

计算机信息管理专业 独立本科段

全国高等教育自学考试

# 软件开发工具习题详解

黄明 梁旭 编著

 机械工业出版社  
CHINA MACHINE PRESS



# 软件开发工具与数据库

第 1 章 数据库系统概论



第 2 章 数据库设计

全国高等教育自学考试

# 软件开发工具 习题详解

(计算机信息管理专业 独立本科段)

黄 明 梁 旭 编著



机械工业出版社

本书是根据“全国自学考试（计算机信息管理专业 独立本科段）考试大纲”以及历年考题编写的。全书共分4部分：第1部分是笔试应试指南；第2部分是笔试题解；第3部分是模拟试卷及参考答案；最后是附录，包括考试大纲和2002年下半年试卷。

本书紧扣考试大纲，内容取舍得当，叙述通俗易懂，附有大量与考试题型类似的习题及答案，以检查读者对考点的掌握程度。

本书适用于准备参加全国自学考试（计算机信息管理专业 独立本科段）的考生，也可作为大专院校和培训班的教学参考书。

## 图书在版编目（CIP）数据

软件开发工具习题详解/黄明，梁旭编著. —北京：机械工业出版社，2004.3  
（全国高等教育自学考试）

ISBN 7-111-13694-2

I. 软... II. ①黄...②梁... III. 软件工具—高等教育—自学考试—解  
题 IV. TP311.56-44

中国版本图书馆CIP数据核字（2003）第120153号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

策 划：胡毓坚

责任编辑：孙 业

责任印制：施 红

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2004年3月第1版·第1次印刷

787mm×1092mm 1/16 ·12印张·289千字

0 001—5 000册

定价：19.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68993821、88379646

封面无防伪标均为盗版

# 出版说明

全国高等教育自学考试指导委员会推出面向社会的高等自学考试，经过 10 多年的实践，已建立起一整套较为完善的规章制度和操作规程，考试组织严密规范，考试纪律严格；坚持考试标准，实行教考分离，确保了毕业生的质量。它为没有机会进入高等学校的中国公民提供了接受高等教育的机会，并以严格的国家考试保证了毕业生的质量，获得了普遍赞誉。国家自考中心于 2002 年开始执行新的考试计划。新计划中开设的专业共 224 个，其中专科 141 个占 63%，独立本科段 61 个占 27%，专本衔接专业 22 个占 10%。为帮助、指导广大自考生深入理解计算机及相关专业考试的基本概念，灵活运用基本知识，掌握解题方法和技巧，熟悉考试模式，进一步提高应试能力和计算机水平，特编写了以下专业的基础课与专业课主要课程的习题详解。

- ◇ 计算机及应用专业 独立本科段
- ◇ 计算机信息管理专业 独立本科段
- ◇ 计算机网络专业 独立本科段
- ◇ 计算机及应用专业 专科

## 丛书特点：

### 1. 以 2002 年最新考试大纲为基准

本丛书是根据 2002 年最新考试大纲，为参加全国高等教育自学考试考生编写的一套习题详解教材。

### 2. 例题反映了历届考试中的难度和水平

书中对大量的例题进行了分析，所选例题都是在对最近几年考题深入研究的基础上精心筛选的，从深度和广度上反映了历届考试中的难度和水平。

### 3. 作者经验丰富

本丛书的作者都是多年从事全国高等教育自学考试辅导的高等院校的教师。

## 读者对象：

- ◇ 准备参加全国高等教育自学考试的考生。
- ◇ 计算机及相关专业的本专科生。

## 前言

自学考试是对自学者进行以学历考试为主的国家高等教育学历考试。本书是为帮助和指导广大考生深入理解自学考试的基本概念，灵活运用基本知识，掌握解题方法和技巧，熟悉考试模式，进一步提高应试能力和计算机水平而编写的。

全书共分 4 部分，即笔试应试指南、笔试题解、模拟试卷及参考答案和附录。书中所选例题均是在对历年真题深入研究的基础上精心筛选的，从深度和广度上反映了考试中的难度和水平。模拟试卷的题型分配与真题一致，这些题目是考试指导教师的多年积累，在辅导班中多次实际使用过。

书中附录给出了“全国自学考试（计算机信息管理专业 独立本科段）软件开发工具考试大纲”，以及“2002 年下半年全国自学考试软件开发工具试卷及参考答案”。

本书由黄明、梁旭编写。

由于编者水平有限，编写时间仓促，书中错误和不妥之处在所难免，请读者和专家批评指正。

读者在使用本书的过程中如有问题，可通过 E-mail 与我们联系：

dlhm@263.net

编者

# 目 录

出版说明  
前言

## 第 1 部分 笔试应试指南

1.1 笔试应试策略 .....	2
1.2 笔试考点归纳 .....	3
1.2.1 绪论 .....	3
1.2.2 软件开发过程及其组织 .....	6
1.2.3 软件开发工具的理论基础 .....	11
1.2.4 软件开发工具的技术要素 .....	16
1.2.5 软件开发工具的使用与开发 .....	21
1.2.6 软件开发工具的现状与发展 .....	24
1.2.7 PowerBuilder 应用开发工具 .....	26
1.2.8 实用的 CASE 工具——NEWCAFG .....	31
1.2.9 多媒体创作工具——TOOLBOOK .....	32
1.2.10 网页制作工具——FrontPage .....	33
1.2.11 UML 与软件工具 Rose .....	35

## 第 2 部分 笔试题解

2.1 绪论 .....	40
2.1.1 单项选择题 .....	40
2.1.2 多项选择题 .....	42
2.1.3 填空题 .....	45
2.1.4 名词解释 .....	46
2.1.5 简答题 .....	47
2.1.6 论述题 .....	49
2.1.7 习题 .....	50
2.2 软件开发过程及其组织 .....	52
2.2.1 单项选择题 .....	52
2.2.2 多项选择题 .....	56
2.2.3 填空题 .....	59
2.2.4 名词解释 .....	60
2.2.5 简答题 .....	60
2.2.6 论述题 .....	62

2.2.7	习题	63
2.3	软件开发工具的理论基础	65
2.3.1	单项选择题	65
2.3.2	多项选择题	69
2.3.3	填空题	72
2.3.4	名词解释	72
2.3.5	简答题	73
2.3.6	论述题	74
2.3.7	习题	76
2.4	软件开发工具的技术要素	79
2.4.1	单项选择题	79
2.4.2	多项选择题	83
2.4.3	填空题	85
2.4.4	名词解释	86
2.4.5	简答题	86
2.4.6	论述题	88
2.4.7	习题	90
2.5	软件开发工具的使用与开发	91
2.5.1	单项选择题	91
2.5.2	多项选择题	94
2.5.3	填空题	95
2.5.4	简答题	95
2.5.5	论述题	96
2.5.6	习题	98
2.6	软件开发工具的现状与发展	100
2.6.1	单项选择题	100
2.6.2	多项选择题	101
2.6.3	填空题	102
2.6.4	名词解释	102
2.6.5	简答题	103
2.6.6	论述题	103
2.6.7	习题	105
2.7	PowerBuilder 应用开发工具	107
2.7.1	单项选择题	107
2.7.2	多项选择题	110
2.7.3	填空题	110
2.7.4	名词解释	111
2.7.5	简答题	112
2.7.6	论述题	115



2.7.7	应用题	115
2.7.8	习题	119
2.8	实用的 CASE 工具——NEWCAFG	122
2.8.1	单项选择题	122
2.8.2	多选题	122
2.8.3	填空题	123
2.8.4	简答题	123
2.8.5	习题	123
2.9	多媒体创作工具——TOOLBOOK	124
2.9.1	单项选择题	124
2.9.2	多项选择题	125
2.9.3	填空题	125
2.9.4	论述题	126
2.9.5	习题	126
2.10	网页制作工具——FrontPage	127
2.10.1	单项选择题	127
2.10.2	多项选择题	128
2.10.3	论述题	128
2.10.4	习题	129
2.11	UML 与软件工具 Rose	130
2.11.1	单项选择题	130
2.11.2	多项选择题	131
2.11.3	习题	131
2.12	习题参考答案	132

### 第 3 部分 模拟试卷及参考答案

3.1	模拟试卷一及参考答案	150
3.1.1	模拟试卷一	150
3.1.2	参考答案	153
3.2	模拟试卷二及参考答案	155
3.2.1	模拟试卷二	155
3.2.2	参考答案	160
附录		163
附录 A	全国自学考试（计算机信息管理专业 独立本科段）软件开发工具考试大纲	164
附录 B	2002 年下半年全国高等教育自学考试软件开发工具试卷及参考答案	175
参考文献		182

# 1

## 第1部分

---

# 笔试应试指南

笔试应试策略

笔试考点归纳

## 1.1 笔试应试策略

全国自学考试（计算机信息管理专业 独立本科段）软件开发工具考试大纲涵盖了绪论、软件开发过程及其组织、软件开发工具的理论基础、软件开发工具的技术要素、软件开发工具的使用与开发、软件开发工具的现状与发展、PowerBuilder 应用开发工具、自制的 CASE 工具——NEWCAFG、多媒体创作工具——TOOLBOOK、网页制作工具——FrontPage、UML 与软件工具 Rose 11 部分内容。使用的教材是由全国高等教育自学考试指导委员会组编，陈禹、方美琪编著的《软件开发工具》，2000 年 3 月由经济科学出版社出版。考试复习的过程中要紧紧围绕大纲的知识点，首先对大纲涉及的各部分基本概念熟练掌握。

第 1 章为绪论，属于基础知识，要求熟练掌握一些概念，特别需要掌握的是软件开发工具的发展过程，以及对软件开发工具概念的定义。熟悉软件开发工具的分类和具体如何分类。复习时要注意灵活运用，从而加深理解。占分量约为 10 分。

第 2 章为软件开发过程及其组织，要求理解软件开发中的困难，特别针对大型软件开发过程中产生的困难及其原因。占分量约为 10 分。

第 3 章为软件开发工具的理论基础，是重点考察的内容，包括概念模式和信息库，需要真正做到理解概念模式的作用和意义；熟练运用八种概念模型，具体为框图、结构图、数据流程图、实体关系图、数据字典、时序网络、数学与逻辑模型、计算机模拟模型；掌握人机界面设计的原则。占分量约为 20 分。

第 4 章为软件开发工具的技术要素，要求掌握软件开发工具的功能和一般的结构；掌握总控部分和人机界面的重要地位，理解信息库的组织方式和代码与文档的生成方式是什么。占分量约为 10 分。

第 5 章为软件开发工具的使用与开发，重点和难点是购置与开发软件开发工具的权衡。要求理解在什么情况下应该购置软件开发工具，而购置后的管理工作是什么，并理解购置的具体步骤；了解自行开发软件开发工具时应该注意什么。占分量约为 10 分。

第 6 章为软件开发工具的现状与发展，要求了解软件开发工具在国内、国外的现状，以及未来的发展趋势。占分量在 10 分。

第 7 章为 PowerBuilder 应用开发工具，主要掌握 PowerBuilder 具体的开发过程。这一部分考试主要出应用题，占分量约为 20 分。

第 8 章为实用的 CASE 工具——NEWCAFG，主要了解该工具的特点。占分量约为 2 分。

第 9 章为多媒体创作工具——TOOLBOOK，主要了解该工具的特点。占分量约为 2 分。

第 10 章为网页制作工具——FrontPage，掌握制作网页的基本方法。占分量约为 2 分。

第 11 章为 UML 与软件工具 Rose，主要了解该工具的特点。占分量约为 4 分。

在复习时根据大纲里提供的考核点和考核要求来进行复习，这样就能抓住重点，进行有效复习。在做练习时，要根据考试的题型进行练习，并在掌握基本概念的基础上，掌握一些解题技巧。软件开发工具的考试题型有：单选题、多选题、填空题、名词解释、简答题、论述题和应用题等。对于不同题型，要采用不同的答题方法。

单选题：这种题型可考查考生的理解、推理分析，综合比较，评分客观。在答题时，可以直接得出正确答案。对于没有把握的试题，可以采用排除法，经过分析比较，逐步排除错

误答案，最终选定正确答案。

**多选题：**这种题型可考查考生对问题的多种理解。在答案中，应有两个以上选项是正确的。这种题容易使考生产生模糊，在考试时应认真读懂题，答题时可以采用排除法，经过分析比较逐步排除错误答案，最终选定正确答案。

**填空题：**这种题型常用于考核考生观察能力与运用有关概念、原理的能力。在答题时，无论有几个空，回答都应明确、肯定。考生在复习时，对学科知识中最基本的知识、概念、原理等要牢记。

**名词解释题：**这种题型着重考核考生对基本概念和原理的准确理解。

**简答题：**这种题型灵活性比较大，着重考核考生对概念、知识、原理的掌握。答案往往是惟一的，是原理性的东西，但有时跨度会很大，需要总结归纳对比。这就要求我们在复习的过程中，仔细理解大纲中涉及到的知识点，因为任何一个知识点都可以出简答题。

**论述题：**这种题型着重考核考生分析、解决实际问题的能力，考核考生综合应用能力和创见性。在答题时，要综合运用所学知识进行分析。

**应用题：**这种题型着重考核考生的综合应用能力，要在理解基本原理的基础上，综合运用所学知识进行分析解答。

考生在复习时在掌握知识点的同时还应抓住这些题型的特点，这样才能达到好的应试效果。

## 1.2 笔试考点归纳

### 1.2.1 绪论

#### 1. 软件开发工具的由来

##### (1) 软件开发工具发展过程。

当现代的电子计算机刚诞生时，人们面对的是只能执行机器指令的硬件设备，即所谓的“裸机”状态。这时，软件的概念还只处于萌芽状态人们的注意力还集中于如何设计出各种各样的物理器件，实现最基本的操作。

汇编语言，即第二代语言的出现，它建立一些专用的“工具”，使某些可以由机器来完成的信息处理工作交给计算机去做，而使得人们的知识、经验转化为计算机的操作。

20世纪60年代初期，FORTRAN, ALGOL 和 COBOL 等高级程序设计语言的成熟与普及，标志着计算机真正走出了难以应用的困窘局面。与汇编语言不同的是，第三代语言突破了与机器指令一一对应的限制用尽可能接近自然语言的表达方式描述人们设想的处理过程，而把这种表达方式向机器指令的转化工作，交给专门的“工具”——编译系统去完成。另一个重要的变化是高级程序语言实现了对机器的独立性，即它不依赖特定的硬件系统，抽象地逻辑地描述处理和算法，而把硬件系统之间的区别交给不同的编译系统去处理，从而大大提高了程序的可移植性。

20世纪70年代末，80年代初，技术人员已经想到了用软件来支持软件开发工作，这个阶段可以称为利用通用软件作为辅助工具的阶段，或者反过来说：没有专用的软件开发工具的阶段。

20 世纪 80 年代以来，一些专门用于支持软件开发的软件开发工具陆续问世，从而进入了专用的软件开发工具的阶段。之后，为了解决一致性的保持问题，导致了集成的软件开发工具的产生。

### (2) 大型软件开发的困难及原因。

大型软件开发中的困难主要在一致性的保持、测试、工作进度难以控制、文档与代码的协调和版本更新等方面。在这些方面出现困难是由于以下原因：

- 1) 大系统的复杂性。
- 2) 许多有主动性的个人之间的组织与协调。
- 3) 各个应用领域差别。
- 4) 时间因素变化因素。

### (3) 软件开发工具产生及应用的必然性及发展趋势。

在 20 世纪 70 年代末，80 年代初，苦于软件开发中遇到各种困难的许多技术人员，已经想出了用软件来支持软件开发工作。用软件帮助软件开发人员编写文档或画图可以减少不少工作量。20 世纪 80 年代以来，一些专门用于支持软件开发的软件开发工具陆续问世，从而进入了专用的软件开发工具的阶段。为了解决一致性的保持问题，导致了集成的软件开发工具的产生。

回顾软件开发工具从无到有，从分散到集成的发展过程，可以清楚地看到软件开发工具的出现绝不是偶然的，它是软件发展的必然趋势，是软件技术发展的一个方向。

## 2. 软件开发工具的概念

### (1) 理解软件与软件开发工作的实质。

软件这个新词，是特指一种看不见、摸不着的，但又发挥着十分重要的作用的、事先编辑好的指令系列。随着计算机软件技术的发展，软件开发工作的概念从一次性的、具体的编程，扩大到了长期的、不断更新的过程。

### (2) 软件开发工具的概念、地位及作用。

在高级程序设计语言（第三代语言）的基础上，为提高软件开发的质量和效率，从规划、分析、设计、测试、成文和管理各方面，对软件开发提供不同程度帮助的一类新型软件。其要点是：

- 1) 它是在高级程序设计语言（第三代语言）之后，软件技术进一步发展的产物。
- 2) 它的目的是在人们开发软件中给予各种不同方面、不同程度的支持或帮助。
- 3) 它支持软件开发的全过程，而不是仅限于编码或其他稳定的工作阶段。

## 3. 软件开发工具的功能与性能

### (1) 软件开发的过程。

关于软件开发的过程有许多不同的划分方法。尽管各种说法有各种不同的用语、有或多或少的不同的阶段的划分，但是应当说是大同小异的。

1) 软件开发工作的第一阶段是初始要求的提出。软件开发工作者的任务是根据这种初始要求形成严格、明确的、可供实际开发使用的功能说明书。由于一般的用户对于计算机的功能并不熟悉，在多数情况下，用户最初提出的要求是不能直接用来作为编制软件的依据的。

2) 第二阶段是总体设计。它的任务是根据软件功能说明书的要求，完成软件的总体设计。这包括整个软件的结构设计、公用的数据文件或数据库的设计、各部分的连接方式及信息交换的标准等主要方面的内容。

3) 第三阶段是程序的编写与文档的编写,可以统称为实现阶段。程序的编写无疑是占用人力、时间较多的任务。在这里需要加以强调的是组织与协调的重要性。

4) 第四阶段是测试或调试阶段。其中包括两个部分,模块的调试与整个软件的联调。模块的测试是根据总体设计所制定的各个模块的设计任务书,对程序员的模块进行验收,看它们是否实现了所要求的功能,是否达到了所要求的性能指标。由于设计不可能是百分之百的完美的,即使每个模块都达到了设计任务书的要求,整个系统能否达到预期的目标还需要进行测试,另外,完成的软件与编写的文档是否一致也必须认真检查。

#### (2) 软件开发工具的五项基本功能指标。

从软件开发过程可以看到,其中有许多工作需要用软件开发工具的支持或帮助。归纳以上的各个工作阶段,可以把软件开发工具应提供的各类支持工作归纳成以下五个主要方面。

1) 认识与描述客观系统。这主要使用在软件开发工作的第一个阶段——需求分析阶段。由于需求分析在软件开发中的地位越来越重要,人们迫切需要在明确需求、形成软件功能说明书方面得到工具的支持。

2) 存储及管理开发过程中的信息。在软件开发的各阶段都要产生及使用许多信息。而这些信息到了测试阶段还要用来对已经编好的软件进行评价。

3) 代码的编写或生成。在整个软件开发过程中,程序编写工作占了相当比例的人力、物力、时间,提高代码的编制速度与效率显然是改进软件工作的一个重要方面。

4) 文档的编制或生成。文档编写是软件开发中一项十分繁重的工作,不但费时费力,而且很难保持一致。

5) 软件项目的管理。这一功能是为项目管理人员提供的。

#### (3) 软件开发工具的主要性能指标。

对于软件开发工具来说,性能的说明则要求说明这些支持或帮助的程度如何。当然,作为一般的软件,效率、响应速度等都是必须考虑的。对于软件开发工具,以下五项应当是特别重要的。

1) 表达能力或描述能力。因为软件项目的情况千变万化,软件开发工具要能够适用于某种类型的软件项目,就要能适应种种不同的情况,否则就不可能对软件开发提供有效的、实际的帮助。

2) 保持信息一致性的能力。软件开发工具要为软件开发管理者在开发过程中涉及的大量信息。这项工作中一致性的检验与控制是十分关键的。

3) 使用的方便程度。既然是工具,当然就应当尽量方便用户,而不能使用户因为用工具而添加麻烦。

4) 工具的可靠程度。软件开发工具应当具有足够的可靠性,即在各种各样干扰下仍能保持正常工作,而不能丢失或弄错信息。

5) 对硬件和软件环境的要求。如果软件开发工具对硬件、软件的环境要求太高,也会影响它的使用范围。一般来说,软件开发工具对环境的要求不应当超出它所支持的应用软件的环境要求,有时甚至还应当低于应用软件的环境要求。

### 4. 软件开发工具的分类

(1) 支持不同工作阶段的三种工具。

软件工作是一个长期的、多阶段的过程，各个阶段对信息和信息处理的需求不同，相应的工具也就不相同。粗略地说，可以把软件开发工具分为三类：设计工具、分析工具和计划工具。

1) 从工作的抽象程度来看，设计工具是最具体的，它是指在实现阶段对人们提供帮助的工具。

2) 分析工具主要是指用于支持需求分析的工具。

3) 计划工具则从更宏观的角度去看待软件开发。它不仅从项目管理的角度帮助人们组织与实施项目，并且把有关进度、资源、质量、验收情况等信息有条不紊地管理起来，而且考虑到了项目的反复循环。

(2) 单项工具与集成工具的区别。

单项工具是指专用的、面对某一工作阶段或某一工作任务的软件开发工具。集成化工具是指集成化的、面对软件开发全过程的软件开发工具。真正集成化的软件开发工具，要求人们对于软件开发过程这样的复杂事物有更深入的认识和了解。

(3) 依赖于机器或软件的工具与独立于机器或软件的工具的区别。

软件开发工具又可以按它与硬件、软件的关系分类。有的软件开发工具依赖于特定的计算机或特定的软件（如某种数据库管理系统）。另一类软件开发工具则是独立于硬件与其他软件的，这当然与工具自身的情况有关。一般来说，设计工具多是依赖于特定软件的，因为它生成的代码或测试数据不是抽象的，而是具体的某一种语言的代码或该语言所要求的格式的数据。软件开发工具是否依赖于特定的计算机硬件或软件系统，对于应用的效果与作用是有直接影响的。这个问题是研究和使用它所必须注意的。

## 5. 软件开发工具的研究与应用

(1) 本课程学习的目的、范围及方法。

学习软件开发工具的目的在于了解软件开发工具的概念、理论基础、基本功能、发展现状与前景，以便能够在实际工作中正确地选择与使用软件开发工具，在必要时能够参加或组织软件开发工具的自行研制，从而达到提高软件工作水平与效率的目标。本课程学习的范围包括软件开发过程及其组织、软件开发工具的理论基础、软件开发工具的技术要素、软件开发工具的使用与开发、软件开发工具的现状与发展以及一些实用的开发工具。学习本课程时要采用先理论，后实践，理论与实践相结合的方法，循序渐进。

(2) 软件开发工具的研究与应用的状况。

作为技术的一个发展方向，或者作为一种软件产品，软件开发工具已经得到各方面的广泛重视。对于软件开发工具的应用、开发与研究已经形成了相当规模。对于软件开发的兴趣来自不同方面。由于有需求，一些厂家与研究单位选中了这个方向，进行了软件开发工具的研制工作。因此，作为一个研究领域，软件开发工具已经引起了相当广泛的关注。

## 1.2.2 软件开发过程及其组织

### 1. 软件开发的困难

(1) 软件开发中的两个转换过程。

第一个转换是从用户的理解到程序员的转换（不同行业人员区别），第二个转换是从程序员的理解到程序的实现。要做好第一个转换，程序员要做到对应用领域尽可能熟悉、

正确理解用户的需求等。做好第二个转换的关键在于程序员要有较高的知识水平和实际经验。

#### (2) 大型软件开发中的困难及其产生原因。

真正的大型软件的研制必须采用现代化大生产的方式才能完成，真正严重的困难恰恰在于大型软件的开发。当程序的规模超出一个人的力量范围时，一系列新的、更难以克服的困难就出现在人们的面前了。

1) 一致性的保持成为十分困难的问题。作为一个统一的软件，各部分之间不可避免地要有各种信息的交流与共享，如参数的传递，公用文件或数据库的格式，各种数据的口径、单位或精度，以至对于一些名词、术语的理解，这些因素直接影响到各部分之间的协调与配合，这决定了它们能否有机地组成一个完整的软件，实现预期的功能。

2) 测试的困难大大增加。一个人写程序，尚且往往忽略修改带来影响，出现所谓的“水波效应”。这就是说，当修改程序的某一处时，由于没有充分考虑这一部分的修改对相关部分带来的影响，结果出现了新的错误。

3) 工作进度难以控制。对于大型软件来说，这个问题更为严重，如果说每个程序员自己那部分的检测尚难以按时完成的话，那么整个系统的调试就更难控制了，因为除了每个程序员对自己所承担部分进行单独的测试之外，更重要的是对整个系统进行联调。许多要求是否真正达到，只有到最后联调时才能看出来。

4) 文档代码的协调十分困难。所说的文档包括功能说明书、总体设计、各模块的计划任务书。每个部分的研制报告、测试报告，直到最终的验收报告和给使用者的用户手册。程序的调试是不断反复进行的，在某些时候，文档是编写程序的依据；在某些时候，又需要根据编程的情况撰写文档。显然，这两者必须一致，否则文档不仅没有用处，反而会造成混乱。

5) 版本更新带来的困难。版本更新的大量工作往往在于对内容重新进行审核，包括对代码和文档两方面的认真检查。因此，这种情况下，很容易出现新版本中各部分代码不一致和代码与文档不一致的情况。

上述这些困难的产生并不能简单的归结为程序员或软件工作者的弱点，其原因是复杂的，多方面的。

首先，这些困难来自大系统的复杂性。一个程序如果只有 50 行，那么什么问题也是好解决的，但是如果一个程序有 50 万行，那么最简单的问题，（如搜索、标识符选用等）都会成为相当困难的问题。由于量的变化造成了质的差别，软件是一个极好的例子。一般来说，人类对于大量元素组成的复杂系统，至今尚没有理想的描述与控制方法。这也是大型软件开发中产生困难的根本原因。

其次，许多具有主动性的个人之间的组织与协调，这本身也带来大量的困难。从管理学的观点来看，当组织许多人共同完成一项大的任务时，分工和协调就成为关键因素。分工是否合理，信息是否能及时地沟通，步调是否能一致，这都不是自然形成的，都需要精心的安排与考虑。

另外，各个应用领域之间的差别也导致这些困难的加重。软件作为应用领域与硬件之间的桥梁，一头是固定的，另一头却是变化无穷的。世间各行各业尽管都要用到信息，都力图与现代信息技术相结合，但是它们之间的差别却是如此之大，给软件开发工作者增添了许多



困难。

最后，时间的因素、变化的因素也给软件开发工作带来许多困难，现实的社会经济生活每日每时都在变化，作为与社会生活紧密联系的应用软件当然也处在不断的变化之中，用户需求的变化，社会环境的变化，硬件及相关技术的变化，无处不给软件提出新的要求。这种易变的性质，使软件工作的困难又增添了不少。

总之，软件工作的困难来源于多方面，而且是人类当代面临的总的困难的一部分，它是与软件的本质联系在一起的，不可能在短期内轻易地得到解决。

## 2. 软件开发方法的发展

### (1) 结构化程序设计的思想方法。

结构化程序设计思想产生的背景：

早在 20 世纪 60 年代末期就产生了结构化程序设计的思想，代表人物有 E. Dijkstra, E. Yourdon 等人，Dijkstra 在他的一篇论文中指出，软件编写中之所以出现种种问题，是因为它本身结构不好。E. Yourdon 后来把程序的结构分解成三种基本模块：处理单元，循环机制，二分决策机制。按照这种方法来构造程序，就可以把程序内容的各种相互影响有效地控制在模块内部，从而避免“水波效应”。这样一来，问题的关键就在于合理地划分模块，并保证模块符合“结构良好”的要求，因而提出了结构化思想。

结构化程序设计中，对模块划分的基本要求：

- 1) 模块的功能在逻辑上尽可能地单一化、明确化、最好做到一一对应。
- 2) 模块之间的联系及互相影响尽可能地少，对于必须联系的都应当加以明确的说明。
- 3) 模块的规模应当足够小，以便使它本身的调试易于进行。

结构化程序设计中，实施模块划分的方法是：不管软件多大，都采取自上而下、逐步分解的方法，分成若干部分，并把它们之间的关系明确地表达出来，直到最底层的模块达到所要求的规模为止。

结构化程序设计的要点：

- 1) 限制甚至不用 GOTO 语句，绝对禁止超越模块边界的 GOTO 语句。
- 2) 子程序尽可能做到只有一个入口、一个出口。
- 3) 程序风格应尽量明确、清晰，包括适当增加注释，书写格式体现层次结构（如循环体内的语句一致地右移两格），变量名称的选用尽量具有逻辑意义等。
- 4) 在程序编写的同时完成有关的文档编撰，不要拖延时间。

结构化程序设计的缺点：是它增大了程序工作量，增加了编程中的麻烦与琐碎的工作，并且降低了程序的运行效率，但是，使用这种方法，可以保证软件的可靠、准确，而且使软件易于修改，故这种代价是值得的。

### (2) 软件工程的基本思想与方法。

软件工程产生的背景：工程常常是指复杂的、需要认真组织的、有许多人参加的事情。工程的观点是和近现代的大生产联系在一起的，它是和依靠个人技艺的手工生产方式相对立的。人类的生产从靠工匠手艺到靠工程化生产是一个重要的进步。这个思想正是软件工程思想的由来。由于软件危机的发生，人们认识到，把软件的质量寄托于各个程序员的技术与认真负责是不牢靠的、危险的，也是不现实的。要想大幅度地从根本上提高软件开发的效率和质量。就应当吸取人们在各种传统产业中的成功经验，从组织和管理的角度加强