

## 传统的软件开发方法

### 本章学习要点：

- (1) 掌握结构化分析方法,了解软件生命周期的阶段划分。
- (2) 掌握问题的定义和可行性研究方法。
- (3) 重点掌握需求分析、系统设计、测试和维护阶段的主要任务。

### 5.1 知识点

#### 5.1.1 可行性研究

可行性研究,也称为可行性分析,是在问题定义基础上进行的。其目的是以相对短的时间和相对低的成本来确定给定的问题是否有解。

可行性分析从客户角度主要是从经济可行性、社会可行性以及操作可行性等几个方面分析解决方案是否合理。从开发方角度主要从技术可行性、调度可行性等方面分析方案是否可行。

可行性研究的主要步骤如下：

- (1) 复查定义,确定项目的规模和目标,明确限制和约束条件。
- (2) 研究现行系统。
- (3) 导出新系统的高层逻辑模型。
- (4) 导出并评价各种解决方案。
- (5) 编写可行性研究报告。

我国制定了一个可行性研究报告的建议性标准,该标准主要包括 8 部分。

- (1) 引言,包括编写目的、背景、定义、参考资料 4 部分;
- (2) 可行性研究的前提;
- (3) 对现有系统的分析;
- (4) 所建议的系统;
- (5) 可选择的其他系统方案;
- (6) 投资及效益分析;
- (7) 社会因素方面的可行性;

(8) 结论：说明进行可行性研究的结论。

## 5.1.2 需求分析

### 1. 需求分析概述

软件需求就是客户对所开发的新系统所应具有功能以及这些功能的表现情况所做出的要求，需求是客户或用户所有待解决问题及对新系统要求的总结，必须经过所有与新系统开发和使用相关的人员的认可。

需求可以分成 3 大类：功能性需求、非功能性需求和约束。

- (1) 功能性需求：描述系统所提供的功能或服务。
- (2) 非功能性需求：系统应提供什么质量的功能或服务。软件的非功能性需求包括可靠性、易用性、有效性、可维护性、可移植性。
- (3) 约束描述对系统的设计或开发过程的限制，常见的约束就是时间和费用上的约束。

对需求的描述应满足以下要求：清晰性、简洁性、一致性、无歧义性、有意义性、可验证性、唯一性、完整性、可追踪性、有界性。

需求分析是一种软件工程活动，其主要步骤包括：

- (1) 需求采集和获取；
- (2) 需求分析与综合；
- (3) 需求评审和确认；
- (4) 需求变更管理。

### 2. 需求分析原则

著名的软件工程领域专家 Pressman 将其总结为 5 条原则：

- (1) 必须表示和理解问题的信息域。
- (2) 必须定义软件所要完成的功能。
- (3) 必须表示软件的行为。
- (4) 必须划分出软件的数据、功能和行为模型，并使用分层的方法揭示细节。
- (5) 分析过程应该遵循自顶向下、逐层细化的原则。

另外，Davis 建议了一组针对需求工程的指导性原则：

- (1) 在开始建立分析模型之前首先理解问题。
- (2) 开发原型，使用户能够了解将如何发生人机交互。
- (3) 记录每个需求的起源和原因。
- (4) 使用多个需求视图，建立数据模型、功能模型和行为模型。
- (5) 给需求赋予优先级。
- (6) 努力消除歧义性。

### 3. 数据流程图

功能建模着重描述用户对系统的功能需求，在结构化需求分析中，数据流程图是进行功能建模的一种应用最广泛的模型。数据流程图描述了计算机系统的 4 大基本功能：输入、处理、输出和存储。数据流程图有两大目标：指明数据在系统中移动时如何被变换；描述对数据流进行变换的功能。

数据流程图中共有 4 种基本元素，其符号如图 5-1 所示。

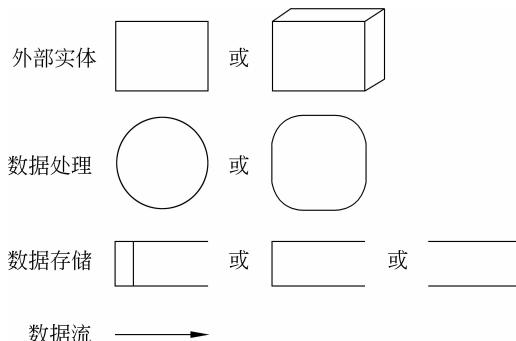


图 5-1 数据流程图 4 种基本元素

(1) 外部实体：为系统提供数据或接收系统输出数据的系统外的事物，是系统边界之外的个人、组织或与该系统有交互关系的其他系统。

(2) 数据处理：又称数据加工或数据变换，是对数据进行处理的地方，将输入数据按照某种方法进行处理，产生输出数据。

(3) 数据存储：保存数据的地方，以供数据处理使用。

(4) 数据流：表示在数据处理、数据存储以及外部实体之间移动的数据，箭头表示数据流动的方向。

一般将数据流程图分为 3 个层次进行描述：总体图、零级图和细节图。

(1) 总体图：由一个数据处理结点(一般用于表示系统)、若干外部实体结点以及数据流组成，用于描述系统与外部实体之间的关系。

(2) 零级图：由若干外部实体、数据处理、数据存储以及数据流组成，描述一个系统的主要子系统(或功能)。

(3) 细节图：由数据处理、数据存储以及数据流组成，描述零级图中某个复杂数据处理的详细过程。

### 4. 数据流程图绘制

(1) 绘制数据流程图应遵循如下规则：

- ① 每个数据处理都有编号；
- ② 细节图和零级图的编号是相互关联的；
- ③ 每个数据存储也有编号；
- ④ 同一个数据处理或数据存储可以在图中不同位置画多个，但其编号相同。

(2) 判断数据流程图绘制正确与否的规则如下：

- ① 外部实体只出现在总体图和零级图中。
  - ② 数据存储只出现在零级图和细节图中。
  - ③ 数据存储在分层的数据流程图中只能出现在某一层上,即同一个数据存储不能在不同的层次上出现。
  - ④ 图中的每个数据流必须与某个数据处理相关联。
  - ⑤ 数据流的源点和目标不能是同一个数据处理。
  - ⑥ 数据流仅仅表示数据的流动,不表示有关的控制逻辑。
  - ⑦ 系统的输入输出命令不能作为图中的数据处理。
  - ⑧ 每个数据处理和数据存储都应该既有输入数据流,也有输出数据流。
  - ⑨ 分层的数据流程图各层之间应保持平衡。
  - ⑩ 表示数据流的箭头上要标有数据流的名字,但流入或流出数据存储的数据流除外。
  - ⑪ 图中每个表达相同含义的元素都应使用唯一的名称,没有两个表达不同含义的元素的名称是相同的。
- (3) 衡量数据流程图绘制质量的规则如下：
- ① 图中的每个数据处理都应有编号,但编号不表示执行的先后顺序。
  - ② 图中的每个数据存储都应有编号,一般使用“D+序号”的形式。
  - ③ 图的绘制尽量遵守  $7 \pm 2$  规则。
  - ④ 同一个数据处理、数据存储或外部实体可以在图中不同位置重复出现,但其编号和名称应完全相同。
  - ⑤ 一个数据处理的输出不应等同于其输入。

## 5. 行为建模与状态变迁图

数据建模和功能建模描述了系统的静态特征,行为建模用于描述系统的动态特征。行为建模中最常用的方法有使用状态变迁图、Petri 网等。

状态变迁图(STD)通过描述系统的状态,以及导致系统状态发生改变的事件,描述系统的行为。

状态变迁图中共有两种符号。

- (1) 状态：使用圆圈表示,在圆圈内部写上系统所处的状态名称。
- (2) 变迁：使用连接两个状态圆圈的箭头表示从一种状态到另一种状态的变迁,在箭头上需要标注出引起这种状态迁移的事件名称。

## 6. 数据字典

数据字典可对数据流程图中所出现的所有被命名的图形元素进行详细定义和描述,使得每个图元都有一个确切的解释说明,可使阅读数据流程图的各方在一个统一的语境中进行交流沟通。

数据字典主要描述对象包括数据元素、数据流、数据存储、数据处理和外部实体。

## 7. 软件需求说明书

软件需求说明书是总结用户需求,进行下一步工作的关键,我国国家标准中规定的软件需求说明书主要包括4部分:

- (1) 引言,包括编写目的、背景、定义、参考资料等内容。
- (2) 任务概述,包括目标、用户的特点、假定和约束等内容。
- (3) 需求规定,包括对功能的规定、对性能的规定、输入输出要求、数据管理能力要求、故障处理要求、其他专门要求等内容。
- (4) 运行环境规定,包括设备、支持软件、接口、控制等内容。

### 5.1.3 系统设计

#### 1. 软件设计原则

软件系统设计的基本目标是使所设计的系统满足系统逻辑方案(即需求分析的结果)所规定的各项功能要求,同时还要满足其非功能性要求,为此需要用比较抽象概括的方式确定新系统如何完成预定的任务,即确定系统的物理模型。

从技术角度看,软件系统设计包括数据设计、体系结构设计、接口设计和过程设计4部分。

从管理角度看,软件系统设计一般包括概要设计(也称为总体设计)和详细设计两个阶段。概要设计涉及数据设计和软件体系结构设计,详细设计涉及过程设计,而接口设计在概要设计和详细设计阶段都存在,只是层次有所区别。

软件系统设计应遵循一些基本原则,包括指导性原则和操作性原则。

指导性原则主要包括以用户为中心、适用性、简单性、完整性、一致性、高质量、复用性等;操作性原则主要包括模块化、抽象、信息隐藏、模块独立性等。

1978年Meyer提出了衡量模块独立性的两个重要指标:内聚和耦合。内聚用来度量模块内部各元素之间联系的紧密程度。模块内部各元素之间联系越紧密,其内聚程度越高。内聚程度越高的模块,其独立性越强。耦合用来度量模块间联系的紧密程度。耦合程度取决于模块之间接口的复杂程度、调用方式以及接口中信息的形式等,耦合程度越低越好。耦合程度高称为紧耦合,耦合程度低称为松耦合。

#### 2. 结构化设计方法

概要设计主要涉及数据设计和软件体系结构设计。数据设计阶段的主要任务是设计系统的公共数据结构及数据库逻辑结构。软件体系结构设计阶段通过数据流程图推导出结构图进行描述,其推导方法主要有一一对应法、基于变换流法和基于事务流法等。

详细设计主要涉及过程设计以及细粒度的接口设计。过程设计主要通过程序流程图或PDL语言等方法描述模块内部实现的逻辑。

界面设计,也称为人机交互设计,指人与计算机系统进行交互的通信媒体和手段,在现代软件工程中占有非常重要的地位。界面设计主要完成:

- (1) 设计交互方式和界面总体结构;
- (2) 设计界面的风格;
- (3) 设计界面之间的转换关系。

### 3. 软件设计文档

我国制定的国家标准中将软件设计文档分成了3个文档:数据库设计说明书、概要设计说明书和详细设计说明书。

数据库设计说明书主要用于对数据设计的结果之一——数据库进行说明,包括引言、外部设计、结构设计和运用设计等内容。

概要设计说明书对概要设计的结果进行说明,包括引言、总体设计、接口设计、运行设计、系统数据结构设计和系统出错处理设计等。

详细设计说明书对详细设计的结果进行说明,包括引言、程序系统的结构、程序(标识符)设计说明等。

#### 5.1.4 系统测试与维护

测试是为了验证软件是否满足用户的需求而执行程序的过程。测试的目的不是为了证明所开发出的系统是正确的,而是为了发现系统中存在的问题。

测试的内容:一是进行功能测试,测试系统的功能是否满足功能性需求,二是进行性能测试,测试系统的性能是否满足非功能性需求。

测试的基本原则主要包括设计合理的测试用例,尽早地开展测试工作,采用第三方进行测试,充分利用Pareto原则,重视回归测试。

##### 1. 软件测试方法

白盒测试,又称玻璃盒测试、结构测试,是指接受测试的模块其内部实现逻辑对测试人员是可见的,测试用例完全根据该模块的实现代码导出。

黑盒测试,也称为功能测试,是指测试用例只根据接受测试模块的接口导出,而完全不考虑该模块的内部实现逻辑。

##### 2. 测试的实施过程

软件测试实施的过程:单元测试→集成测试→确认测试。

单元测试主要进行模块的接口测试、局部数据结构测试、路径测试和出错处理测试等内容,主要采用白盒测试和黑盒测试方法。

集成测试对系统组装后各模块之间配合情况进行测试。一般采用黑盒测试法进行,主要用于检查模块间接口错误和全局数据结构错误等。集成测试主要分3种方式进行:自顶向下、自底向上、自顶向下和自底向上结合。

确认测试是在用户参与下进一步检查软件的功能和性能是否满足用户的需求。

一般采用黑盒测试法进行,测试过程分为有效性测试、 $\alpha$  测试、 $\beta$  测试和验收测试 4 个阶段。

### 3. 系统维护

在软件的生命周期中,软件维护是持续时间最长、花费时间和费用最多的一个阶段。一般将软件维护分为 4 类:正确性维护、适应性维护、完善性维护和预防性维护。

软件的可维护性可从 5 个方面进行衡量:可理解性、可测试性、可修改性、可靠性和可移植性。

系统维护的实施过程主要包括维护申请、维护申请审查、维护实施、总结及复审等几个阶段。

## 5.2 例题解析

### 1. 选择题

(1) 下面哪一个不属于软件项目计划的内容\_\_\_\_\_。

- A) 详细的项目实施范围
- B) 人力资源计划
- C) 确定项目目标
- D) 制定项目实施的时间计划

【答案】 C

【解析】 软件项目计划内容包括范围,对该软件项目的综合描述,定义起所要做的工作以及性能限制;资源,包括人力资源、硬件资源、软件资源等;进度安排。

(2) 下面哪一个不是制订软件项目计划的方法策略\_\_\_\_\_。

- A) 注重项目计划的层次性
- B) 重视与客户的沟通
- C) 开始尽量简略
- D) 制订的项目计划要现实

【答案】 C

【解析】 制订软件项目计划的方法策略主要包括注重项目计划的层次性;重视与客户的沟通;该详细的详细,该简略的就简略以及制订的项目计划要现实。

(3) 在各种不同的软件需求中,\_\_\_\_\_是从各个角度对系统的约束和限制,反映了应用对软件系统质量和特性的额外要求。

- A) 业务需求
- B) 功能需求
- C) 非功能需求
- D) 用户需求

【答案】 C

【解析】 功能需求描述了用户使用产品必须要完成的任务,可以在用例模型中予以说明,而非功能需求是从各个角度对系统的约束和限制,反映了应用对软件系统质量和特性的额外要求。

(4) 模块的独立性是由内聚性和耦合性来度量的,其中内聚性是指\_\_\_\_\_。

- A) 模块间的联系程度
- B) 模块的功能强度
- C) 信息隐蔽程度
- D) 接口的复杂程度

【答案】 B

【解析】 内聚性是度量一个功能模块强度的相对指标。耦合性则用来度量模块之间相互联系的程度,同时模块的接口的复杂程度是关于模块耦合性的,信息隐蔽程度主要是关于模块耦合性的。本题答案选 B。

(5) 在面向数据流的软件设计方法中,一般将信息流分为\_\_\_\_\_。

- A) 变换流和事务流
- B) 变换流和控制流
- C) 事务流和控制流
- D) 数据流和控制流

【答案】 A

【解析】 在面向数据流的软件系统的设计中,将信息流分为两类流,即变换流和事务流,通过对这两类信息流的不同映射,有数据流图转换成软件结构。答案为 A。

(6) 黑盒测试方法是根据\_\_\_\_\_设计测试用例。

- A) 程序调用规则
- B) 模块间的逻辑关系
- C) 软件要完成的功能
- D) 数据结构

【答案】 C

【解析】 黑盒测试的测试用例的基本原则如下:不关心程序内部的逻辑,只是根据程序的功能说明来设计测试用例,测试时完全不考虑程序内部细节、结构和实现方式,仅检验程序结果与说明书的一致性。所以在使用黑盒测试法时,只要有程序功能说明即可。答案为 C。

(7) 对软件进行分解,是为了\_\_\_\_\_。

- A) 降低模块的复杂程度
- B) 降低模块间接口的复杂程度
- C) 降低模块的复杂程度,并降低模块接口间的复杂程度
- D) 降低模块的复杂程度,并提高模块接口间的复杂程度

【答案】 A

【解析】 大型软件系统是十分复杂的,人们对付复杂系统的手段是“抽象”和“分解”,把一个大问题分解成若干个小问题。使用结构化设计方法进行设计时,就是用分解的方法将大系统分解为多个功能模块。而分解的目的就是为了降低每一个模块的复杂程度。当觉得模块仍比较复杂时,就应对其进行进一步分解。

分解虽然能够降低模块的复杂程度,但会增加模块间的复杂程度。但题目问的是对软件进行分解的目的,很显然提高模块接口间的复杂程度并非分解的目的,所以答案 D 不正确,本题答案为 A。

(8) 下面哪一项不是系统设计的依据\_\_\_\_\_。

- A) 系统分析的成果
- B) 现行技术
- C) 软件开发工期
- D) 用户需求

【答案】 C

【解析】 系统设计阶段工作的主要依据可从如下几个方面考虑:系统分析的成果、现行技术、现行的信息管理和信息技术的标准、规范和有关法律制度、用户需求、系统运行环境。

(9) 下面哪一项与绘制系统流程图无关\_\_\_\_\_。

- A) 信息处理的步骤和内容
- B) 输入输出的内容
- C) 各步骤之间的物理和逻辑关系
- D) 系统配置环境

【答案】 D

【解析】 绘制系统流程图时应主要根据信息处理的步骤和内容,每一步骤所涉及的物理过程。主要包括输入输出的内容和形式、存储要求、加工所用的物理设备、各步骤之间的物理和逻辑关系。

(10) 模块的组成部分有\_\_\_\_\_。

- A) 输入与输出
- B) 运行环境
- C) 逻辑功能
- D) A、B、C

【答案】 D

【解析】 模块是指一组程序语句,它包括输入输出和逻辑处理功能,内部信息及其运行环境。

(11) 结构化程序的基本思想有\_\_\_\_\_。

- A) 任何程序的形成和编写都必须按一定的规范和格式进行
- B) 按照自顶向下(分层)逐步求精(细化)模块化的原则编写
- C) 结构化的程序只能够由顺序、选择和循环3种基本结构构成并允许嵌套、每个程序模块应该单入口/单出口
- D) A、B、C

【答案】 D

【解析】 结构化程序的基本思想包括:(1)任何程序的形成和编写都必须按一定的规范和格式进行;(2)按照自顶向下(分层)、逐步求精(细化)、模块化的原则编写程序;(3)结构化的程序只能够由顺序、选择和循环3种基本结构构成并允许嵌套、每个程序模块应该单入口/单出口;(4)结构化程序具有好的可读性。

(12) 一个可读性良好的程序应具有\_\_\_\_\_。

- A) 程序内部文档
- B) 数据说明
- C) 语句构造简单而直接
- D) A、B、C

【答案】 D

【解析】 一个可读性好的程序应具有程序内部文档,数据说明,语句构造简单而直接,要遵循有关输入输出的规则,有一定的时间效率和存储效率。

(13) 软件测试目的是\_\_\_\_\_。

- A) 评价软件的质量
- B) 发现软件的错误
- C) 改正软件的错误
- D) 证明软件正确

【答案】 B

【解析】 测试的目的是发现软件程序中的错误,找出错误的原因和位置并加以改正。

(14) 为了提高测试的效率,应该\_\_\_\_\_。

- A) 随机选取数据
- B) 取一切可能的输入数据作为测试数据

- C) 在完成编码以后制定软件的测试计划
- D) 选择发现错误可能性大的数据作为测试数据

**【答案】 D**

**【解析】** 不同的测试数据发现程序错误的能力差别很大。为了能以最少的时间及人力找出最多的系统错误,应设计出合理的测试用例。合理的测试用例应该满足两项要求:为了有效地测试,必须采取能够尽可能多地发现错误的测试用例;为了高效地测试,必须用尽可能少的测试用例发现尽可能多的错误。

- (15) 下面哪一项不属于软件维护的副作用\_\_\_\_\_。
- A) 程序代码进行修改有可能带来新的错误
  - B) 修改数据及数据结构常给软件带来灾难性的错误
  - C) 对源代码的修改没有反映在设计文档或用户手册中
  - D) 修改后的新界面不符合用户的习惯

**【答案】 D**

**【解析】** 维护的副作用是指由于修改而导致的错误或其他多余的动作的发生。软件维护的副作用主要体现在以下 3 个方面:程序代码修改的副作用有可能带来新的错误;修改数据及数据结构常给软件带来灾难性的错误;修改文档的副作用是指对源代码的修改没有反映在设计文档或用户手册中。

- (16) 下面有关软件维护的叙述不正确的是\_\_\_\_\_。
- A) 要维护一个软件,必须先理解这个软件
  - B) 在软件维护时很可能引入新的差错
  - C) 由谁编写的软件就由谁来维护
  - D) 无论是否有文档,都可以进行软件的维护

**【答案】 C**

**【解析】** 负责进行维护的人员往往不是进行软件开发的人员。

## 2. 填空题

- (1) 结构化设计思想的目标是一个\_\_\_\_\_、\_\_\_\_\_的结果良好的软件系统。

**【答案】** 耦合度尽可能小、内聚度应尽可能强

**【解析】** 结构化设计的目标就是一个耦合度尽可能小、内聚度应尽可能强的结果良好的软件系统。由于结构化设计方法不仅可以和需求阶段所采用的结构化分析方法很好地衔接,而且还能与编码阶段的结构化程序设计相适应,因此成为广泛使用的设计方法。

- (2) 数据处理问题的两种类型为\_\_\_\_\_和\_\_\_\_\_。

**【答案】** 变换型、事物处理型

**【解析】** 结构化方法是面向数据流图的,需要根据数据流图确定问题的类型,数据处理问题有两种结构:变换型和事物处理型。

- (3) 软件项目的进度安排需要首先识别一组项目任务,建立任务间的相互关联,然后估计各个任务的工作量,\_\_\_\_\_ ,最后指定进度时序。