

全国计算机技术与软件专业技术资格（水平）考试参考用书

软件设计师考试科目2：

# 软件设计

## ——考点解析与模拟训练

全国计算机技术与软件专业技术资格（水平）考试办公室推荐

刘华 主编 屈红刚 鲁强 高磊 等 编著

清华大学出版社



全国计算机技术与软件专业技术资格（水平）考试参考用书

软件设计师考试科目2：

# 软件设计

## ——考点解析与模拟训练

全国计算机技术与软件专业技术资格（水平）考试办公室推荐

刘华 主编 屈红刚 鲁强 高磊 等 编著

清华大学出版社  
北京

## 内 容 简 介

本书是全国计算机技术与软件专业技术资格(水平)考试办公室推荐使用的参考用书,书中内容涵盖了软件设计师考试大纲中科目2(软件设计)的所有知识点,全书的主要内容有:外部设计、内部设计、程序设计、系统实施、软件工程。书中重要章节都包含考点提炼、难点解析、典型例题以及相应的习题和参考答案,有效地帮助考生进行考前复习和训练。

本书适合参加全国计算机技术与软件专业技术资格(水平)考试的考生备考使用,同时也可作为学习软件设计的自学用书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书扉页为防伪页,封面贴有清华大学出版社防伪标签,无上述标识者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

软件设计师考试科目2:软件设计——考点解析与模拟训练/刘华主编;屈红刚等编著. —北京:清华大学出版社,2006.10

(全国计算机技术与软件专业技术资格(水平)考试参考用书)

ISBN 7-302-13605-X

I. 软… II. ①刘… ②屈… III. 软件设计-工程技术人员-资格考核-自学参考资料 IV. TP311.5

中国版本图书馆CIP数据核字(2006)第091688号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

责任编辑:薛 阳

印 刷 者:北京牛山世兴印刷厂

装 订 者:三河市化甲屯小学装订二厂

发 行 者:新华书店总店北京发行所

开 本:185×230 印张:23 防伪页:1 字数:517千字

版 次:2006年10月第1版 2006年10月第1次印刷

书 号:ISBN 7-302-13605-X/TP·8215

印 数:1~4000

定 价:33.00元

# 前 言

全国计算机技术与软件专业技术资格（水平）考试是国家级的专业认定考试，分为计算机软件、计算机网络、计算机应用技术、信息系统、信息服务等 5 个专业类别。每个专业按级别层次划分为初级资格、中级资格和高级资格，并有相应的资格名称。软件设计师属于计算机软件专业，中级资格。

本书是全国计算机技术与软件专业技术资格（水平）考试办公室推荐使用的参考用书。书中内容涵盖了软件设计师考试大纲中考试科目 2（软件设计）的全部内容。全书共分为 5 章，具体内容如下。

第 1 章是外部设计。主要内容包括系统需求说明、系统开发的准备、设计系统功能、设计数据模型、编写外部设计文档、设计评审等内容。

第 2 章是内部设计。主要内容包括设计软件结构、设计输入输出、设计物理数据、构件的创建和重用、编写内容设计文档、设计评审等内容。

第 3 章是程序设计。主要内容包括模块划分、编写程序设计文档、程序设计评审等内容。

第 4 章是系统实施。主要内容包括配置计算机系统及其环境、选择合适的计算机程序设计语言、掌握 C 和 C++ 语言进行程序设计、系统测试等内容。

第 5 章是软件工程。主要内容包括软件生存期模型和软件成本模型的介绍，定义和描述软件需求，软件设计方法以及软件测试的原则与方法，软件质量评测，软件过程评估基本方法，软件能力成熟度评估基本方法，软件开发环境和开发工具、分析工具、设计工具、编程工具、测试工具、维护工具的介绍，软件工程发展趋势，软件过程改进模型和方法等内容。

书中重要的章节都包含考点提炼、难点分析、典型例题、相应的习题和答案，帮助考生进行考前复习和训练。

本书在全国计算机技术与软件专业技术资格（水平）考试办公室的领导下组织编写。全书由刘华主编，另外参与本书编写的还有：屈红刚、鲁强、高磊、刘剑锋、于玥、李林果、李成都等，在此表示衷心的感谢！

由于时间仓促，加之作者水平有限，书中不足之处在所难免，欢迎大家批评指正！

编 者

2005 年 4 月

# 目 录

|                         |     |
|-------------------------|-----|
| <b>第 1 章 外部设计</b> ..... | 1   |
| 1.1 理解系统需求说明 .....      | 2   |
| 1.2 系统开发的准备 .....       | 6   |
| 1.3 设计系统功能 .....        | 25  |
| 1.4 设计数据模型 .....        | 60  |
| 1.5 编写外部设计文档 .....      | 73  |
| 1.5.1 外部设计文档各要素 .....   | 73  |
| 1.5.2 设计系统测试要求 .....    | 88  |
| 1.6 设计评审 .....          | 91  |
| 1.7 模拟训练题 .....         | 95  |
| 模拟训练题答案 .....           | 107 |
| <b>第 2 章 内部设计</b> ..... | 113 |
| 2.1 设计软件结构 .....        | 113 |
| 2.2 设计输入输出 .....        | 136 |
| 2.3 设计物理数据 .....        | 158 |
| 2.4 构件的创建和重用 .....      | 174 |
| 2.5 编写内部设计文档 .....      | 179 |
| 2.6 内部设计评审 .....        | 186 |
| 2.7 模拟训练题 .....         | 193 |
| <b>第 3 章 程序设计</b> ..... | 198 |
| 3.1 模块划分 .....          | 198 |
| 3.2 编写程序设计文档 .....      | 207 |
| 3.3 程序设计评审 .....        | 239 |
| 3.4 模拟训练题 .....         | 242 |
| <b>第 4 章 系统实施</b> ..... | 260 |
| 4.1 系统实施环境 .....        | 260 |
| 4.1.1 系统实施概论 .....      | 261 |

|              |               |            |
|--------------|---------------|------------|
| 4.1.2        | 系统环境设置        | 261        |
| 4.2          | 程序设计语言        | 263        |
| 4.2.1        | 程序设计语言概述      | 264        |
| 4.2.2        | 程序设计语言的选择     | 266        |
| 4.3          | 程序的应用         | 268        |
| 4.3.1        | 编码规范          | 268        |
| 4.3.2        | 应用程序          | 273        |
| 4.4          | 系统测试          | 278        |
| 4.4.1        | 模块测试          | 278        |
| 4.4.2        | 测试环境和测试工具的准备  | 280        |
| 4.4.3        | 测试数据的准备       | 282        |
| 4.4.4        | 测试报告的书写       | 286        |
| 4.5          | 全真模拟训练题       | 288        |
| <b>第 5 章</b> | <b>软件工程</b>   | <b>299</b> |
| 5.1          | 软件生存期模型       | 299        |
| 5.1.1        | 瀑布与演化模型       | 300        |
| 5.1.2        | 螺旋模型          | 302        |
| 5.1.3        | 喷泉模型          | 303        |
| 5.2          | 软件需求分析        | 304        |
| 5.2.1        | 需求分析的目标       | 304        |
| 5.2.2        | 需求的分析过程和获取    | 305        |
| 5.2.3        | 可行性研究和可行性研究报告 | 306        |
| 5.3          | 定义软件需求        | 308        |
| 5.3.1        | 结构化方法         | 308        |
| 5.3.2        | 面向对象方法        | 319        |
| 5.4          | 软件设计          | 322        |
| 5.4.1        | 软件设计的过程       | 322        |
| 5.4.2        | 软件设计的原则       | 323        |
| 5.5          | 软件设计方法        | 326        |
| 5.5.1        | 结构化设计方法       | 326        |
| 5.5.2        | Jackson 设计方法  | 329        |
| 5.5.3        | 对象建模技术        | 331        |
| 5.6          | 程序设计          | 334        |
| 5.6.1        | 结构化程序设计       | 334        |

---

|       |                      |     |
|-------|----------------------|-----|
| 5.6.2 | 面向对象的程序设计 .....      | 340 |
| 5.7   | 软件测试、质量保证和过程改进 ..... | 343 |
| 5.7.1 | 软件测试概述 .....         | 343 |
| 5.7.2 | 软件测试的过程与策略 .....     | 346 |
| 5.7.3 | 软件测试方法 .....         | 350 |
| 5.7.4 | 软件质量管理 .....         | 351 |
| 5.7.5 | 软件过程改进与评估 .....      | 355 |
| 5.8   | 模拟训练题 .....          | 358 |
|       | 模拟训练答案 .....         | 360 |

# 第1章 外部设计

## 本章提示

软件外部设计是软件开发中一个承上启下的重要阶段，是后续开发步骤及软件维护工作的基础。在需求分析阶段内，由系统分析人员对被设计的系统进行系统分析，确定对该软件的各项功能、性能需求和设计约束，确定对文件编制的要求，作为本阶段工作的结果，一般地说，需要确定软件需求说明书、数据要求说明书和初步的用户手册。而软件的外部设计就是在系统需求说明的基础之上，以需求分析所产生的文档为依据，着手系统开发的准备工作和设计实现“需求”的软件体系结构，简称软件结构。系统开发的准备工作主要包括选择开发方法、准备开发环境和制定开发计划等工作。这一阶段确定软件结构的具体任务是将系统分解成子系统，确定各子系统的功能及调用关系，处理流程，将用户的需求分配到适当的位置上去，得出系统的结构图。为了得到好的系统结构，需要一组标准化准则和工具。这一阶段要形成的文档主要包括系统配置图、各子系统关系图、系统流程图、系统功能说明书、输入输出规格说明、数据规格说明、用户手册框架、系统测试要求等。本章将根据考试大纲的要求，全面地介绍理解系统需求说明、系统开发的准备和设计系统功能、设计数据模型、编写外部设计文档和设计评审等方面的主要知识点，并重点介绍其中的难点，在详细的典型例题分析之后，还将给出适量的模拟训练题，以加深对这些知识的理解。图 1-1 是本章的知识框图。

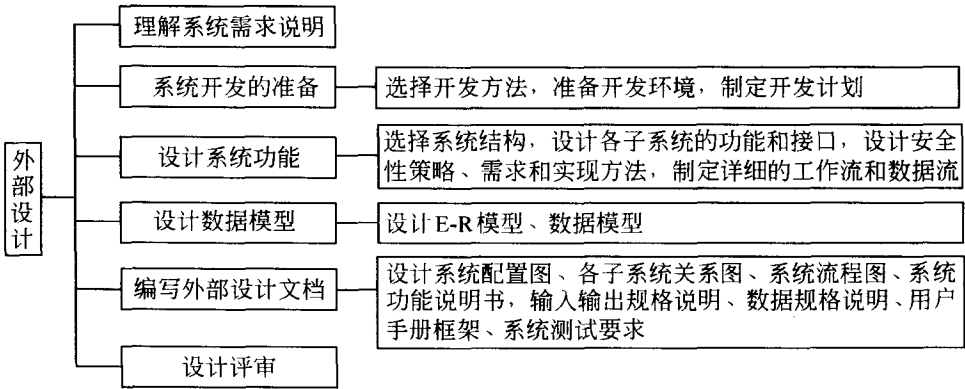


图 1-1 外部设计知识框图

## 1.1 理解系统需求说明

系统需求是系统或软件必须达到的目标和能力，即给出“开发什么”的问题，是否符合系统需求是衡量软件开发成败的最重要标准。随着计算机技术的飞速发展，软件已经成为人们生活中不可缺少的一部分。人们在使用软件的过程中，常常会抱怨它无法执行某些基本操作。但对于软件开发人员而言，用户不断提出新的要求是一件麻烦的事。其实，在软件开发过程中遇到的许多问题，都是由于收集、编写、协商、修改软件需求过程中的失误造成的，诸如信息收集不全、功能不明确、交流不充分、文档不完善、需求发生变化等。开发软件系统最为困难的部分就是准确说明开发什么。如果软件需求设计不合理，将会给系统带来极大的损害，同时对以后的修改带来极大的困难。因此，完全理解软件需求对软件开发的成败起着至关重要的作用。

理解系统需求说明也是进行软件外部设计的首要工作，而且是外部设计是否成功的关键工作之一。只有在对系统需求充分理解的基础上，才可能给出良好的软件外部设计。此阶段的主要工作是阅读和理解软件需求说明书，确认用户要求能否实现，明确实现的条件，从而确定设计的目标以及它们的优先顺序。如图 1-2 所示是本节的知识框图。

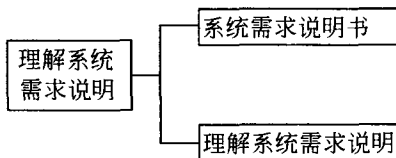


图 1-2 理解系统需求说明知识框图

### 1. 考点提炼

#### (1) 系统需求说明书

软件需求规格说明是需求分析的最终结果，通过建立详细的功能和行为描述、完整的性能需求和设计约束的说明以及恰当的验收标准，精确地阐述一个软件系统必须提供的功能和性能以及它所要考虑的限制条件。软件需求规格说明不仅是系统测试和用户文档的基础，也是所有子系列项目规划、设计和编码的基础。

系统需求说明书具有以下主要作用：

- 软件需求规格说明是用户、分析人员和设计人员之间进行理解和交流的手段。
- 测试人员可以根据软件需求规格说明中对产品行为的描述，制定测试计划，设计测试用例和测试过程。
- 文档人员根据软件需求规格说明和用户界面设计编写用户手册等。
- 软件需求规格说明指导整个系统的开发过程，评审过的需求规格说明需要进行变更控制。

下面给出一个系统需求说明书的模板。

#### T1 引言

##### T1.1 编写目的

说明编写这份软件需求说明书的目的，指出预期的读者。

### T1.2 背景

说明：

- a. 待开发的软件系统的名称；
- b. 本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络；
- c. 该软件系统同其他系统或其他机构的基本的相互来往关系。

### T1.3 定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

### T1.4 参考资料

列出所需的参考资料，如：

- a. 本项目的经核准的计划任务书或合同、上级机关的批文；
- b. 属于本项目的其他已发表的文件；
- c. 本文件中所引用的文件、资料，包括所要用到的软件开发标准。列出这些文件和资料的标题、文件编号、发表日期和出版单位，并说明这些文件与资料的来源。

## T2 任务概述

### T2.1 目标

叙述本次软件开发的意图、应用目标、作用范围以及其他应向读者说明的有关该软件开发的背景材料。解释被开发软件与其他有关软件之间的关系。如果本软件产品是一个独立的软件，而且全部内容自含，则应说明这一点。如果所定义的产品是一个更大的系统的一个组成部分，则应说明本产品与该系统其他各组成部分之间的关系，为此可使用一张方框图来说明该系统的组成和本产品同其他各部分的联系和接口。

### T2.2 用户的特点

列出本软件的最终用户的特点，充分说明操作人员、维护人员的教育水平和技术专长以及本软件的预期使用频度。这些是软件设计工作的重要约束。

### T2.3 假定和约束

列出进行本软件开发工作的假定和约束，例如经费限制、开发期限等。

## T3 需求规定

### T3.1 对功能的规定

用列表的方式（例如 IPO 表即输入、处理、输出表的形式）逐项定量和定性地叙述对软件所提出的功能要求，说明输入量、处理过程以及输出结果，说明软件应支持的终端数和应支持的并行操作的用户数。

### T3.2 对性能的规定

#### T3.2.1 精度

说明对该软件的输入、输出数据精度的要求，可能包括传输过程中的精度。

#### T3.2.2 时间特性要求

说明对于该软件的时间特性要求，如对以下时间的要求：

- a. 响应时间；
- b. 更新处理时间；
- c. 数据的转换和传送时间；
- d. 解题时间。

#### T3.2.3 灵活性

说明对该软件的灵活性的要求，即当需求发生某些变化时，该软件对这些变化的适应能力，如：

- a. 操作方式上的变化；
- b. 运行环境的变化；
- c. 同其他软件的接口的变化；
- d. 精度和有效时限的变化；
- e. 计划的变化或改进。

对于为了提供这些灵活性而进行的专门设计的部分应该加以标明。

#### T3.3 输入输出要求

解释各输入输出数据类型，并逐项说明其媒体、格式、数值范围、精度等。对软件的数据输出及必须标明的控制输出量进行解释并举例，包括对硬拷贝报告（正常结果输出、状态输出及异常输出）以及图形或显示报告的描述。

#### T3.4 数据管理能力要求

说明需要管理的文卷和记录的个数、表和文卷的大小规模，要按可预见的增长对数据及其分量的存储要求做出估算。

#### T3.5 故障处理要求

列出可能的软件、硬件故障以及对各项性能而言所产生的后果和对故障处理的要求。

#### T3.6 其他专门要求

如用户单位对安全保密的要求，对使用方便的要求，对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。

### T4 运行环境规定

#### T4.1 设备

列出运行该软件所需要的硬设备。说明其中的新型设备及其专门功能，包括：

- a. 处理器型号及内存容量；
- b. 外存容量、联机或脱机、媒体及其存储格式，设备的型号及数量；
- c. 输入及输出设备的型号和数量，联机或脱机；
- d. 数据通信设备的型号和数量；
- e. 功能键及其他专用硬件。

#### T4.2 支持软件

列出支持软件，包括要用到的操作系统、编译（或汇编）程序、测试支持软件等。

#### T4.3 接口

说明该软件同其他软件之间的接口、数据通信协议等。

#### T4.4 控制

说明控制该软件的运行的方法和控制信号，并说明这些控制信号的来源。

### (2) 理解系统需求说明

需求分析阶段的主要任务是用户和软件人员双方一起来充分地理解用户的要求，并把双方共同的理解准确地表达成一份书面文档——系统需求说明书。

系统说明书通常包括功能要求、性能要求、可靠性要求、安全保密要求以及开发费用、开发周期、可使用的资源等方面的限制，其中功能要求是最基本的，它又包括数据要求和加工要求两方面。所以，软件阶段的任务是理解软件系统“做什么”，而不是急于考虑“怎么做”。在具体实行时可以以问答的方式进行思考，比如：

- 用户所期望的功能和性能是什么；
- 可靠性和质量问题是哪些；
- 总的系统目标是什么；
- 成本与进度限制如何；
- 市场与竞争情况怎样；
- 有效的技术有哪些……。

## 2. 难点解析

### (1) 理解系统需求的重要性

理解系统需求工作作为软件外部设计的开始，对外部设计的质量，进而对整个软件的质量起着重要的影响作用。只有在对系统需求充分理解的基础上，才可能给出成功的软件外部设计。

### (2) 理解系统需求的主要内容

理解系统需求通常包括明确用户期望的功能和性能是什么，可靠性和质量问题是哪些，总的系统目标是什么，进行系统开发的假定和约束是什么，运行环境规定是什么等。

## 3. 典型例题

**【例 1-1】** 理解系统需求是软件外部设计前的首要工作。下面通常不属于系统需求说明书内容的是（ ）。

- A. 用户对功能和性能的要求
- B. 系统的总体目标
- C. 运行的环境的规定
- D. 子系统如何划分

**【答案】** D

**【解析】** 系统需求是系统或软件必须达到的目标和能力，即给出“开发什么”的问题，而不是“怎么开发的问题”。用户对功能和性能的要求、系统的总体目标和运行的环境的规定都属于理解系统需求的工作，而子系统如何划分是“怎么做”的问题。

## 1.2 系统开发的准备

在充分理解了系统的需求之后,就着手系统开发的准备工作。其主要工作包括选择开发方法、准备开发环境、制定开发计划。选择开发方法主要是根据软件特点,选择一种合适的开发方法,是选择结构化开发方法,还是选择面向对象开发方法或其他方法。准备开发环境,主要是准备软件开发的支撑性环境和一些配套的自动化工具等。制定开发计划,就是制定出整个项目的开发进度、人员投入计划、人员的组织以及资源的利用等内容。如图 1-3 所示是本节的知识框图。

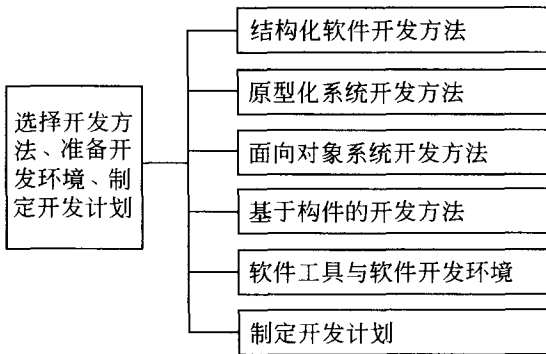


图 1-3 系统开发准备知识框图

### 1. 考点提炼

#### (1) 结构化软件开发方法

结构化开发方法是迄今为止最传统、应用最广泛的一种软件开发方法。结构化系统开发方法的基本思想是,用系统工程的思想 and 工程化的方法,按照用户至上的原则,结构化、模块化、自顶向下地对系统进行分析与设计。

具体来说,结构化开发方法就是先将整个软件系统开发过程划分出若干个相对比较独立的阶段,比如系统规划、系统分析、协调设计、系统实施(系统编码)等。在前三个阶段,坚持自顶而下地对系统进行结构化划分。例如,在给某企业开发管理信息系统时,在需求调查或理顺管理业务时,应从最顶层的管理业务入手,即从组织管理金字塔结构的塔尖入手,层层深入直至最底层。在系统分析、提出新系统方案和系统设计时,应从宏观整体考虑入手,即先考虑系统整体的优化,然后再考虑局部的优化问题。而在系统实施阶段,则应坚持自底向上地逐步组织实施。也就是说,按照前几阶段设计的模块,组织人力从最底层的模块开始编程,然后按照系统设计的结构,将模块一个个拼接到一起进行调试,自底向上、逐渐地构成整体系统。

用结构化系统开发方法开发一个系统时,将整个开发过程划分为 5 个首尾相连的阶段,一般称之为系统开发生命周期(Life Cycle),如图 1-4 所示。

#### ● 系统规划阶段。

系统规划阶段的工作就是根据用户的系统开发请求,初步规划并确定系统开发的可行性。系统规划阶段的主要工作有以下两个:

① 明确问题,初步调查。明确用户要求,初步对对象进行调查。

② 可行性研究。根据初步调查的结果和用户要求,从技术、经济和社会等方面进行可行性分析,以确定系统有无开发的可行性。如果通过可行性研究,则进入下一个阶段开

发。否则停止开发工作进程。

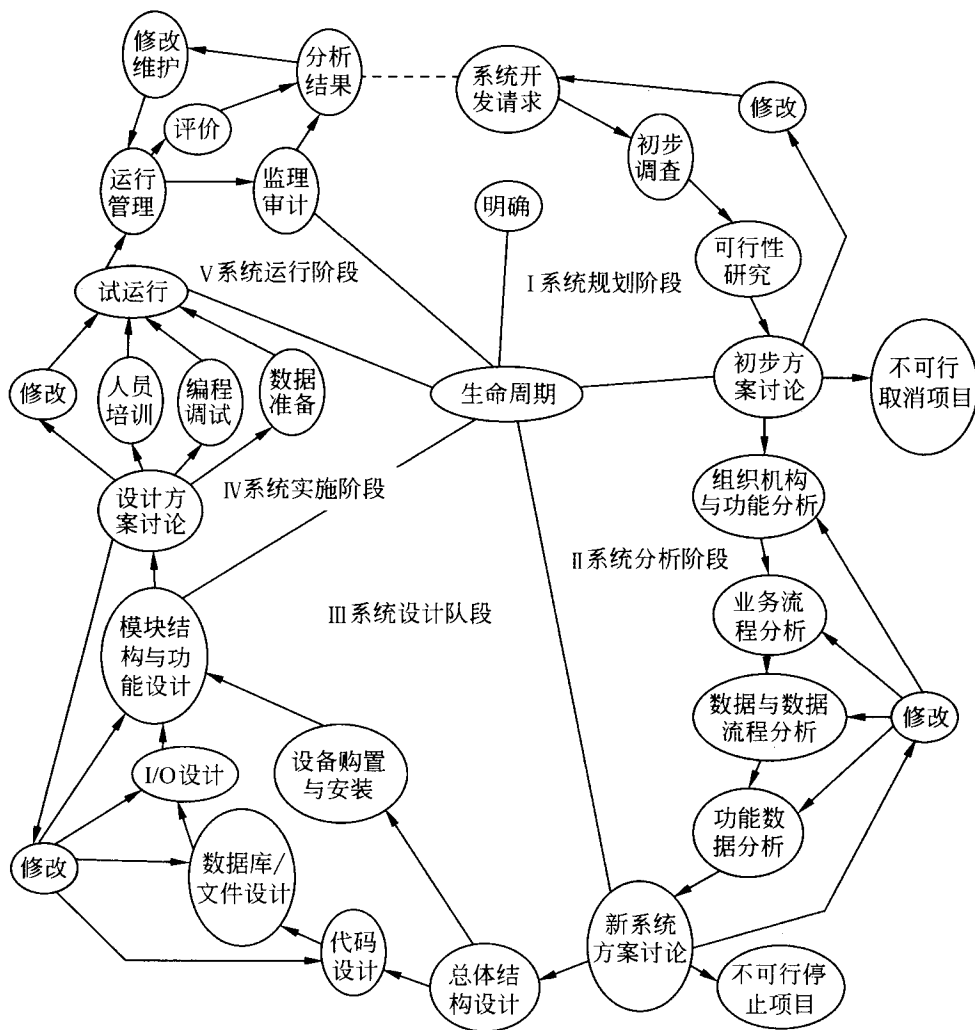


图 1-4 系统开发生命期

#### • 系统分析阶段。

系统分析阶段的任务是对系统进行详细的调查和分析。

系统分析阶段的主要工作有如下 4 项。

① 业务流程分析。详细调查、分析和了解具体的业务流程，并将业务流程优化，最终用业务流程图的方式将优化后的业务流程图确定下来。

② 数据与数据流程分析。对详细调查的数据进行分析和整理，确定数据结构、数据

字典和数据流程图。

③ 功能/数据分析。通过对功能和数据直接关系的分析,确定子系统的结构、网络数据分布等。

④ 提出新系统逻辑方案。

- 系统设计阶段。

系统设计阶段的任务是在前一阶段系统分析的基础上,对系统进行全面设计。

系统设计阶段的主要工作有如下 5 项。

① 总体结构设计。子系统的划分、整体系统结构、软硬件设计配置以及网络结构设计等。

② 代码设计。各种系统代码,例如:设备代码、原材料代码、产品代码、会计科目代码、生成任务代码等。

③ 数据库/文件结构设计。数据库和文件结构(模型)设计等。

④ 输入/输出设计。输入和输出系统的设计。例如:输入/输出界面设计、输入方法和校对方法设计等。

⑤ 模块结构与功能设计。系统结构和各模块之间调用关系设计,以及各模块内部处理功能的详细设计等。

与此同时,根据总体设计的要求购买与安装设备,最终给出设计方案。

- 系统实施阶段。

系统实施阶段的任务是按照前一阶段系统设计的要求进行编程和调试。

系统实施阶段的主要工作有如下 3 项:

① 程序编码与调试。按各模块内部处理功能详细设计的要求,编制程序代码,调试程序模块,以及系统联调等。

② 人员培训以及数据准备。培训系统操作和使用人员,准备系统运行时所需要的数据等。

③ 试运行。

- 系统运行阶段。

系统运行阶段的任务是同时进行体系的日常运行管理、评价、监理审计三部分工作,然后分析运行结果。如果运行结果良好,则送管理部门指导组织生产经营活动;如果有问题,则要对系统进行修改或者是局部调整;如果出现了不可调和的大问题(这种情况一般是在系统运行若干年后,系统运行的环境已经发生了根本的变化时才可能出现),则用户将进一步提出开发新系统的要求,这标志着老系统生命的结束,新系统的诞生。这就是所谓的系统开发生命周期。

结构化软件开发方法具有以下主要特点。

① 面向用户的观点。

用户的要求是一个系统开发建设的出发点和归宿。软件是为用户服务的,最终要交给

用户去使用。因此,软件的成败取决于用户是否对其满意,是否符合了用户的要求。结构化方法充分重视了需求分析在软件开发中的作用和地位。

② 严格划分工作阶段。

结构化方法的每一个阶段都有明确的工作内容,前一阶段的工作成功是后一阶段的工作依据,工作任务要严格按照工作阶段的顺序来安排,以避免造成开发工作的返工、延期甚至失败。

③ 运用结构化系统分析设计方法。

采用自顶而下、逐层分解细化和模块化的思想,便于系统的分析、设计、实施和维护。

④ 文档资料标准化、规范化。

软件开发是一项内容复杂、耗时较长的系统工程,为了保证工作的连续性,在每个阶段的工作完成之后,要将工作结果按要求形成相应的图表和文档资料。这些资料既是开发过程开发人员与用户共同完成的成果,又是今后系统维护的重要依据。

结构化方法克服了此前在软件开发中“软件作坊”式开发方法的诸多弊端,是在信息系统开发中最成熟、应用最广泛的一种工程化方法。当然,该方法也有一些局限性和不足之处。

⑤ “需求冻结”问题。根据结构化方法的特点和要求,开发工作严格地分阶段进行,上一阶段的工作成果是下一阶段工作的依据。因此,如果在系统分析阶段系统的需求不能确定,需要经常变动调整的话,会导致后面的工作返工甚至无法进行。而且,由于用户对计算机和软件系统的开发不太熟悉,而系统分析人员往往缺乏对系统业务的了解和认识,再加上双方在相互理解和交流上存在的偏差与误解,因此,很难一次对系统的需求进行准确的描述。

⑥ 开发周期较长。这样一方面使得用户在较长的时间里不能得到,甚至不能感觉到一个可以实际运行的物理系统;另一方面,由于周期较长,系统的外部环境必定有变化,这就使得系统的需求不得不随之改变。

⑦ 结构化方法要求的文档资料较多,而且内容较细,这也导致了文档审批的困难。

(2) 原型化系统开发方法

原型化系统开发方法(简称原型方法)是20世纪80年代随着计算机软件技术的发展,特别是关系数据库系统、第四代程序生成语言和各式各样的系统开发方法生成环境产生的基础上,而推出的一种新的系统开发方法。与前面介绍的结构化方法相比,它摒弃了那一步步周密细致地调查分析,然后逐步整理出文字档案,最后才能使用户看到结果的繁琐作法。而是一开始就凭借着系统开发人员对用户要求的理解,在强有力的软件环境支持下,给出一个实实在在的系统原型,然后与用户反复协商修改,最终形成实际系统。

原型方法的工作流程如图1-5所示。首先用户提出开发要求,然后开发人员识别和归纳用户要求,根据识别归纳的结果,构造出一个原型(即程序模块),然后同用户一起评价这个原型。如果做不到这一点,则回到第3步重新构造原型;如果不满意原型,则修改原

型，直到用户满意为止，这就是原型方法工作的一般流程。

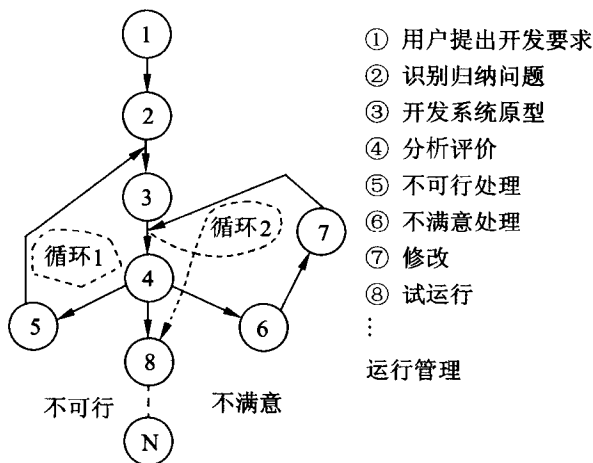


图 1-5 原型方法的工作过程

从原型法的开发步骤可以看出，原型法通过对系统原型的反复试用、评价和修改，给开发人员和用户双方创造了一个不断学习、实践和提高的机会，这一过程与人们对事物的认识过程是一致的。原型法的特点主要体现在以下几方面。

① 原型法可以使系统开发的周期缩短、成本降低、开发速度加快，获得较高的开发效益。

② 原型法是以用户为中心来开发系统的，用户参与的程度大大提高，开发系统符合用户的需求，因而增加了用户的满意度，提高了系统开发的成功率。

③ 由于用户参与了系统开发的全过程，对系统的功能和结构容易理解和接受，使得系统的移交工作能够比较顺利地完

但是原型法的使用是有一定的适用范围和局限性的。主要表现在以下 4 个方面。

① 对于一个大型的系统，如果不经过程序分析来进行整体性规划，想要直接用屏幕来一个一个地模拟或只靠用户口头叙述是很困难的。

② 对于大量运算、逻辑性较强的程序模块，原型法很难构造出模型来供人评价。因为这类问题没有那么多的交互方式（如果有现成的数据或逻辑计算软件包，则情况例外），也不是三言两语就可以把问题说得清楚的。

③ 对于原基础管理不善，信息处理过程混乱的问题，使用有一定的困难。首先是由于对象工作过程不清，构造原型有一定的困难；其次是由于基础管理不好，没有科学合理的方法可依，系统开发容易走上机械地模拟原来手工系统的轨道。

④ 对于一个批处理系统，其大部分是内部处理过程，这时用原型方法也有一定的困难。因此，在实际系统开发过程中，人们常常将原型法与结构化法相结合来开发系统。即先用