少的。

本书在介绍项目的背景知识时，不得不探讨了一下中国的远古哲学，这种哲学在中国属于一种深奥文化，古往今来流传下来的基本都是最终的结论和方法，而不涉及其实质性的原理，因此，本书则专门探讨其底层实质性的原理，以便揭开其神秘面纱，并普及于人类，本书作者曾经对这套哲学的原理进行了较长时间的研究，研究成果基本浓缩在本书之中。部分读者可能对这部分内容较难理解，但这属于较为底层的知识，并不影响对其他部分的理解。就像软件设计人员并不需要读懂 WINDOWS 的底层源代码就可以进行 WINDOWS 软件开发的道理是一样的。但是，这部分内容又不得不介绍，因为 WINDOWS16BIT 人类高层软件系统的运行在宏观上要受到一种客观存在的控制，在设计人类高层软件系统的工作中只能认识和运用这种控制，而不能改变这种控制，中国远古哲学是揭示这种控制的有力武器。当 WINDOWS16BIT 人类高层软件系统的设计达到上乘境界时，还是要启用这种武器的。

未来的 WINDOWS16BIT 人类在宏观上存在三级分工：初级为：施工员，中级为：设计师，高级为：策划师。以上级别的确定通过 WINDOWS16BIT 人类的良性竞争机制完成，同一级别之内的竞争也为良性竞争。级别具有动态性，一个人根据具体情况也可能兼任多个级别的工作。WINDOWS16BIT 人类世界的竞争是前进的动力，是相互尊敬的纽带。

目 录

下篇　WINDOWS16BIT 人类软件工程概论

上 篇

软件工程知识的全息化

知识全息化的核心是找出知识中最基本和最核心的东西，然后根据全息规则推知相关的一切，本书各章节基本遵循工程类知识的全息算法，全书总体上又由计算机软件工程学推出了 WINDOWS16BIT 人类软件工程学。

整个人类知识的全息化工程是一项无比壮丽的事业，本书目前所提供的只不过是一个尚未成熟的开端。

第1章 概 论

1.1 软件危机

软件危机是指在计算机软件开发和维护过程中所出现的一系列严重问题。一般来说，系统的规模越大，则潜在危机的可能性也越大。这些危机主要表现在如下几个方面：

1. 所开发出来的系统在功能、性能、操作方式上不符合用户的要求。

2. 软件开发到中后期才发现设计方案不合理，或在经济、技术、操作方面不可行，因而放弃开发或重新设计。

3. 软件难以维护和升级，软件的生存期过短。

4. 在管理和协调上失去控制，导致出现不可收拾的局面。

5. 某些软件产品中的隐患会引发重大的事故。

实质上，软件产品不管规模是大是小，其中都很容易存在一些或大或小的问题，从而使软件的质量得不到保证。通过测试也只能证明程序中的错误，而不能证明程序是正确的。

1.2 软件工程

软件危机一方面是由软件产品固有的特点决定的，另一方面与开发方法有关，因此，我们必须根据软件产品的固有特点，采用一套专门的方法和步骤去解决这一问题。

像机械、建筑等行业的工程活动一样，软件开发实质上也是一项工程活动，这种软件开发工程就叫软件工程。我们可以用“工程化”的思想作指导，把工程学的基本原理和方法引进到软件设计和生产中来，以便减少软件开发成本并提高软件质量，并且可以进一步在此基础上为软件开发建立一门学科，这门学科被称之为软件工程学。

软件工程学的最终目的是以较低的成本研制具有较高质量的软件产品。

1.3 软件工程知识的全息化

本书下篇在设计 WINDOWS16BIT 人类高层软件系统时，设计了一个通用管理类，在这个通用管理类中，有一个成员函数叫做知识全息化成员函数，或称之为知识全息机制，详见 16.3.2。

针对人类的知识爆炸局面，未来的人类有两件事情非抓不可，第一件事就是要构造出完善的全球知识检索系统，这套系统置于知识服务器之中，知识接收器可以通过无线电波联网接收知识

服务器中的知识，任何人只要有一台知识接收器便可以获得他想知道的一切知识，目前的 Internet 就是这套知识处理系统的原型。第二件事就是要引入知识全息机制对所有的知识来一次清理和全息化。其中，第二件事是一件更基础性的工作，知识服务器中的知识是按知识全息机制来组织的。

应用知识全息机制的方法是：先构造出相应的知识全息算法，然后便可以简明地按照算法步骤，一步一步地把相关的知识串联出来。本书在 16.3.2 中构造了一个技术类知识的全息算法，但这一算法不适用于软件工程知识的全息化。因此，本书专门为软件工程知识构造了一套知识全息算法，这套算法适用于所有工程类知识，也很简单，仅三句话，其内容如下：

A. 找出目标

B. 找出实现目标的步骤

C. 找出实现目标的基本技术手段

其入口点在目标，由目标引出步骤，由步骤引出基本技术手段，应用思路相当明确。本书上下两篇都是按上述算法进行表达的，但是下篇把基本技术手段完全融入到步骤中去了。而上述全息算法为把软件开发过程的改善与技术提升的相结合提供了具体途径，过程的改善主要通过步骤的完善来进行，本书尽量通过完善的步骤把所有相关的技术手段串联起来，从而使基本技术手段的提高可以马上融入到过程之中，提高开发质量。

以上算法的运用是分层次的，每一层次都运用相同的模式进行描述，对于整个软件工程学层次来说，运用结果如下：

1. 目标：用最小的代价获得最有生命力的软件产品。

2. 步骤：分成如下七个步骤实现上述目标：

【1】可行性研究，判断该项目是否值得进行下去。

【2】需求分析，收集并分析出完整的用户需求。

【3】概要设计，设计工作进行到模块级。

【4】详细设计，设计工作进行到算法流程级。

【5】编码，选一种恰当的工具，编写出代码。

【6】测试，找出代码中可能存在的问题。

【7】维护，交付给用户后的进一步完善工作。

3. 基本技术手段：采用一套严密的管理技术对以上七个步骤进行控制，每一阶段完成之后都要写出相应的文档，并进行评审，然后决定是否进行到下一步。在具体的软件项目中，要根据具体情况选择一种最恰当的开发模型，用项目管理技术将上述七大步骤连成一体。

下面介绍五种具体的开发模型，除了螺旋模型外，在其他模型中，本书都在传统模型的基础上增加了一个项目管理方框，作为模型的核心，这样便把所有模型统一成了一个旋转体，所有各项活动都在核心的控制下旋转，这种统一的旋转模型可以作为整个软件工程活动的全息模型。可行性研究阶段相当于处在混沌状态，混沌一开，模型便开始旋转，与整个宇宙的全息模型具有一致性。

一、瀑布旋转模型

瀑布旋转模型如图 1－1 所示：

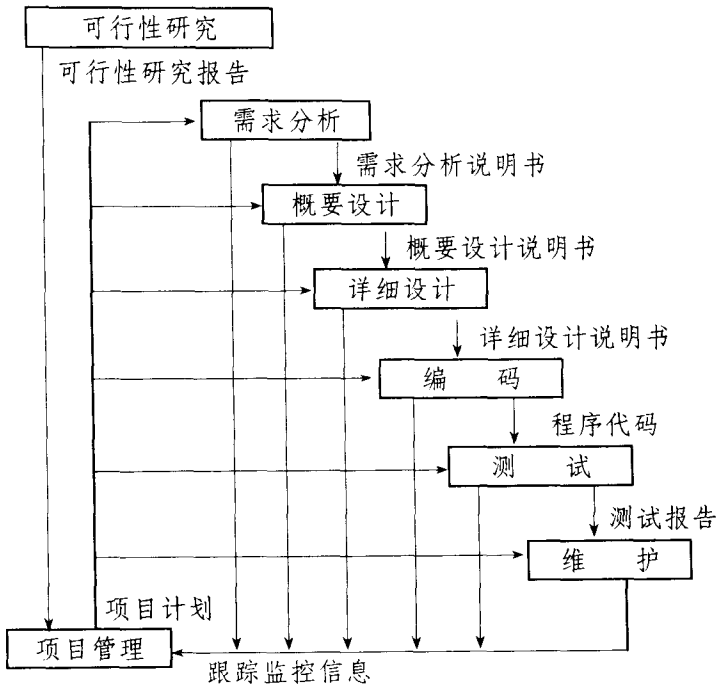


图 1-1 瀑布旋转模型

以上瀑布旋转模型是由传统的瀑布模型改造而成，其特点是以项目管理为核心，随时监控软件开发的每一个阶段，通过反馈机制，保证整个软件开发周期的各个阶段不断地处于优化和可控状态。例如，当设计工作进行到详细设计阶段时，项目管理同时在同步地监控详细设计的信息。发现问题后，如果该问题属详细设计的问题，则要及时对详细设计工作的相关方面及时进行调整。如果属于前一阶段的遗留问题，则要抓住源头，经过一定的审批之后，在局部将项目返到前一阶段，如概要设计，需求分析阶段等，并且要再次任用相应阶段的步骤去工作，直到完成之后再进入下一步。

二、原型旋转开发模型

原型旋转开发模型如图 1-2 所示：

以上原型旋转开发模型是在原型开发模型的基础上修改而成，其特点仍然是以项目管理为核心，随时监控软件开发的每一个阶段，通过一种反馈机制，保证整个软件开发周期的各个阶段不断地处于优化状态。

原型旋转开发模型在指导思想上就没有准备一次成功，采用了一种逐步求精的策略。但是，瀑布旋转开发模型仍然是最基本的，原型旋转模型是在瀑布旋转开发模型的基础上稍加变化形成，二者的本质是一致的，但在设计的指导思想上有差别，前者在指导思想上企图一次成功，返工属于一种不正常的现象，而后者的指导思想是让系统经过逐步扩充而形成，向前一阶段返工属于一种正常行为。