《软件工程导论》课后习题答案

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合成一个整体,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？它们的开发方法有可特点？

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

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、定量化;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

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.

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?

判定树、判断表和结构化语言等。

第四章 总体设计

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。

第五章 详细设计

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:

(1) 面向数据流的设计以数据流图为基础，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终 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.

第六章 编码

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；

(6） 循环控制结构:如 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 C 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；

（3) 能满足多功能、一体化的要求.为此，语言中除必须含有控制程序逻辑与实现数据库操作的语句外，还应包括生成与处理报表、表格、图形,以及实现数据运算和分析统计功能的各种语句，共同构成一个一体化的语言,以适应多种应用开发的需要。

第七章　软件测试

1. 软件测试的基本任务?

软件测试是按照特定的规则，发现软件错误的过程；好的测试方案是尽可能发现迄今尚未发现错误的测试;成功的测试方案是发现迄今尚未发现错误的测试;

2. 测试与调试的主要区别?

（1） 测试从一个侧面证明程序员的失败；调试证明程序员的正确;

（2） 测试从已知条件开始,使用预先定义的程序，且有预知的结果，不可预见的仅是程序是否通过测试；调试从不可知内部条件开始，除统计性调试外，结果是不可预见的;

（3） 测试有计划并且要进行测试设计；调试不受时间约束；

（4） 测试是发现错误、改正错误、重新测试的过程;调试是一个推理的过程;

（5） 测试执行是有规程的；调试执行要求程序员进行必要的推理；

(6） 测试由独立的测试组在不了解软件设计的件下完成；调试由了解详细设计的程序员完成;

(7) 大多数测试的执行和设计可由工具支持；调试用的工具主要是调试器。

3。人工复审的方式和作用?

人工复审的方式：代码会审、走查和排练和办公桌检查;

人工复审的作用：检查程序的静态错误.

4。什么是黑盒测试?黑盒测试主要采用的技术有哪些?

黑盒测试：也称为功能测试,它着眼于程序的外部特征，而不考虑程序的内部逻辑结构。测试者把被测程序看成一个黑盒，不用关心程序的内部结构。黑盒测试是在程序接口处进行测试，它只检查程序功能是否能按照规格说明书的规定正常使用，程序是否能适当地接收输入数据产生正确的输出信息，并且保持外部信息（如数据库或文件)的完整性。

黑盒测试主要采用的技术有:等价分类法、边沿值分析法、错误推测法和因果图等技术。

5.什么是白盒测试？白盒测试主要采用的技术有哪些?

测试者了解被测程序的内部结构和处理过程,对程序的所有逻辑路径进行测试，在不同点检查程序状态,确定实际状态与预期状态是否一致。

白盒测试主要采用的技术有:路径测试技术和事务处理流程技术，对包含有大量逻辑判断或条件组合的程序采用基于逻辑的测试技术.

6. 路径测试技术中几种主要覆盖的含义？举例说明?

语句覆盖：至少执行程序中所有语句一次。

判定覆盖:使被测程序中的每一个分支至少执行一次。故也称为分支覆盖。

条件覆盖:执行所有可能的穿过程序的控制路流程。

条件组合测试:设计足够的测试用例，使每个判定中的所有可能条件取值组合至少执行一次。

7。 等价分类法的测试技术采用的一般方法?举例说明？

（1） 为每个等价类编号;

（2） 设计一个新的测试方案，以尽可能多的覆盖尚未被覆盖的有效等价类，重复这一步骤，直到所有有效等价类被覆盖为止.

（3） 设计一个新的测试方案,使它覆盖一个尚未被覆盖的无效等价类, 重复这一步骤,

直到所有无效等价类被覆盖为止。

8. 软件测试的一般步骤？

单元测试、子系统测试、系统测试、验收测试、平行测试。

9. 比较集成试的两种方式的优劣?

非渐增式测试方式：分别测试模块，再把所有模块按设计要求放在一起组成所要的程序。该方法编写测试软件工作量大，模块间的接口错误发现得晚,错误定位较难诊断,总体测试有的错误容易漏掉，测试时间相对较少，可以并行测试所有模块,能充分利用人力,加快工程进度。。

渐增式测试方式：把下一个要测试的模块，同已经测试好的那些模块结合起来进行测试。该方法利用已测试过的模块作测试软件，开销小,较早发现模块间的接口错误,错误定位往往和最近入的模块相关，对已测试好的模块可在新加入模块的条件下受到新的检验，测试更彻底，需要较多的测试时间,不能并行测试。

总的来说，渐增式测试方法比较好。

10.软件测试的策略？

(1)　在任何情况下都应使用边界值分析的方法。

(2)　必要时用等价类划分法补充测试方案.

(3)　必要时再用错误推测法补充测试方案。

(4)　对照程序逻辑，检查已设计出的测试方案。

(5)　根据对程序可靠性的要求采用不同的逻辑覆盖标准，再补充一些测试方案.

第八章 软件维护

1. 为什么说软件的维护是不可避免的？

因为软件的开发过程中，一般很难检测到所有的错误,其次软件在应用过程中需要随用户新的要求或运行环境的变化而进行软件的修改或完成功能的增删等,为了提高软件的应用水平和使用寿命，软件的维护是不可避免的。

2。 软件的维护一般分为哪几类？

改正性维护:满足用户对已开发产品的性能与运行环境不断提高的要求，进而达到延长软件寿命的目的.

适应性维护：对程序使用期间发现的程序错误进行诊断和改正的过程,配合变化了的环境进行修改软件的活动;

完善性维护：满足用户在使用过程中提出增加新的功能或修改已有功能的建议而进行的工作;

预防性维护：为了改善未来的可维护性或可靠性而修改软件的工作.

3。 影响软件维护的因素有哪些?

开发方法：采用模块化详细设计文档有助于理解软件的结构、界面功能和内部流程；开发过程中严格而科学的管理规划及清晰可靠的文档资料对发生错误后的理解与纠错是至关重要的；开发过程中模块的独立程度越高，对软件修改越容易，对软件的改进和移植越方便.

开发条件：软件开发及维护人员的水平决定了软件开发的质量和维护的效率；开发过程中使用标准的程序设计语言和标准的操作系统接口，可以大大提高软件的可维护性；在测试

过程中用例的有效性,可极大地减少软件存在的错误；其次使用规范化的文档资料可为维护提供更好的依据。

4。 软件维护困难主要表现在什么方面？

（1） 一般来讲，维护人员对开发人员写的程序及文档，理解都比较困难，对维护工作不会喜欢;

（2） 维护持续时间都很长，在开发人员不在现场的轻快下，维护软件通常是很困难的；

(3） 绝大多数软件在设计时对将来的软件修改都没有考虑或考虑不多，尤其未能在设计中强调并认真解决好模块的独立性,使软件的修改既困难又易发生差错。

5。 决定软件可维护性的因素？

（1) 软件的可理解性、可测试性、可修改性；

（2) 文档描述符合要求、用户文档简洁明确、系统文档完整并且标准。

6. 软件价格应该计入维护成本吗？为什么？

在软件的生命周期中，软件维护的工作量非常大,不同应用领域的维护成本差别也很大。一般大型软件的维护成本远远高于开发成本若干倍。因此软件价格中应该计入维护成本。

第九章 软件工程管理

1。 软件工程管理的内容？

(1） 费用管理: 对软件开发进行成本核算，使软件生产按照商品生产的规律办事.包括:以简单、科学方法估算软件开发费用，作为签定开发合同的根据；管理开发费用的有效使用,即用经济手段来保证产品如期按质完成。

(2) 质量管理: 按项目的质量保证计划,确保各个开发阶段的开发和维护工作全部按软件工程的规范进行,保证软件产品的质量。

(3） 配置管理:通过对于程序、文档和数据的各种版本所进行的管理,保证资料的完整性与一致性。

（4) 项目管理：制定《项目实施计划》，按照计划的内容组织和实施软件的工程化生产。最终目标是以合理的费用和进度,圆满完成计划所规定的软件项目。

2。 软件项目有哪些特点？

(1） 软件项目与其他任何产业项目不同,它是算法、思想、概念、组织、流程、效率、优化等的融合体；

(2) 开发软件项目产品，在多数情况下,用户给不出明确的想法和要求。

（3） 在开发过程中，程序及其相关的文档资料常常需要修改,在修改过程中又可能带来新的问题,且这些问题要在很久以后才会发现.

(4) 在研制开发过程中，文档资料是不可缺少的，但工作量又是巨大的，往往也是人们不愿去作的。

(5) 参加软件项目的工作人员,要求具有一定的业务水平和实际工作经验，而很难完全避免的人员流动，对工作的影响是很大的。离开的人员不仅带走了重要的信息，而且带走了工作经验.

3. 软件成本估算的一般方法？

自顶向下估计：首先估算出项目总的开发成本，然后在项目内部进行成本分配。由少数专家参与，依靠他们过去的经验，将要开发的软件与过去开发过的软件进行”类比”，以估计新的软件开发所需要的工作量和成本。

自底向上估计：将开发任务分成若干子任务,子任务又分成子子任务,直到每一个单元内容足够明确为止；把各个任务单元的成本估计出来，汇合成项目的总成本.该方法得到的结果比较接近实际。

4. 为什么在软件开发中，不能用简单增加人员的方法来缩短开发时间?

大量软件开发实践说明:向一个已经延迟的项目追加开发人员,可能使它完成得更晚。因为当开发人员以算术级数增长时，而人员之间的通信将以几何级数增长,往往"得不偿失"。

5。 影响软件质量的主要因素有哪些?

(1) 产品运行:正确性、风险性、效率、完整性、健壮性和可用性;

(2） 产品修改:可理解性、可维护性、灵活性、可测试性;

（3) 产品转移:可移植性、可重用性和互运行性.