

第一章 软件工程概论

1. 什么是软件危机？

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题表现在以下几个方面：

- (1)用户对开发出的软件很难满意。
- (2)软件产品的质量往往靠不住。
- (3)一般软件很难维护。
- (4)软件生产效率很低。
- (5)软件开发成本越来越大。
- (6)软件成本与开发进度难以估计。
- (7)软件技术的发展远远满足不了计算机应用的普及与深入的需要。

2. 为什么会产生软件危机？

(1)开发人员方面，对软件产品缺乏正确认识，没有真正理解软件产品是一个完整的配置组成。造成开发中制定计划盲目、编程草率，不考虑维护工作的必要性。

(2)软件本身方面，对于计算机系统来说，软件是逻辑部件，软件开发过程没有统一的、公认的方法论和规范指导，造成软件维护困难。

(3)尤其是随着软件规模越来越大,复杂程度越来越高,原有软件开发方式效率不高、质量不能保证、成本过高、研制周期不易估计、维护困难等一系列问题更为突出，技术的发展已经远远不能适应社会需求。

3. 怎样克服软件危机？

(1)充分吸收和借鉴人类长期以来从事各种工程项目中积累的行之有效的有效原理、概念、技术与方法，特别是吸取几十年来人类从事计算机硬件研究和开发的经验教训。在开发软件的过程中努力作到良好的组织，严格的管理，相互友好的协作。

(2)推广在实践中总结出来的开发软件的成功的技术和方法，并研究更好、更有效的技术和方法，尽快克服在计算机系统早期发展阶段形成的一些错误概念和作法。

(3)根据不同的应用领域，开发更好的软件工具并使用这些工具。将软件开发各个阶段使用的软件工具集成成一个整体，形成一个很好的软件开发支环环境。

总之为了解决软件危机，既要有技术措施(方法和工具)，又要有必要的组织管理措施。

4. 构成软件项目的最终产品：

应用程序、系统程序、面向用户的文档资料和面向开发者的文档资料。

5. 什么是软件生存周期？

软件生存周期是指从软件定义、开发、使用、维护到淘汰的全过程。

6. 软件生存周期为什么划分成阶段？

(1) 任何一个阶段的具体任务不仅独立，而且简单，便于不同人员分工协作，从而降低整个软件开发工作的困难程度。

(2) 可以降低每个阶段任务的复杂程度，简化不同阶段的联系，有利于工程的组织管理，也便于采用良好的技术方法。

(3) 使软件开发的全过程以一种有条不紊的方式进行，保证软件的质量，特别是提高了软件的可维护性。

7. 应该怎样来划分阶段？

(1) 每一个阶段的任务尽可能独立；

(2) 同一阶段内的任务性质尽可能相同；

(3) 每一个阶段任务的开始和结束有严格的标准。

8. 软件开发模型有几种？它们的开发方法有什么特点？

软件开发模型有瀑布型、渐增型和变换型。

瀑布型开发方法是按照软件生存周期的划分依次实施，每一个阶段有明确规定的任务。它的特点：

(1) 各个阶段的顺序性和依赖性；

(2) 划分逻辑设计与物理设计，尽可能推迟程序的物理实现；

(3) 每个阶段必须完成规定的文档，对其中问题通过复审及早发现，及早解决。

渐增型开发方法及特点：

(1) 从部分需求出发，先建立一个不完全的系统，通过测试运行该系统取得经验和信息反馈，加深对软件需求的理解，进一步使系统扩充和完善。如此反复，直至软件人员和用户对所设计完成的软件系统满意为止。

(2) 在渐增型开发下的软件是随软件开发的过程而逐渐形成的。

(3) 渐增型开发方法适合于知识型软件的开发，设计系统时对用户需求的认识开始不是很清楚的，需要在开发过程中不断认识、不断获得新的知识去丰富和完善系统。多数研究性质的试验软件，一般采用此方法。

变换型开发方法及特点:

(1)从软件需求的形式化规格说明出发,经过一系列的程序变换,得到最终的程序系统。

(2)该方法必须有严格的数学理论和形式化技术的支持。

9.什么是软件工程?

软件工程是指导计算机软件开发和维护的工程学科。

(1)它采用工程的概念、原理、技术和方法来开发和维护软件;

(2)它将管理技术与当前经过时间考验的而证明是正确的技术方法结合起来;

(3)它强调使用生存周期方法学和结构分析和结构技术;

(4)经过人们长期的努力和探索,围绕着实现软件优质高产这个目标,从技术到管理两个方面做了大量的努力,逐渐形成了 软件工程学 这一新的学科。

10. 什么是软件工程环境:

方法与工具的结合,加上配套的软、硬件支持称为软件工程环境。它能支持开发者按照软件工程的方法,全面完成生存周期中的各项任务。

第二章 可行性研究 习题答案

1. 问题定义的任务和主要工作?

问题定义的任务:将用户提出的要求具体化、量化;确定研制系统的范围,明确研制的边界。

问题定义阶段的工作:

(1)通过调查研究,了解系统需求;

(2)确定系统的功能需求、性能需求、可靠性需求、安全及保密性、资源、开发费用及开发进度等的需求;

(3)问题定义阶段的产品—系统目标与范围说明书。

2.可行性研究目的?

确定在问题定义中所提出的问题是否值得去解,在限制条件下,问题能否解决。

3. 可行性研究的任务?

(1)进一步分析和澄清问题的定义,在澄清问题的基础上,导出系统的逻辑模型;

(2)从系统逻辑模型中,选择问题的若干种主要解法,研究每一种解法的可行性,为以后的行动提出建议;

(3)如果问题没有可行的解,建议停止系统开发;如果问题有可行的解,应该推荐一个较好的解决方案,并为工程制定一个初步的计划。

4. 可行性研究包括哪几方面的内容?

(1)技术可行性:现有技术能否实现本系统,现有技术人员能否胜任,开发系统的资源能否满足;

(2)经济可行性:经济效益是否超出开发成本;

(3)操作可行性:系统操作在用户内部行得通吗?

(4)法律可行性:新系统开发是否会侵犯他人、集体或国家利益,是否违反国家法律。

5. 可行性研究的步骤?

(1)复查系统的规模和目标;

(2)研究目前正在使用的系统,总结现有系统的优劣,提出新系统的雏形;

(3)导出新系统的高层逻辑模型;

(4)推荐建议方案;

(5)推荐行动方针;

(6)书写计划任务书(可行性报告);

(7)提交审查。

6. 可行性研究报告的主要内容?

可行性分析的结果是可行性研究报告,内容包括:

(1)系统概述:说明开发的系统名称,提出单位和开发单位。

(2)可行性研究的前提:系统目标;要求;约束和限制;可行性研究的基本准则等。

(3)对现有系统的分析:处理流程,图示说明现有系统的处理流程和数据流程;现有系统存在的问题。

(4)系统需求:主要功能;主要性能及其要求;操作要求;信息要求;限制性要求。

(5)建议系统:系统目标;处理流程;系统结构,功能,性能;系统技术可行性;投资和效益分析;操作可行性;法律可行性。

(6)其它可选方案:与国内外同类型方案的比较;提出一两个可行性方案供论证和探讨。

(7)制定下一阶段的预算。

(8)结论性意见:由用户方、设计方和投资方共同签署意见。

第三章 需求分析 习题答案

1. 需求分析的描述工具有哪些?

有数据流图、数据字典、判定表、判定树、结构化自然语言、层次方框图、Warnier图、IPO 图和需求描述语言等。

2. 需求分析的基本任务是什么?

准确定义未来系统的目标, 确定为了满足用户的需要系统必须做什么。

3. 怎样建立目标系统的逻辑模型? 要经过哪些步骤?

建立目标系统的逻辑模型的过程也就是数据流图的分解过程。它的导出过程如图:

4. 什么是结构化分析? 它的结构化体现在哪里?

结构化分析: 使用数据流程图、数据字典、结构化英语、判定表和判定树等工具, 来建立一种新的、称为结构化说明书的目标文档—需求规格说明书。

结构化体现在将软件系统抽象为一系列的逻辑加工单元, 各单元之间以数据流发生关联。

5. 软件需求规格说明书由哪些部分组成?

组成包括:

(1) 引言: 编写目的、背景说明、术语定义及参考资料等。

(2) 概述主要功能、约束条件或特殊需求。

(3) 数据流图与数据字典。

(4) 用户接口、硬件接口及软件接口。

(5) 性能需求、属性等。

(6) 其它需求, 如数据库、操作及故障处理等。

6. 为什么数据流图要分层? 画分层的 DFD 要遵循哪些原则?

分层的目的: 便于逐步细化、结构清晰。

画分层的 DFD 要遵循哪些原则:

(1) 父图与子图之间数据要平衡。

(2) 分解的深度和层次达到使加工足够简单、易于理解的基本加工为止。

(3) 区分局部文件和局部外部项(局限于数据流中某一层或某几层的文件和外部项)。

(4) 不要把控制流作为数据流。

(5)忽略琐碎的枝节。

(6)每个数据流要有一个合适的名字，尽量使用现实系统中有具体意义的名字。

7.系统流程图与数据流程图有什么区别？

系统流程图描述系统物理模型的工具，数据流程图描述系统逻辑模型的工具。

系统流程图从系统功能的角度抽象的描述系统的各个部分及其相互之间信息流动的情况。

数据流程图从数据传送和加工的角度抽象的描述信息在系统中的流动和数据处理的工作状况。

8.数据字典包括哪些内容？它的作用是什么？

数据字典是描述数据流图中数据的信息的集合。它对数据流图上每一个成分：数据项、文件(数据结构)、数据流、数据存储、加工和外部项等给以定义和说明；它主要由数据流描述、加工描述和文件描述三部分组成。对用户来讲,数据字典为他们提供了数据的明确定义；对系统分析员来讲,数据字典帮助他们比较容易修改已建立的系统逻辑模型。

9.描述加工逻辑的工具有哪些？

有决策树(又称判定树)、决策表(又称判断表)和结构化语言等。

10.某单位拟开发一个计算机房产管理系统，要求系统具有分房、调房、退房和查询统计等功能。房产科将用户申请表输入系统后，系统首先检查申请表的合法性，对不合法的申请表，系统拒绝接收；对合法的申请表根据类型分别进行处理。

(1)如果是分房申请，则根据申请者的情况(年龄、工龄、职称、职务、家庭人口等)计算其分数，当分数高于阈值分数时，按分数高低将申请单插到分房队列的适当位置。在进行分房时，从空房文件中读出空房信息，如房号、面积、等级、单位面积房租等，把好房优先分给排在分房队列前面的符合该等级房条件的申请者；从空房文件中删掉这个房号的信息，并从分房队列中删掉该申请单，再把此房号的信息和住户信息一起写到住房文件中，输出住房分配单给住户，同时计算房租，并将算出的房租写到房租文件中。

(2)如果是退房申请，则从住房文件和房租文件中删除有关信息，再把此房号的信息写到空房文件中。

(3)如果是调房申请，则根据申请者的情况确定其住房等级，然后在空房文件中查找属于该等级的空房，退掉原住房，再进行与分房类似的处理。

(4) 住户可以向系统查询目前分房的阈值分数, 居住某类房屋的条件, 某房号的单位面积及房租等信息。房产科可以要求系统打印住房情况的统计表, 或更改某类房屋的居住条件、单位面积和房租等。

用数据流图描绘该系统的功能需求; 在数据字典中给出主要的数据流、文件和加工说明。

参考第四章 4.5 应用举例教案之二分析。

第四章 总体设计 习题答案

1. 系统设计包括哪两个阶段?

系统设计包括总体设计与详细设计两个阶段。

2. 总体设计的主要任务是什么?

总体设计的主要任务是完成软件结构的设计, 确定系统的模块及其模块之间的关系。

3. 什么是模块? 模块具有哪几个特征? 总体设计主要考虑什么特征?

模块是数据说明、可执行语句等程序对象的集合, 可以单独命名且可通过名字来访问。

模块具有输入和输出(参数传递)、功能、内部数据结构(局部变量)和程序代码四个特性。

概要设计主要考虑输入、输出(参数传递)和功能两个特性。

4. 什么是模块化? 模块设计的准则?

模块化是按规定的原则将一个大型软件划分为一个个较小的、相对独立但又相关的模块。

模块设计的准则:

(1) 改进软件结构, 提高模块独立性: 在对初步模块进行合并、分解和移动的分析、精化过程中力求提高模块的内聚, 降低藕合。

(2) 模块大小要适中: 大约 50 行语句的代码, 过大的模块应分解以提高理解性和可维护性; 过小的模块, 合并到上级模块中。

(3) 软件结构图的深度、宽度、扇入和扇出要适当。一般模块的调用个数不要超过 5 个。

(4) 尽量降低模块接口的复杂程度;

(5) 设计单入口、单出口的模块。

(6) 模块的作用域应在控制域之内。

5. 变换型数据流由哪几部分组成？

变换型结构由三部分组成:传入路径、变换(加工)中心和传出路径。

6. 变换分析设计的步骤？

(1) 区分传入、传出和变换中心三部分，划分 DFD 图的分界线；

(2) 完成第一级分解:建立初始 SC 图的框架；

(3) 完成第二级分解:分解 SC 图的各个分支；

(4) 对初始结构图按照设计准则进行精化与改进。

7. 事务型数据流由哪几部分组成？

事务型结构由至少一条接受路径、一个事务中心与若干条动作路径组成。

8. 事务分析设计的步骤？

(1) 在 DFD 图中确定事务中心、接收部分(包含全部接收路径)和发送部分(包含全部动作路径)；

(2) 画出 SC 图框架,把 DFD 图的三部分分?quot;映射 为事务控制模块,接收模块和动作发送模块.一般得到 SC 图的顶层和第一层(如果第一层简单可以并入顶层)；

(3) 分解和细化接收分支和动作分支,完成初始的 SC 图；

(4) 对初始结构图按照设计准则进行精化与改进。

9. 比较层次方框图与结构图是的异同？

(1) 层次方框图描绘数据的层次结构,结构图描绘的是软件结构。

(2) 二者都采用多层次矩形框树形结构。层次方框图的顶层矩形框代表完整的数据结构,下面各层矩形框依次代表上个框数据的子集;结构图 是在层次图的每一个方框内注明模块的名字或主要功能,方框之间的直线表示模块的调用关系,用带注解的箭头表示模块调用过程中传递的信息。

10. 欲开发一个银行的活期存取款业务的处理系统: 储户将填好的存/取款单和存折交给银行工作人员, 然后由系统作以下处理;

(1)业务分类处理: 系统首先根据储户所填的存/取款单, 确定本次业务的性质, 并将存/取款单和存折交下一步处理;

(2)存款处理: 系统将存款单上的存款金额分别记录在存折和帐目文件中, 并将现金存入现金库; 最后将存折还给储户;

(3)取款处理: 系统将取款单上的取款金额分别记录在存折和帐目文件中, 并从现金库提取现金; 最后将现金和存折还给储户。

绘制该系统的数据流图和软件结构图。

顶层

一层

二层

软件结构图

第五章 习题参考答案

一.回答问题

1.详细设计的目的?

为软件结构图(SC图或HC图)中的每一个模块确定采用的算法和块内数据结构,用某种选定的表达工具给出清晰的描述.

2.详细设计的主要任务?

编写软件的“详细设计说明书”软件人员要完成的工作:

(1) 为每一个模块确定采用的算法,选择某种适当的工具表达算法的过程,写出模块的详细过程描述.

(2) 确定每一模块使用的数据结构.

(3) 确定模块结构的细节,包括对系统外部的接口和用户界面,对系统内部其它模块的接口,以及关于模块输入数据、输出数据及局部数据的全部细节.

(4) 为每一个模块设计出一组测试用例,以便在编码阶段对模块代码(即程序)进行预定的测试.

3.结构化程序设计的基本原则?

在详细设计中所有模块都使用单入口、单出口的顺序、选择、循环三种基本控制结构.

4.比较面向数据流和面向数据结构两类设计方法的异同?

相同点:

(1) 遵守结构程序设计“由顶向下”逐步细化的原则,并以其为共同的基础;

(2) 均服从“程序结构必须适应问题结构”的基本原则,各自拥有从问题结构(包括数据结构)导出程序结构的一组映射规则。

不同点:

在分析阶段用 DFD 表示软件的逻辑模型，在设计阶段按数据流类型，将数据流图转换为软件结构。面向数据结构的设计以数据结构为基础，从问题的数据结构出发导出它的程序结构。

(2) 面向数据流的设计的最终目标是软件的最初 SC 图，面向数据结构的设计的最终目标是程序的过程性描述。

5. 比较 Jackson 方法和 LCP 方法的异同？

Jackson 与 LCP 设计方法都是以数据结构为出发点，以程序的过程描述为最终目标，设计步骤基本相似。它们的主要差别是：

(1) 使用不同的表达工具，其中 LCP 方法中的表达工具 Warnier 图

比 Jackson 设计方法中的表达工具 Jackson 图有更大的通用性；

(2) Jackson 方法的步骤和指导原则有一定的灵活性，而 LCP 设计方法则更加严密。

6. 详细设计的描述工具应具备什么功能？

无论哪类描述工具不仅要具有描述设计过程，如控制流程、处理功能、数据组织及其它方面的细节的能力，而且在编码阶段能够直接将它翻译为用程序设计语言书写的源程序。

二. 给出一组数从小到大的排序算法，分别用下列工具描述其详细过程：

(1) (1)流程图；(2)N S 图；(3)PAD 图；(4)PDL 语言。

PDL 语言略

三. 根据后面给出“卡片的分类及统计”项目说明，完成下列工作：

(1) 用 Jackson 图表示输入与输出数据结构，找出它们之间的对应单元；

(2) 画出用 Jackson 图表示的程序结构；

(3) 出程序所需用的操作，并加到上一步画出的程序结构图上；

(4) 用 Jackson 伪代码写出程序的过程表示。

输入卡片分析程序。一叠输入卡片按内容分为 K1、K2、K3 三类，卡片的排列规则是：以 K1 卡始，以 K2 卡终。起始卡之前属前置部分，不含 K1 卡；起始卡与终了卡之间属分批部分，不含 K2 卡。如图：

Jackson 方法编一程序,要求依次完成下列分析工作:

- 统计起始卡前卡片的张数,存入 A;
- 打印起始卡的内容;
- 统计起始卡后出现的 K1 卡和 K3 卡总批数,存入 B;
- 统计起始卡后出现的 K1 卡的张数,存入 C;
- 统计起始卡后出现的 K3 卡的批数,存入 D;
- 打印终了卡的内容;
- 打印 A、B、C、D 四个统计值。

程序的过程表示及伪代码略

第六章 习题参考答案

1. 编码的任务?

使用选定的程序设计语言,把模块的过程性描述翻译为用语言书写的源程序(源代码)。

2. 对源程序基本要求?

源程序要求:正确可靠、简明清晰、效率高。

- (1) 源程序的正确性是对程序质量的最基本要求;
- (2) 源程序的简明清晰,便于验证源代码和模块规格说明的一致性,容易进行测试和维护;
- (3) 对于大多数模块,编码时应该把简明清晰放在第一位;
- (4) 除了编码阶段产生源代码外,在测试阶段也需要编写一些测试程序,用于对软件的测试。

3. 程序设计语言的特点?

- (1) 名字说明:程序中使用对象的名字,能为编译程序所检查和识别;
- (2) 类型说明:定义对象的类型,确定该对象的使用方式;
- (3) 初始化:为变量提供适当的初始值或由系统给变量赋一特殊的表明未初始化的值;
- (4) 对象的局部性:程序中真正需要的那部分才能访问的对象;
- (5) 程序模块:控制程序对象的名字;

FOR 语句、WHILE-DO 语句、REPEAT-UNTIL 语句等；

(7) 分支控制结构：如 IF 语句、CASE 语句等；

(8) 异常处理：为程序运行过程中发生的错误和意外事件提供检测和处理上的帮助；

(9) 独立编译：能分别编译各个程序单元。

4. 选择程序设计语言需要考虑的因素？

(1) 选择用户熟悉、便于用户维护的语言。

(2) 选择目标系统的环境中可以提供的编译程序所能选用的语言。

(3) 选择可以得到的软件工具,能支持程序开发中可以利用的语言。

(4) 根据工程规模的大小、目标系统应用范围,如实时应用选择 Ada 语言或汇编语言,系统软件开发选择 语言或汇编语言,软件开发中若含有大量数据操作则选择 SQL、dBASE 等数据库语言等。

(5) 选择程序员熟悉的语言。

(6) 选择标准化程度高、程序可移植性好的语言。

(7) 根据算法与计算的复杂性、数据结构的复杂性选择。如对于系统程序和结构复杂的应用程序,选择支持数组、记录(或结构)与指针动态数据结构的 Pascal 语言或 C 语言。

(8) 根据实时要求系统需要的响应速度和效率选择相应的语言。

5. 编码风格的指导原则。

(1) 源程序：包括适当的标识符、适当的注解、程序清单的合理布局与清晰；

(2) 数据说明：数据结构或数据类型的说明次序标准化；变量名称尽量有意义；对复杂的数据结构在注解中要说明在程序设计中实现这个数据结构的方法。

(3) 语句的构造简单明了：不要为节省空间将多个语句写在同一行；尽量避免复杂的条件及非”条件的测试；避免大量使用循环嵌套和条件嵌套；括号的使用是为了使逻辑表达式和算术表达式的运算顺序清晰直观。

(4) 效率：考虑程序运行的时间存储器效率、输入/输出的效率；在处理程序正确性、清晰与效率之间的关系时先求程序正确后求快；先求清楚后求快；保持程序简单以求快；书写清楚,不为”效率”牺牲清晰。

6. 第四代语言(4GL)应具备哪些的特征？

(1) 具有很强的数据管理能力,能对数据库进行有效的存取、查询和其它有关操作；

(2) 能提供一组高效的、非过程化的命令,组成语言的基本语句,编程时用户只需用这些命令说明”做什么”,不必描述实现的细节；

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/348117107130006124>