

例题1



4.2 试题精解

例题1（ 2006年5月试题15 ）

在软件项目管理中可以使用各种图形工具来辅助决策，下面对Gantt图的描述中，不正确的是_____。

- A.Gantt图表现各个活动的持续时间
- B.Gantt图表现了各个活动的起始时间
- C.Gantt图反映了各个活动之间的依赖关系
- D.Gantt图表现了完成各个活动的进度

试题分析

Gantt图也就是甘特图（参看图4-6），它使用水平线段表示任务的工作阶段，线段的起点和终点分别对应着任务的开工时间和完成时间，线段的长度表示完成任务所需的时间，这样能直观地表现出任务与作任务之间的依赖关系。所以此题应选C。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题2

例题2（ 2006年5月试题16 ）

耦合度描述了_____。

- A.模块内各种元素结合的程度
- B.模块内多个功能之间的接口
- C.模块之间公共数据的数量
- D.模块之间相互关联的程度

试题分析

此题为概念题。耦合描述的是模块之间联系的紧密程度。
模块的耦合性类型通常分为7种，根据耦合度从低到高排序如表4-2所示。

表4-2 7种耦合的描述对比

耦合类型	描 述
非直接耦合	没有直接联系，互不依赖对方
数据耦合	借助参数表传递简单数据
标记耦合	一个数据结构的一部分借助于模块接口被传递
控制耦合	模块间传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	与软件以外的环境有关
公共耦合	多个模块引用同一个全局数据区
内容耦合	一个模块访问另一个模块的内部数据 一个模块不通过正常入口转到另一模块的内部 两个模块有一部分程序代码重叠 一个模块有多个入口

试题答案

D

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题3

例题3（ 2006年5月试题17 ）

- 内聚是一种指标，表示一个模块_____。
- A.代码优化的程度 B.代码功能的集中程度
- C.完成任务的及时程度 D.为了与其他模块连接所要完成的工作量

试题分析

此题也属概念题。内聚是指模块内部各元素之间联系的紧密程度，也就是代码功能的集中程度。

模块的内聚类型通常可以分为7种，根据内聚度从高到低排序如表4-3所示。

表4-3 7种内聚的描述对比

内聚类型	描 述
功能内聚	完成一个单一功能，各个部分协同工作，缺一不可
顺序内聚	处理元素相关，而且必须顺序执行
通信内聚	所有处理元素集中在一个数据结构的区域上
过程内聚	处理元素相关，而且必须按特定的次序执行
瞬时内聚	所包含的任务必须在同一时间间隔内执行（如初始化模块）
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚	完成一组没有关系或松散关系的任务

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题4

例题4（2006年5月试题18）

在软件项目开发过程中，评估软件项目风险时，_____与风险无关。

- A.高级管理人员是否正式承诺支持该项目
- B.开发人员和用户是否充分理解系统的需求
- C.最终用户是否同意部署已开发的系统
- D.开发需要的资金是否能按时到位

试题分析

此题要求判断某种情况是否属于项目风险。这要求考生对项目风险的概念非常熟悉。

项目风险是指项目的执行期间的风险，项目的一次性使其不确定性比其他经济活动大得多，因而项目风险的可测性就差得多。在进行重复性的生产和经营活动时，可以根据历史资料和同行业的经验数据预测大多数风险，而项目各有特点，每个项目都有各自不同的具体问题，风险的预测要困难得多。

一般项目风险的风险因素主要有两类，一类是物质风险因素，俗称硬件风险因素；另一类是人为风险因素，人为风险因素是指与人们的心理和行为有关的一种无形因素。备选答案中的"高级管理人员是否正式承诺支持该项目"和"开发人员和用户是否充分理解系统的需求"属人为风险因素，而"开发需要的资金是否能按时到位"属于物质风险因素，至于"最终用户是否同意部署已开发的系统"已与软件开发过程毫无关系，不属于项目风险。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

例题5

例题5（2006年5月试题26）

最主要的软件质量度量指标有正确性、可维护性、完整性和可用性，下面关于这些质量指标的描述中正确的是_____。

- A.软件维护的工作量比开发阶段的工作量小
- B.正确性用每千行代码的故障（fault）数来度量
- C.软件完整性是指软件功能与需求符合的程度
- D.可用性与用户的操作效率和用户对软件的主观评价有关

试题分析

本题考查软件的质量度量指标。

软件维护的工作量比开发阶段的工作量大，通常的估计是，开发阶段的工作量占软件生命期整

个工作量的40%,而维护阶段的工作量则占60%,甚至更多。软件的正确性是指软件完成所需功能的程度,尽管这种程序与每千行代码的故障数有关,但不完全等同。软件完整性是指软件在安全方面攻击的能力。软件可用性用来度量软件的"用户友好性".可以从4个方面来测量可用性。

学会操作软件所需的体力/智力;

对系统的使用达到中等效率所需的时间;

当系统由一个中等效率的人使用时测量到的生产率增长值;

用户对系统的主观评价。

试题答案

D

版权方授权希赛网发布,侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题6

例题6 (2006年5月试题27)

某工程计划图如图4-1所示,弧上的标记为作业编码及其需要的完成时间(天),作业E最迟应在第____天开始。

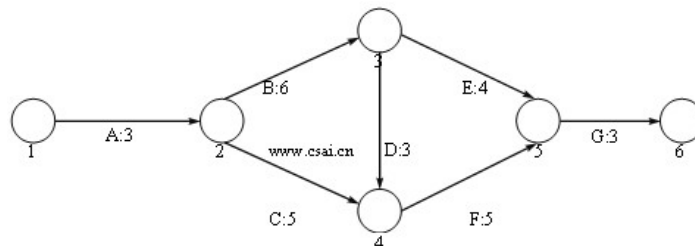


图4-1 工程计划图

A.7 B.9 C.12 D.13

试题分析

本题实际考查内容为关键路径相关知识。

要求作业E最迟开始时间,首先得求出关键路径。关键路径为:1->2->3->4->5->6,路径长度为20.再从终点向起点反推,G的最迟开始时间为:20-3=17,E的最迟开始时间为:17-4=13,所以作业E的最迟开始时间为第13天。

试题答案

D

版权方授权希赛网发布,侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题7

例题7（2006年5月试题28）

正式的技术评审FTR（Formal Technical Review）是软件工程师组织的软件质量保证活动，下面关于FTR指导原则中不正确的是_____。

- A. 评审产品，而不是评审生产者的能力
- B. 要有严格的评审计划，并遵守日程安排
- C. 对评审中出现的问题要充分讨论，以求彻底解决
- D. 限制参与者人数，并要求评审会之前做好准备

试题分析

本题考查软件评审的基本概念。

软件评审是评审软件产品，不要涉及对软件生产者能力的评价；评审前要制定严格的评审计划，并严格遵守预计的日程安排；对评审中出现的问题要记录在案，不要过多地讨论解决方案，把问题留给软件生产者来解决；要限制参与者人数，并要求参加评审的人员在评审会之前仔细阅读文档，做好充分的准备。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第4章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题8

例题8（2006年5月试题29）

两个小组独立地测试同一个程序，第一组发现25个错误，第二组发现30个错误，在两个小组发现的错误中有15个是共同的，那么可以估计程序中的错误总数是_____个。

- A.25 B.30 C.50 D.60

试题分析

本题考查软件测试策略的基本概念。

对于第一小组：发现了第二组发现的错误的 $15/30=0.5=50\%$ 。

对于第二小组：发现了第一组发现的错误的 $15/25=0.6=60\%$ 。

根据第一组发现的错误数和第一组的效率得到： $25/50\%=50$ 。

根据第二组发现的错误数和第二组的效率得到： $30/60\%=50$ 。

由于两个小组是独立进行测试的，所以可以估计：程序中的错误总数为50个。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

例题9

例题9（2006年5月试题30）

下面有关测试的说法正确的是_____。

- A.测试人员应该在软件开发结束后开始介入
- B.测试主要是软件开发人员的工作
- C.要根据软件详细设计中设计的各种合理数据设计测试用例
- D.严格按照测试计划进行，避免测试的随意性

试题分析

此题考查的是测试的一些基础概念。测试这种活动，是一个贯穿于软件开发全过程的活动，并不是先开发完产品，再进行测试。所以A选项所述的"测试人员应该在软件开发结束后开始介入"这种说法是不正确的。测试这项工作应由专门的测试人员进行，因为如果软件开发人员测试自己的程序往往难以发现错误，或是潜意识不想以"破坏性"的方式对程序进行测试，导致测试效果不佳，所以B选项的说法是错误的。测试工作的进行是有严格控制的，什么阶段做什么样的工作都有明确规定，在软件的详细设计阶段应是单元测试实施的时期，所以C选项的说法是错误的。D答案正确。

试题答案

D

[版权方授权希赛网发布，侵权必究](#)

例题10

例题10（2006年5月试题31）

在绘制数据流程图时，应遵循父图与子图平衡的原则，所谓平衡是指_____。

- A.父图和子图都不得改变数据流的性质
- B.子图不改变父图数据流的一致性
- C.父图的输入/输出数据流与子图的输入/输出数据流一致
- D.子图的输出数据流完全由父图的输入数据流确定

试题分析

在绘制数据流程图时，应遵循父图与子图平衡的原则，这个平衡包括两方面的内容：一方面是：父图的输入/输出数据流与子图的输入/输出数据流一致，如果加工A有输入/输出数据流，在其父图中也应该有对应的数据流；另一方面是：对于一个加工，应有输入数据流和输出数据流，两者缺一不可。

例题11

例题11 (2006年5月试题32)

图4-2中的程序由A、B、C、D、E 5个模块组成，下表中描述了这些模块之间的接口，每一个接口有一个编号。此外，模块A、D和E都要引用一个专用数据区，那么A和E之间耦合关系是_____。

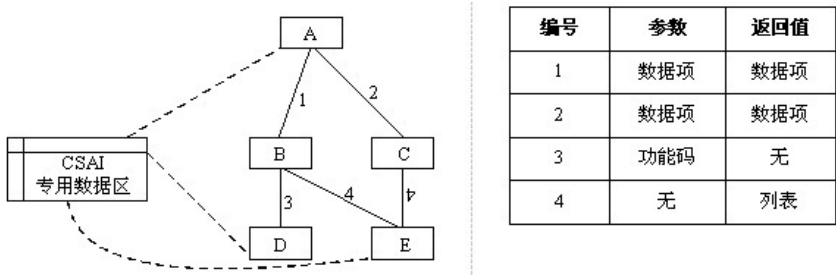


图4-2 模块联系图

A.公共耦合 B.数据耦合 C.内容耦合 D.无耦合

试题分析

模块的耦合性类型通常分为7种，根据耦合度从低到高排序如表4-7所示。我们不难发现，A与E之间的关系为"多个模块引用同一个全局数据区",所以为公共耦合。此题答案为：A.

试题答案

例题12

例题12 (2006年11月试题15, 16)

常见的软件开发模型有瀑布模型、演化模型、螺旋模型、喷泉模型等。其中

(15) 模型适用于需求明确或很少变更的项目，(16) 模型主要用来描述面向对象的软件开发过程。

(15) A.瀑布模型B.演化模型 C.螺旋模型D.喷泉模型

(16) A.瀑布模型B.演化模型 C.螺旋模型D.喷泉模型

试题分析

此题考的是常见软件开发模型的特点，下面是对这几种开发模型的特点总结。

瀑布模型：严格遵循软件生命周期各阶段的固定顺序，一个阶段完成再进入另一个阶段。它的优点是：可以使过程比较规范化，有利于评审；缺点是：过于理想，缺乏灵活性，容易产生需求偏差。

演化模型：也是一种原型化开发，但与快速原型不同的是，快速原型模型在获得真实需求时，就将抛弃原型。而演化模型则不然，它将从初始的模型中逐渐演化为最终软件产品，是一种"渐进式"原型法。

螺旋模型：结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制订计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

喷泉模型：主要用于描述面向对象的开发过程，最核心的特点是迭代。所有的开发活动没有明显的边界，允许各种开发活动交叉进行。

试题答案

(15) A (16) D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第4章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题13

例题13 (2006年11月试题18)

软件项目开发成本的估算依据，通常是开发成本估算模型。常用的模型主要有：

- ① IBM模型 ② Putnam模型 ③ 基本COCOMO模型
④ 中级COCOMO模型 ⑤ 高级COCOMO模型

其中 均是静态单变量模型。

A.①② B.②④⑤ C.①③ D.③④⑤

试题分析

此题考查成本模型基本知识，下面是常用的成本模型基本特性。

IBM模型：是在60多个项目的基础上进行统计得出的静态模型。

Putnam (普特南)模型：是一种动态多变量模型，它通过建立一个"资源需求曲线模型"来导出一系列等式，模型化资源特性。

COCOMO模型：是最有代表性的方法。在该模型中使用了源指令条数 (DSI)、开发工作量 (MM)、开发进度 (TDEV) 三个基本量，它将项目分为组织型 (相对较小、较简单的项目)、嵌入式 (软硬件限制较多的项目)、半独立型 (介于两者之间，规模和复杂性中等以上)。它包括基本 (静态单变量模型)、中间 (静态多变量模型)、详细三种不同的模型。

根据这些特性，我们可以得到答案C。

试题答案

C

例题14

例题14 (2006年11月试题26)

统一过程（UP）的基本特征是"用例驱动、以架构为中心的和受控的迭代式增量开发".UP将一个周期的开发过程划分为4个阶段，其中 _____ 的提交结果包含了系统架构。

- A.先启阶段 B.精化阶段 C.构建阶段 D.提交阶段

试题分析

统一过程4个阶段的核心任务，以及需要提交的文档和模型分别如下。

（1）先启阶段

① 核心任务

明确地说明项目规模。这涉及了解环境及最重要的需求和约束，以便于可以得出最终产品的验收标准。

计划和准备商业理由。评估风险管理、人员配备、项目计划和成本/进度/收益率折中的备选方案。

综合考虑备选构架，评估设计和自制/外购/复用方面的折中，从而估算出成本、进度和资源。此处的目标在于通过对一些概念的证实来证明可行性。该证明可采用可模拟需求的模型形式或用于探索被认为高风险区域的初始原型。先启阶段的原型设计工作应该限制在确信解决方案可行就可以了。该解决方案在精化和构建阶段实现。

准备项目的环境，评估项目和组织，选择工具，决定流程中要改进的部分。

② 需要提交的文档和模型（见表4-4）

表4-4 先启阶段需要提交的文档和模型

核心文档及模型	里程碑状态
前景	已经对核心项目的需求、关键功能和主要约束进行了记录
商业理由	已经确定并得到了批准
风险列表	已经确定了最初的项目风险
软件开发计划	已经确定了最初阶段及其持续时间和目标。软件开发计划中的资源估算（特别是时间、人员和开发环境成本）必须与商业理由一致。资源估算可以涵盖整个项目直到交付所需的资源，也可以只包括进行精化阶段所需的资源。此时，整个项目所需的资源估算应该看作是大致的“粗略估计”。该估算在每个阶段和每次迭代中都会更新，并且随着每次迭代变得更加准确。根据项目的需要，可能在某种条件下完成了一个或多个附带的“计划”工件。此外，附带的“指南”工件通常也至少完成了“草稿”。
迭代计划	第一个精化迭代的迭代计划已经完成并经过了复审
软件验收计划	完成复审并确定了基线；随着其他需求的发现，将对其在随后的迭代中进行改进
项目专用模板	已使用文档模板制作了文档工件
用例建模指南	确定了基线
工具	选择了支持项目的所有工具。安装了对先启阶段的工作必要的工具
词汇表	已经定义了重要的术语；完成了词汇表的复审
用例模型	已经确定了重要的主角和用例，只为最关键的用例简要说明了事件流
领域模型	已经对系统中使用的核心概念进行了记录和复审。在核心概念之间存在特定关系的情况下，已用作对词汇表的补充
原型	概念原型的一个或多个证据，以支持前景和商业理由，解决非常具体的风险

（2）精化阶段

① 核心任务

快速确定构架，确认构架并为构架建立基线。

根据此阶段获得的新信息改进前景，对推动构架和计划决策的最关键用例建立可靠的了解。

为构建阶段创建详细的迭代计划并为其建立基线。

改进开发案例，定位开发环境，包括流程和支持构建团队所需的工具和自动化支持。

改进构架并选择构件。评估潜在构件，充分了解自制/外购/复用决策，以便有把握地确定构建阶段的成本和进度。集成了所选构架构件，并按主要场景进行了评估。通过这些活动得到的经验有可能导致重新设计构架、考虑替代设计或重新考虑需求。

② 需要提交的文档和模型（见表4-5）

表4-5 精化阶段需要提交的文档和模型

核心文档及模型	里程碑状态
原型	已经创建了一个或多个可执行构架原型，以探索关键功能和构架上的重要场景
风险列表	已经进行了更新和复审。新的风险可能是构架方面的，主要与处理非功能性需求有关
项目专用模板	已使用文档模板制作了文档工件
工具	已经安装了用于支持精化阶段工作的工具
软件构架文档	编写完成并确定了基线，如果系统是分布式的或必须处理并行问题，则包括构架上重要用例的详细说明（用例视图）、关键机制和设计元素的标识（逻辑视图），以及（部署模型的）进程视图和部署视图的定义
设计模型（和所有组成部分）	制作完成并确定了基线。已经定义了构架方面重要场景的用例实现，并将所需行为分配给了适当的设计元素。已经确定了构件并充分理解了自制/外购/复用决策，以便有把握地确定构建阶段的成本和进度。集成了所选构架构件，并按主要场景进行了评估。通过这些活动得到的经验有可能导致重新设计构架、考虑替代设计或重新考虑需求
数据模型	制作完成并确定了基线。已经确定并复审了主要的数据模型元素（如重要实体、关系和表）
实施模型（以及所有组成工件，包括构件）	已经创建了最初结构，确定了主要构件并设计了原型
前景	已经根据此阶段获得的新信息进行了改进，对推动构架和计划决策的最关键用例建立了可靠的了解
软件开发计划	已经进行了更新和扩展，以便涵盖构建阶段和产品化阶段
指南，如设计指南和编程指南	使用指南对工作进行了支持
迭代计划	已经完成并复审了构建阶段的迭代计划
用例模型	用例模型（大约完成 80%）——已经在用例模型调查中确定了所有用例、确定了所有主角并编写了大部分用例说明（需求分析）
补充规约	已经对包括非功能性需求在内的补充需求进行了记录和复审
可选	里程碑状态
商业理由	如果构架调查不涵盖变更基本项目假设的问题，则已经对商业理由进行了更新
分析模型	可能作为正式工件进行了开发；进行了经常但不正式的维护，正演进为设计模型的早期版本
培训材料	用户手册与其他培训材料，根据用例进行了初步起草。如果系统具有复杂的用户界面，可能需要培训材料

（3）构建阶段

① 核心任务

资源管理、控制和流程优化。

完成构件开发并根据已定义的评估标准进行测试。

根据前景的验收标准对产品发布版进行评估。

② 需要提交的文档和模型（见表4-6）

表4-6 构建阶段需要提交的文档和模型

核心文档及模型	里程碑状态
“系统”	可执行系统本身随时可以进行“Beta”测试
部署计划	已开发最初版本，进行了复审并建立了基线
实施模型	对在精化阶段创建的模型进行了扩展；构建阶段末期完成所有构件的创建
测试模型	为验证构建阶段所创建的可执行发布版而设计并开发的测试
培训材料	用户手册与其他培训材料，根据用例进行了初步起草。如果系统具有复杂的用户界面，可能需要培训材料
迭代计划	已经完成并复审了产品化阶段的迭代计划
设计模型	已经用新设计元素进行了更新，这些设计元素是在完成所有需求期间确定的
项目专用模板	已使用文档模板制作了文档工件
工具	已经安装了用于支持构建阶段工作的工具
数据模型	已经用支持持续实施所需的所有元素（如表、索引、对象关系型映射等）进行了更新
可选	里程碑状态
补充规约	已经用构建阶段发现的新需求（如果有）进行了更新
用例模型	已经用构建阶段发现的新用例（如果有）进行了更新

（4）产品化阶段（提交阶段）

① 核心任务

执行部署计划。

对最终用户支持材料定稿。

在开发现场测试可交付产品。

制作产品发布版。

获得用户反馈。

基于反馈调整产品。

使最终用户可以使用产品。

② 需要提交的文档和模型（见表4-7）

表4-7 提交阶段需要提交的文档和模型

核心文档及模型	里程碑状态
产品工作版本	已按照产品需求完成，客户应该可以使用最终产品
发布说明	完成
安装产品与模型	完成
培训材料	完成，以确保客户自己可以使用和维护产品
最终用户支持材料	完成，以确保客户自己可以使用和维护产品
可选	里程碑状态
测试模型	在客户想要进行现场测试的情况下，可以提供测试模型

从以上说明可以看出：在精化阶段要求创建一个或多个可执行构架原型，所以正确答案为B.

试题答案

B

版权方授权希赛网发布，侵权必究

上一节 本书简介 下一节

例题15

例题15（2006年11月试题27）

某软件在应用初期运行在Windows NT环境中，现因某种原因，该软件需要在UNIX环境中运行，而且必须完成相同的功能。为适应这个要求，软件本身需要进行修改，而所需修改的工作量取决于该软件的_____。

A.可扩充性 B.可靠性 C.复用性 D.可移植性

试题分析

此题描述的实质是：一个软件要从Windows NT平台移植到UNIX平台中应用。所以其工作量取决于软件的可移植性。可移植性在ISO/IEC 9126软件质量模型中的标准定义为：与软件可从某一环境转移到另一环境的能力有关的一组软件属性。此模型中还对几个常见的质量特性进行了定义，这些特性也需要掌握。

功能性：与功能及其指定的性质有关的一组软件属性。

可靠性：与软件在规定的一段时间内和规定的条件下维持其性能水平有关的一组软件属性。

可用性：与使用的难易程度及规定或隐含用户对使用方式所做的评价有关的软件属性。

效率：与在规定条件下软件的性能水平与所用资源量之间的关系有关的一组软件属性。

可维护性：与软件维护的难易程度有关的一组软件属性。

对于这些特性的更详细说明可以参看《软件设计师考试考点分析与真题详解（计算机与软件工程专业知识篇）》中"4.5.2 软件质量特性"一节。

试题答案

D

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第4章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题16

例题16（2006年11月试题28）

按照ISO/IEC 9126软件质量度量模型定义，一个软件的可靠性的子特性包括_____。

- A.容错性和安全性 B.容错性和适应性
C.容错性和易恢复性 D.易恢复性和安全性

试题分析

ISO/IEC 9126软件质量度量模型的质量特性和质量子特性如表4-8所示。

表4-8 ISO/IEC 9126软件质量度量模型质量特性和质量子特性表

质量属性	子 特 性	说 明
功能性	与功能及其指定的性质有关的一组软件质量	
	适合性	规定任务提供一组功能的能力，以及这组功能的适合程度
	准确性	系统满足需求规格说明和用户目标的程度
	互操作性	同其他指定系统的协同工作能力
	依从性	软件服从有关标准、约定、法规及类似规定的程度
	安全性	避免对程序及数据的非授权故意或意外访问的能力
可靠性	衡量在规定的时间内和规定条件下维护性能水平的一组软件质量	
	成熟性	由软件故障引起失效的频率
	容错性	在错误或违反指定接口情况下维护指定性能水平的能力
	易恢复性	在故障发生后重新建立性能水平，恢复数据的能力和恢复时间
易使用性	与使用难易程度及规定或隐含用户对使用方式所做的评价相关的属性	
	易理解性	用户理解该软件系统的难易程度
	易学习性	用户学习使用该软件系统的难易程度
	易操作性	用户操作该软件系统的难易程度
效率	衡量在规定条件下软件的性能水平和所用资源量之间的关系的属性	
	时间特性	响应和处理时间及软件执行其功能时的吞吐量
	资源特性	软件执行其功能时，所使用的资源量及持续使用时间
可维护性	与软件维护的难易程度相关的一组软件属性	
	易分析性	诊断缺陷或失效原因、判定待修改程度的难易程度
	易更改性	修改、排错或适应环境变化的难易程度
	稳定性	修改造成难以预料的后果的风险程度
	易测试性	测试已修改软件的难易程度
可移植性	与从某一环境转移到另一环境的能力有关的属性	
	适应性	软件无需采用特殊处理就能够适应不同规定环境程度
	易安装性	在指定环境下安装软件的难易程度
	一致性	软件服从与可移植性有关的标准或约束的程度
	易替换性	软件在特定软件环境中用来替代指定的其他软件的可能性和难易程度

故正确答案为：C。

试题答案

C

版权方授权希赛网发布，侵权必究

上一节 本书简介 下一节

例题17

例题17（2006年11月试题29）

详细描述软件的功能、性能和用户界面，以使用户了解如何使用软件。

- A.概要设计说明书 B.详细设计说明书
- C.用户手册 D.用户需求说明书

试题分析

此题考查软件文档的基本知识，下面是软件开发中常见的13种文档的内容说明。

可行性研究报告：说明该软件项目的实现在技术上、经济上和社会因素上的可行性，评述为合理地达到开发目标可供选择的各种可能的实现方案，说明并论证所选定实施方案的理由。

项目开发计划：为软件项目实施方案制定出的具体计划。它应包括各部分工作的负责人员、开发的进度、开发经费的概算、所需的硬件和软件资源等。项目开发计划应提供给管理部门，并作为

开发阶段评审的基础。

软件需求规格说明：对所开发软件的功能、性能、用户界面机运行环境等做出详细的说明。它是用户与开发人员双方对软件需求取得共同理解基础上达成的协议，也是实施开发工作的基础。

数据要求规格说明：给出数据逻辑描述和数据采集的各项要求，为生成和维护系统的数据文件做好准备。

概要设计规格说明：是概要设计工作阶段的成果。它说明系统的功能分配、模块划分、程序的总体结构、输入/输出及接口设计、运行设计、数据结构设计和出错处理设计等，为详细设计奠定基础。

详细设计规格说明：着重描述每个模块如何实现，包括实现算法、逻辑流程等。

用户手册：详细描述软件的功能、性能和用户界面，使用户了解如何使用该软件。

操作手册：为操作人员提供该软件各种运行情况的有关知识，特别是操作方法细节。

测试计划：针对组装测试和确认测试，需要为组织测试制定计划。计划应包括测试的内容、进度、条件、人员、测试用例的选取原则、测试结果允许的偏差范围等。

测试分析报告：测试工作完成以后，应当提交测试计划执行情况的说明。对测试结果加以分析，并提出测试的结论性意见。

开发进度月报：该月报是软件人员按月向管理部门提交的项目进展情况的报告。报告应包括进度计划与实际执行情况的比较、阶段成果、遇到的问题和解决的办法，以及下个月的打算等。

项目开发总结报告：软件项目开发完成之后，应当与项目实施计划对照，总结实际执行的情况，如进度、成果、资源利用、成本和投入的人力。此外，还需对开发工作做出评价，总结经验教训。

维护修改建议：软件产品投入运行之后，可能有修正、更改等问题，应当对存在的问题、修改的考虑，以及修改的影响估计等做详细的描述，写成维护修改建议，提交审批。

从中可以得知，题目所描述的是用户手册，所以答案为：C。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题18

例题18 (2006年11月试题30)

在各类软件维护活动中，_____维护占整个维护工作的比重最大。

A.完善性 B.改正性 C.适应性 D.预防性

试题分析

维护活动可以按维护原因分为：改正性维护、适应性维护、完善性维护和预防性维护。其所占比重情况如图4-3所示。此外，维护在整个软件生存周期中，也占了非常大的比例，一般在60%以上。

图4-3 各种维护比例图

试题答案

A

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题19

例题19 (2006年11月试题31)

给定C语言程序如下：

```
int foo ( int x, int y, int d )
{
    if ( x!=0 ) {
        if ( y==0 ) d=d/x;
        else d=d/ ( x*y ) ;
    }else{
        if ( y==0 ) d=0;
        else d=d/y;
    }
    return d;
}
```

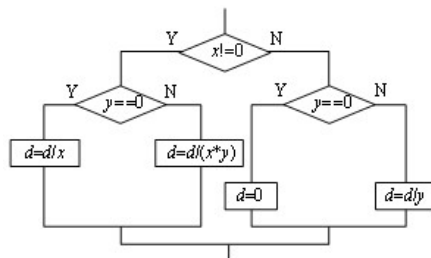
当用路径覆盖法进行测试时，至少需要设计_____个测试用例。

A.3 B.4 C.5 D.8

试题分析

路径覆盖属于白盒测试，其覆盖要求是程序的每一条有效路径都有用例覆盖。所以只需求出程序有多少条有效路径，即可得知需要设计多少个测试用例。但从程序来判断，非常抽象，我们可以画出流程图，这样能清楚地了解程序的结构。程序流程图如图4-4所示。

图4-4 程序流程图



从中我们可以发现有4条有效路径，所以至少需要设计4个测试用例。

试题答案

B

例题20

例题20 (2006年11月试题32)

软件的测试通常分为单元测试、组装测试、确认测试和系统测试4个阶段进行。
属于确认测试阶段的活动。

- A.设计评审 B.代码审查 C.结构测试 D.可靠性测试

试题分析

此题考的是测试相关的基本概念，下面是备选答案中提及的几个活动说明。

设计评审：对设计方案进行正式审查，以确认其是否能够满足需求。

代码审查：对源程序进行正式审查，以确认其是否满足设计的需要，以及能否达到预定的规范要求。

结构测试：也称为路径测试，对软件系统内部进行逻辑测试，常使用白盒测试法。

可靠性测试：验证软件是否满足需求说明书中的可靠性要求，通常使用MTBF（平均无故障时间）、MTTR（平均修复时间）来衡量。由于是验证需求说明的内容，故在确认测试阶段完成。

从以上说明，可以得出此题答案为：D.此外，下面这一组测试相关概念也是需要掌握的，请大家牢记。

功能测试：在规定的一段时间内运行软件系统的所有功能，以验证软件系统有无严重错误。

回归测试：用于验证软件修改后是否引出新的错误。

强度测试：检查在系统运行环境不正常到发生故障的情况，系统可以运行到何种程度。

性能测试：检查系统是否满足在需求规格说明书中规定的性能。

恢复测试：证实在克服硬件故障后，系统能否继续正常工作。

启动/停止测试：验证在机器启动及关机阶段，软件系统正确处理的能力。

配置测试：检查计算机系统内各个设备或各种资源之间的相互连接和功能分配中的错误。

安全性测试：检验系统中已有的系统安全性、保密性措施是否发挥作用，有无漏洞；

可使用性测试：主要从使用的合理性、方便性等角度来对软件系统进行检查，发现人为因素或使用上的问题。

可支持性测试：主要验证系统的支持策略对公司与用户是否切实可行。

安装测试：目的在于查找安装过程中的错误。

互连测试：验证两个或多个不同的系统之间的互连性。

兼容性测试：验证软件产品在不同版本之间的兼容性。

容量测试：也称为压力测试，验证系统的能力最高能达到什么程度。

文档测试：检查用户文档的清晰性和精确性。

表4-9是对以上测试项目与不同测试步骤的关联性进行了总结。

表4-9 各测试活动所处阶段总结表

测试步骤 测试种类	开发阶段的测试					产品阶段的测试				
	设计	单元测试	模块测试	集成测试	部件测试	确认测试	α测试	β测试	验收测试	系统测试
设计评审	√			○						
代码评审		√		√						○
功能测试（黑盒）		√	√	√	√	√	√	√	√	√
结构测试（白盒）		√	√	○						
回归测试			○	√	√	√				√
可靠性测试					√	√	√	√	√	√
强度测试					√	√				√
性能测试			○		√	√	√	√	√	√
恢复测试						√				
启动/停止测试						√				
配置测试					√	√				√
安全性测试						√				
可使用性测试					○	√	√	√		
可支持性测试							√	√		
安装测试						√	√	√		
互连测试				○		√				√
兼容性测试					√	√				
容量测试					√	√				√
文档测试						√	○	√	√	

注：√表示需要，○表示建议。

试题答案

D

版权方授权希赛网发布，侵权必究

上一节 本书简介 下一节

例题21

例题21 (2007年5月试题15)

结构化开发方法中，数据流图是_____阶段产生的成果。

- A. 需求分析
- B. 总体设计
- C. 详细设计
- D. 程序编码

试题分析

结构化分析是面向数据流进行需求分析的方法，数据流图是分析过程中用来描述数据处理过程的工具。它从数据传递和加工的角度，以图形的方式刻画数据流从输入到输出的移动变换过程，是对软件所要处理数据的抽象。由于数据流图只反映系统必须完成的逻辑功能，所以它是一种功能模型。

试题答案

A

版权方授权希赛网发布，侵权必究

上一节 本书简介 下一节

例题22

例题22 (2007年5月试题16)

CVS 是一种_____工具。

- A. 需求分析
- B. 编译
- C. 程序编码
- D. 版本控制

试题分析

CVS (Concurrent Versions System) 是一种广泛应用的，开源的，透明于网络的版本控制系统。它只保存一份源码并记录所有对它的改动。当开发者需要文件的某个特定版本时，CVS会根据那些记录重建出需要的版本。

试题答案

D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

例题23

例题23 (2007年5月试题17)

以下关于原型化开发方法的叙述中，不正确的是_____。

- A. 原型化方法适应于需求不明确的软件开发
- B. 在开发过程中，可以废弃不用早期构造的软件原型
- C. 原型化方法可以直接开发出最终产品
- D. 原型化方法利于确认各项系统服务的可用性

试题分析

原型化开发方法核心理念是通过原型，来不断的获取与完善需求，开发出符合用户需求的软件。在这一个过程中，开发人员会通过原型来与客户进行交流。所以这种方法很适合需求不明确的软件开发。原型开发有多种形式，有时的原型，只是简单的做一个界面出来，给用户看效果，提需求，所以原型化开发时，可以把早期的一些软件原型废弃，而用最新最完整的需求来进行重新开发。从这里也可以看出原型化方法，是不可能直接就把最终产品开发出来的。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

例题24

例题24 (2007年5月试题18)

通常在软件的_____活动中无需用户参与。

A. 需求分析 B. 维护 C. 编码 D. 测试

试题分析

在软件开发的过程中，要求用户参与的活动有：系统分析、软件项目计划、需求分析、软件测试、软件维护。在编码阶段只要高级程序员和程序员参与。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

例题25

例题25 (2007年5月试题19)

进行软件项目的风险分析时，风险避免、风险监控和风险管理及意外事件计划是_____活动中需要考虑的问题。

A. 风险识别 B. 风险预测 C. 风险评估 D. 风险控制

试题分析

项目风险管理实际上就是贯穿在项目开发过程中的一系列管理步骤，其中包括风险识别、风险估计、风险管理策略、风险解决和风险监控。它能让风险管理者主动"攻击"风险，进行有效的风险管理。

在项目管理中，建立风险管理策略和在项目的生命周期中不断控制风险是非常重要的，风险管理包括四个相关阶段：

风险识别：识别风险的方法常用的有风险识别问询法（座谈会、专家法）、财务报表法、流程图法、现场观察法、相关部门配合法和环境分析法等。

风险评估：对已识别的风险要进行估计和评价，风险估计的主要任务是确定风险发生的概率与后果，风险评价则是确定该风险的经济意义及处理的费/效分析，常用的方法有：概率分布、外推法、多目标分析法等。

风险处理：一般而言，风险处理有三种方法，①风险控制法，即主动采取措施避免风险，消灭风险，中和风险或采用紧急方案降低风险。②风险自留，当风险量不大时可以余留风险。③风险转移。

风险控制：包括对风险发生的监督和对风险管理的监督，前者是对已识别的风险源进行监视和控制，后者是在项目实施过程中监督人们认真执行风险管理的组织和技术措施。

从以上分析可以看出"风险避免、风险监控和风险管理及意外事件计划"应是风险控制中的需要考虑的问题。

试题答案：

D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题26

例题26 (2007年5月试题29)

统一过程（UP）是一种用例驱动的迭代式增量开发过程，每次迭代过程中主要的工作流包括捕获需求、分析、设计、实现和测试等。这种软件过程的用户图（Use Case Diagram）是通过_____得到的。

A. 捕获需求 B. 分析 C. 设计 D. 实现

试题分析

统一过程（UP）的基本特征是"用例驱动、以架构为中心的和受控的迭代式增量开发".其核心的工作流包括捕获用户需求、分析、设计、实现和测试等。首先，开发人员通过和用户的沟通、了解，捕获用户需求并制作软件用例图，从而得到软件的用户模型。然后分析并设计满足这些用例的系统，得到分析模型、设计模型和实施模型，进而实现该系统，最后通过测试模型来验证系统是否满足用况中描述的功能。需求捕获过程中得到的用例将各个核心工作流结合为一个整体，驱动整个软件开发过程。

试题答案

A

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题27

例题27 (2007年5月试题31)

ISO/IEC 9126 软件质量模型中第一层定义了六个质量特性，并为各质量特性定义了相应的质量子特性，其中易分析子特性属于软件的质量特性。

A.可靠性 B. 效率 C. 可维护性 D. 功能性

试题分析

参看例题16分析可得知：易分析子特性属于软件的可维护性质量特性。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题28

例题28 (2007年5月试题32)

软件_____的提高，有利于软件可靠性的提高。

A. 存储效率 B. 执行效率 C. 容错性 D. 可移植性

试题分析

软件可靠性是指软件按照用户的要求和设计的目标，执行其功能的正确程度，所以一个可靠的软件应当是正确的、完整的、一致的和健壮的。通常，软件可靠性受到其他质量特性的影响，在4个备选项中，A、B、D都是不利于软件可靠性的因素。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题29

例题29 (2007年5月试题33)

软件能力成熟度模型将软件能力成熟度自低到高依次划分为初始级、可重复级、定义级、管理级和优化级，并且高级别成熟度一定可以达到低级别成熟度的要求。其中_____ 中的开发过程及相应的管理工作均已标准化、文档化，并已建立完善的培训制度和专家评审制度。

A. 可重复级和定义级
B. 定义级和管理级
C. 管理级和优化级
D. 定义级、管理级和优化级

试题分析

软件能力成熟度模型（CMM）将软件能力成熟度自低到高依次划分为初始级、可重复级、定义级、管理级和优化级。

初始级（Initial）。处于这个最低级的组织，基本上没有健全的软件工程管理制度。每件事情都以特殊的方法来做。如果一个特定的工程碰巧由一个有能力的管理员和一个优秀的软件开发组来做，则这个工程可能是成功的。然而通常的情况是，由于缺乏健全的总体管理和详细计划，时间和费用经常超支。结果，大多数的行动只是应付危机，而非事先计划好的任务。处于成熟度等级1的组织，由于软件过程完全取决于当前的人员配备，所以具有不可预测性，人员变化了，过程也跟着变化。结果，要精确地预测产品的开发时间和费用之类重要的项目，是不可能的。

可重复级（Repeatable）。在这一级，有些基本的软件项目的管理行为、设计和管理技术是基于相似产品中的经验，故称为“可重复”。在这一级采取了一定措施，这些措施是实现一个完备过程所必不可少的第一步。典型的措施包括仔细地跟踪费用和进度。不像在第一级那样，在危机状态下行动，管理人员在问题出现时便可发现，并立即采取修正行动，以防它们变成危机。关键的一点是，如没有这些措施，要在问题变得无法收拾前发现它们是不可能的。在一个项目中采取的措施也可用来为未来的项目拟定实现的期限和费用计划。

已定义级（Defined）。在第3级，已为软件生产的过程编制了完整的文档。软件过程的管理方面和技术方面都明确地做了定义，并按需要不断地改进过程，而且采用评审的办法来保证软件的质量。在这一级，可引用CASE环境来进一步提高质量和生产率。而在第-级过程中，“高技术”只会使这一危机驱动的过程更混乱。

已管理级（Managed）。一个处于第4级的公司对每个项目都设定质量和生产目标。这两个量将被不断地测量，当偏离目标太多时，就采取行动来修正。利用统计质量控制，管理部门能区分出随机偏离和有深刻含义的质量或生产目标的偏离（统计质量控制措施的一个简单例子是每千行代码的错误率。相应的目标就是随时间推移减少这个量）。

优化级（Optimizing）。一个第5级组织的目标是连续地改进软件过程。这样的组织使用统计质量和过程控制技术作为指导。从各个方面中获得的知识将被运用在以后的项目中，从而使软件过程融入了正反馈循环，使生产率和质量得到稳步的改进。

除第一级外，每一级都设定了一组目标，如果达到了这组目标，则表明达到了这个成熟级别，自然可以向上一更为成熟的高一级别迈进。CMM体系不主张跨级别的进化，因为从第二级开始，每一个低级别的实现均是更高级别实现的基础。

试题答案

D

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题30

例题30（2007年5月试题34）

LOC 是软件规模的一种量度，它表示_____。

- A. 软件功能数
- B. 源代码行数

C. 每单位成本数

D. 工作量

试题分析

软件规模代码行（LOC, Line of Code）是软件规模的一种量度，它表示源代码行数。

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第4章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题31

例题31（2007年5月试题35）

阅读下列流程图：

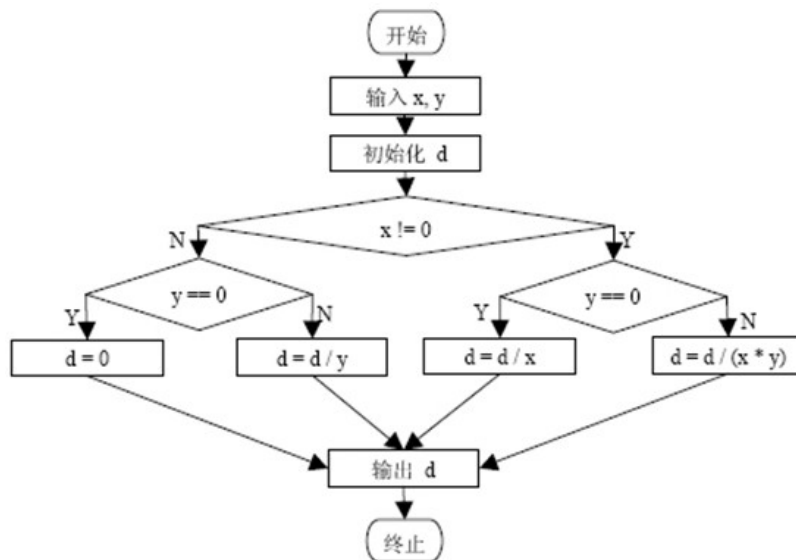


图4-5 程序流程图

当用判定覆盖法进行测试时，至少需要设计_____个测试用例。

A. 2 B. 4 C. 6 D. 8

试题分析

判定覆盖又称为分支覆盖，它要求设计足够多的测试用例，使得程序中每个判定至少有一次为真值，有一次为假值，即：程序中的每个分支至少执行一次。根据题图中，上层一个条件语句，下层有两个条件语句，所以上层为真或假时，下层的两个判定也要为真为假各一次，总共4次。

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题32

例题32 (2007年5月试题37)

在某大学学生学籍管理信息系统中，假设学生年龄的输入范围为 16~40,则根据黑盒测试中的等价类划分技术，下面划分正确的是_____。

- A. 可划分为 2 个有效等价类，2 个无效等价类
- B. 可划分为 1 个有效等价类，2 个无效等价类
- C. 可划分为 2 个有效等价类，1 个无效等价类
- D. 可划分为 1 个有效等价类，1 个无效等价类

试题分析

是把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例。划分等价类：等价类是指某个输入域的子集合。在该子集中，各个输入数据对于揭露程序中的错误都是等效的，并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试。因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据，取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

有效等价类：是指对于程序的规格说明来说是合理的，有意义的输入数据构成的集合。利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

无效等价类：与有效等价类的定义恰巧相反。

本题当中指明了输入范围是16~40,所以在这个区间的为有效等价类，小于16或大于40的为无效等价类。所以选B答案，可划分为1个有效等价类，2个无效等价类。

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题33

例题33 (2007年11月试题29-30)

在软件开发中，（29）不能用来描述项目开发的进度安排。在其他三种图中，可用（30）动态地反映项目开发进展情况。

- （29）A. 甘特图 B. PERT图 C. PERT/CPM图 D. 鱼骨图
- （30）A. 甘特图 B. PERT图 C. PERT/CPM图 D. 鱼骨图

试题分析

"鱼骨图"又称"特性要因图",就是将造成某项结果的众多原因，以系统的方式图解之，也就是以

图表的方式来表达结果与原因的关系，其图形像鱼骨，因此称为“鱼骨图”。

甘特图--甘特图可以直观地表明任务计划在什么时候进行，及实际进展与计划要求的对比。管理者由此可以非常便利地弄清每一项任务（项目）还剩下哪些工作要做，并可评估工作是提前还是滞后，亦或正常进行。除此以外，甘特图还有简单、醒目和便于编制等特点。

PERT（性能评审技术）图是一个项目管理工具，用于规划，组织和调整项目内的任务。一个PERT图显示了一个项目的图形解释，这种图是网络装的，由号码标记的节点组成，节点由带标签的带方向箭头的线段连接，展现项目中的事件或转折点，以及展现项目中的任务。带方向箭头的线段表示任务的先后顺序。例如，在PERT图中，在节点1,2,4,8和10之间的任务必须按顺序完成，这叫做系列任务的依存性。

PERT/CPM图是一个项目管理工具，用于规划，组织和调整项目内的任务。PERT是基于性能评审技术，一种美国海军于1950年代发展起来的管理潜艇导弹计划的方法。一种类似的方法是关键途径方法（CPM），它是在同样的时间内由私营部门发展的项目管理方法，现在已经成为PERT的同义词。

试题答案

D A

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题34

例题34（2007年11月试题31）

选择软件开发工具时，应考虑功能、_____、稳健性、硬件要求和性能、服务和支持。

A. 易用性 B. 易维护性 C. 可移植性 D. 可扩充性

试题分析

开发工具是给开发者用的，开发人员是这些工具的用户。我国的软件开发已经逐步从原来的手工工作坊式发展到了软件工程的阶段。同时，软件开发本身也在不断发展，已从“算法+数据结构=程序”逐步发展到了“设计模式+对象组件+开发工具=程序”。开发工具的选择，已经成为软件开发成功的要素之一。开发工具的选择主要决定于两个因素：所开发系统的最终用户和开发人员。最终用户需求是一切软件的来源和归宿，也是影响开发工具的决定性因素；开发人员的爱好、习惯、经验也影响着开发工具的选择。严格的软件工程管理和开发人员的技术水平是软件开发成功的关键。所以在选择时，应考虑功能、易用性、稳健性、硬件要求和性能、服务和支持。

试题答案

A

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

例题35

例题35 (2007年11月试题32)

内聚性和耦合性是度量软件模块独立性的重要准则，软件设计时应力求_____。

- A. 高内聚，高耦合 B. 高内聚，低耦合
C. 低内聚，高耦合 D. 低内聚，低耦合

试题分析

在软件设计时，我对软件模块独立性通常提出高内聚，低耦合；原因是：

1.高耦合度的系统是很难维护的。一处的修改引起另一处甚至更多处的变动。你可以通过以下方法降低程序的耦合度：隐藏实现细节，强制构件接口定义，不使用公用数据结构，不让应用程序直接操作数据库。耦合度低的软件可以很容易被重用、维护和扩充。

2.高内聚性的软件更容易维护和改进。判断一个模块是否有高的内聚性，看一看你是否能够用一个简单的句子描述它的功能就行了。如果你用了一段话或者你需要使用类似"和"、"或"等连词，则说明你需要将该模块细化。只有高内聚性的模块才可能被重用。

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题36

例题36 (2007年11月试题34)

对于如下的程序流程，当采用语句覆盖法设计测试案例时，至少需要设计_____个测试案例。

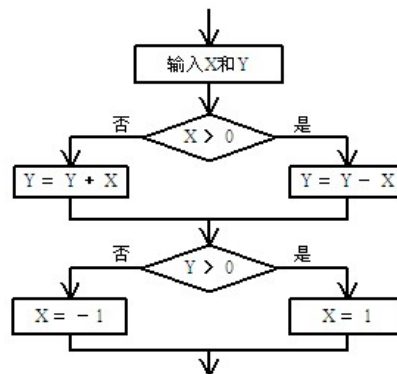


图4-6 程序流程图

- A. 1B. 2 C. 3 D. 4

试题分析

语句覆盖是指设计测试用例，使程序中的每个可执行语句至少执行一次。从流程图来看，图中有两个判断条件 $x > 0$ 和 $y > 0$ ，分别有四种情况的组合出现，但由于是顺序执行且求语句覆盖的至少测试

案例，所以只要选择两个测试案例就可以了。

试题答案

B

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题37

例题37 (2007年11月试题35-36)

为验证程序模块A是否正确实现了规定的功能，需要进行 (35) ;为验证模块A能否与其他模块按照规定方式正确工作，需要进行 (36) 。

(35) A. 单元测试 B. 集成测试 C. 确认测试 D. 系统测试

(36) A. 单元测试 B. 集成测试 C. 确认测试 D. 系统测试

试题分析

单元测试：所谓单元是指程序中的一个模块或一个子程序，是程序运行的最小单元，或是程序运行的独立编译单位，因此单元测试也称模块测试。从模块的内部结构出发设计测试用例，进行测试。

集成测试：对每一个模块完成了单元测试以后，需要按照设计时作出的层次模块图把它们连接起来，进行组装测试，也称为组合测试或综合。

确认测试：确认测试是进一步检查软件是否符合软件需求规格说明书的全部要求。

系统测试：系统测试是将信息系统的所有组成部分包括软件、硬件、用户以及环境等综合在一起进行测试，以保证系统的各组成部分协调运行。

试题答案：

A B

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题38

例题38 (2008年5月试题15)

_____是一种面向数据流的开发方法，其基本思想是软件功能的分解和抽象。

A. 结构化开发方法 B. Jackson系统开发方法

C. Booch 方法 D. UML (统一建模语言)

试题分析

本题考查软件工程的结构化开发方法，是常考的知识点。

结构化开发方法（Structured Method,结构化方法）是强调开发方法的结构合理性以及所开发软件的结构合理性的软件开发方法。结构是指系统内各个组成要素之间的相互联系、相互作用的框架。结构化开发方法提出了一组提高软件结构合理性的准则，如分解与抽象、模块独立性、信息隐蔽等。针对软件生存周期各个不同的阶段，它有结构化分析（SA）、结构化设计（SD）和结构化程序设计（SP）等方法。它的基本思想：把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。

Jackson系统开发方法（JSD,Jackson System Development）是一种典型的面向数据结构的分析设计方法。Jackson系统开发方法的系统模型就是相互通讯的一组进程的集合。进程间的通讯方式有以下三种：

- （1）进程同步发生。
- （2）通过数据通道发送/接收活动发生。
- （3）访问公用存储信息。

Booch最先描述了面向对象的软件开发方法的基础问题，指出面向对象开发是一种根本不同于传统的功能分解的设计方法。面向对象的软件分解更接近人对客观事务的理解，而功能分解只通过问题空间的转换来获得。

UML（统一建模语言）软件工程领域在1995年~1997年取得了前所未有的进展，其成果超过软件工程领域过去15年的成就总和，其中最重要的成果之一就是统一建模语言（UML）的出现。UML将是面向对象技术领域内占主导地位的标准建模语言。

UML不仅统一了Booch方法、OMT方法、OOSE方法的表示方法，而且对其作了进一步的发展，最终统一为大众接受的标准建模语言。UML是一种定义良好、易于表达、功能强大且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。它的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发全过程。

所以本题中讲面向数据流的开发方法，其基本思想是软件功能的分解和抽象开发方法是结构化开发方法，A选项。

试题答案

A

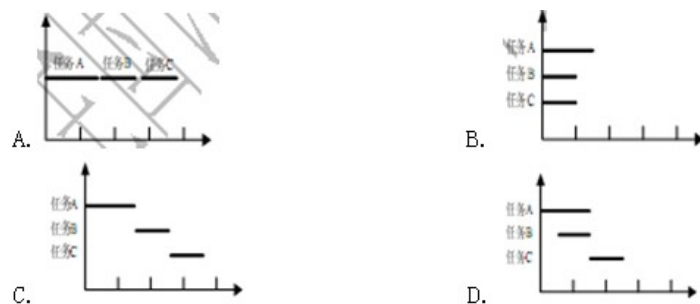
版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题39

例题39（2008年5月试题17）

某项目制定的开发计划中定义了三个任务，其中任务A首先开始，且需要3周完成，任务B必须在任务A启动1周后开始，且需要2周完成，任务C必须在任务A完成后才能开始，且需要2周完成。该项目的进度安排可用下面的甘特图来描述。



试题分析

本题考查甘特图，是常考的知识点。

在Gantt图即甘特图中，横轴表示时间，纵轴表示任务，用水平线段表示任务的工作阶段，线段的起点和终点分别对应任务的开工时间和完成时间。依题意，A是要3周完成，而B是在A开工后1周开工，且做2周，所以和A同时完成。C是在A完成后才动工，所以C在A和B的后面。则D选项正确。

试题答案

D

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题40

例题40（2008年5月试题18-19）

风险分析在软件项目开发中具有重要作用，包括风险识别、风险预测、风险评估和风险控制等。"建立风险条目检查表"是（18）时的活动，"描述风险的结果"是（19）时的活动。

（18）A. 风险识别 B. 风险预测 C. 风险评估 D. 风险控制

（19）A. 风险识别 B. 风险预测 C. 风险评估 D. 风险控制

试题分析

本题考查软件工程的风险分析，是常考的知识点。

风险分析是找出行动方案的不确定性（主观上无法控制）因素，分析其环境状况和对方案的敏感程度；估计有关数据，包括行动方案的费用，在不同情况下得到的收益以及不确定性因素各种机遇的概率，计算各种风险情况下的经济效果；作出正确判断，等等。

进行风险分析时需要考虑的因素很多，而且多变，主要有：①技术经济分析中采用的数据来源和精确程度。②企业的类型及其稳定性。如采矿企业因地质条件的变化通常要承担较大的风险。③企业的厂房和设备的类型。如有些建筑物和设备有较明确的经济寿命和转卖价值，有的经济寿命不明确，也无多大转卖价值。这时对前一类企业投资的风险就小于后一类企业。④分析阶段的长短。如投资回收期的延长将会增大投资的风险，等等。

风险识别是试图系统化地确定对项目计划（估算、进度、资源分配）的威胁。风险识别的一个方法是建立风险条目检查表。

风险预测，又称风险估算，它从两个方面评估一个风险：风险发生的可能性或概率；以及如果风险发生了，所产生的后果。通常，项目计划人员与管理人员、技术人员一起，进行4种风险预测活动：

- (1) 建立一个尺度或标准，以反映风险发生的可能性；
- (2) 描述风险的后果；
- (3) 估计风险对项目和产品的影响；
- (4) 标注风险预测的整体精确度，以免产生误解。

所以，本题的问题建立风险条目检查表是风险识别时的活动，描述风险的结果是风险预测时的活动。

试题答案

A B

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题41

例题41 (2008年5月试题29)

极限编程 (eXtreme Programming) 是一种轻量级软件开发方法，_____不是它强调的准则。

- A. 持续的交流和沟通 B. 用最简单的设计实现用户需求
- C. 用测试驱动开发 D. 关注用户反馈

试题分析

本题考查软件工程的极限编程，是常考的知识点。

Extreme Programming (极限编程，简称XP) 是由Kent Beck在1996年提出的。XP是一个轻量级的、灵巧的软件开发方法；同时它也是一个非常严谨和周密的方法。它的基础和价值观是交流、朴素、反馈和勇气；即，任何一个软件项目都可以从四个方面入手进行改善：加强交流；从简单做起；寻求反馈；勇于实事求是。XP是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。

题中对于极限编程不是强调的准则是用测试驱动开发，选择C选项。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题42

例题42 (2008年5月试题30)

学赛网采用的软件开发过程通过了CMM2认证，表明该公司_____。

- A. 开发项目成效不稳定，管理混乱
- B. 对软件过程和产品质量建立了定量的质量目标
- C. 建立了基本的项目级管理制度和规程，可对项目的成本、进度进行跟踪和控制
- D. 可集中精力采用新技术新方法，优化软件过程

试题分析

本题考查软件管理的CMM2。

CMM 2 (可重复级) 就是建立了基本的项目级管理过程，可对项目的成本、进度进行跟踪和控制，生产的过程、标准、工作产品以及服务都是被严格定义和文档化的。基于以往管理类似项目的经验，计划和管理新项目，并可依据一定的标准重复利用类似的软件产品。CMM 2的核心就是重复利用。

CMM2由6个关键过程域 (KPA) 组成：需求管理 (RM)、软件项目计划 (SPP)、软件项目跟踪与监控 (SPTO)、软件子合同管理 (SSM) (本文略)、软件质量保证 (SQA)、软件配置管理 (SCM)。

需求管理 (Requirement Management) 的目的是为了在客户和处理客户需求的软件项目之间建立共识。这是软件项目规划 (SPP) 和管理 (SPTO) 的基础，需求变更依赖于配置管理 (SCM) 的变更控制流程。在项目实施过程中，最突出的现象就是项目组成员没有完全理解需求，软件需求不稳定，客户经常变更需求，无法有效控制需求变更，需求变更往往造成项目延期和费用超支。

试题答案

D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题43

例题43 (2008年5月试题32)

在软件设计和编码过程中，采取"_____"的做法将使软件更加容易理解和维护。

- A. 良好的程序结构，有无文档均可
- B. 使用标准或规定之外的语句
- C. 编写详细正确的文档，采用良好的程序结构
- D. 尽量减少程序中的注释

试题分析

本题考查软件工程的软件设计风格，是常考的知识点。

为了提高程序的可阅读性，要建立良好的编程风格。可分为以下几条：

1.源程序文档化

(1) 标识符应按意取名。

(2) 程序应加注释。注释是程序员与日后读者之间通信的重要工具，用自然语言或伪码描述。它说明了程序的功能，特别在维护阶段，对理解程序提供了明确指导。注释分序言性注释和功能性注释。序言性注释应置于每个模块的起始部分，主要内容有：

- ①说明每个模块的用途、功能。
- ②说明模块的接口：调用形式、参数描述及从属模块的清单。
- ③数据描述：重要数据的名称、用途、限制、约束及其他信息。
- ④开发历史：设计者、审阅者姓名及日期，修改说明及日期。

功能性注释嵌入在源程序内部，说明程序段或语句的功能以及数据的状态。注意以下几点：

- ①注释用来说明程序段，而不是每一行程序都要加注释。
- ②使用空行或缩格或括号，以便很容易区分注释和程序。
- ③修改程序也应修改注释。

2.数据说明

为了使数据定义更易于理解和维护，有以下指导原则：

(1) 数据说明顺序应规范，使数据的属性更易于查找，从而有利于测试、纠错与维护。例如按以下顺序：常量寿命、类型说明、全程量说明、局部量说明。

- (2) 一个语句说明多个变量时，各变量名按字典序排列。
- (3) 对于复杂的数据结构，要加注释，说明在程序实现时的特点。

3.语句构造

语句构造的原则是：简单直接，不能为了追求效率而使代码复杂化。为了便于阅读和理解，不要一行多个语句。不同层次的语句采用缩进形式，使程序的逻辑结构和功能特征更加清晰。要避免复杂的判定条件，避免多重的循环嵌套。表达式中使用括号以提高运算次序的清晰度等等。

4.输入和输出

在编写输入和输出程序时考虑以下原则：

- (1) 输入操作步骤和输入格式尽量简单。
- (2) 应检查输入数据的合法性、有效性，报告必要的输入状态信息及错误信息。
- (3) 输入一批数据时，使用数据或文件结束标志，而不要用计数来控制。
- (4) 交互式输入时，提供可用的选择和边界值。
- (5) 当程序设计语言有严格的格式要求时，应保持输入格式的一致性。
- (6) 输出数据表格化、图形化。

输入、输出风格还受其他因素的影响，如输入、输出设备，用户体验及通信环境等。

5.效率

效率是指处理机时间和存储空间的使用，对效率的追求明确以下几点：

- (1) 效率是一个性能要求，目标在需求分析给出。
- (2) 追求效率建立在不损害程序可读性或可靠性基础上，要先使程序正确，再提高程序效率，先使程序清晰，再提高程序效率。
- (3) 提高程序效率的根本途径在于选择良好的设计方法、良好的数据结构算法，而不是靠编程时对程序语句做调整。

使软件更加容易理解和维护，就要对开发文档编写详细和正确，所以本题的正确答案是C答案。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题44

例题44 (2008年5月试题33)

软件维护成本在软件成本中占较大比重。为降低维护的难度，可采取的措施有_____。

- A. 设计并实现没有错误的软件
- B. 限制可修改的范围
- C. 增加维护人员数量
- D. 在开发过程中就采取有利于维护的措施，并加强维护管理

试题分析

本题考查软件工程的软件维护知识，是常考的知识点。

软件维护通常分为四类维护活动：改正性维护，也就是诊断和改正在使用过程中发现的软件错误；适应性维护，即修改软件以适应环境的变化；完善性维护，即根据用户的要求改进或扩充软件使它更完善；预防性维护，即修改软件为将来的维护活动预先做准备。

A 答案是错误的。设计并实现没有错误的软件本身是不可能的，只能说设计时尽量少一些错误。

B 答案和 C 答案与降低软件维护的难度无关。

D 答案是正确的。在开发的过程中，采取有利于维护的措施，比如：完善使用手册。另个还加强维护管理。

试题答案

D

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题45

例题45 (2008年5月试题34)

软件文档按照其产生和使用的范围可分为开发文档、管理文档和用户文档。其中开发文档不包括_____。

- A. 软件需求说明 B. 可行性研究报告
C. 维护修改建议 D. 项目开发计划

试题分析

本题考查软件工程的文档知识，是常考的知识点。

软件文档（document）也称文件，通常指的是一些记录的数据和数据媒体，它具有固定不变的形式，可被人 and 计算机阅读。它和计算机程序共同构成了能完成特定功能的计算机软件（有人把源程序也当作文档的一部分）。我们知道，硬件产品和产品资料在整个生产过程中都是有形可见的，软件生产则有很大不同，文档本身就是软件产品。没有文档的软件，不成其为软件，更谈不到软件产品。软件文档的编制在软件开发工作中占有突出的地位和相当的工作量。高效率、高质量地开发、分发、管理和维护文档对于转让、变更、修正、扩充和使用文档，对于充分发挥软件产品的效益有着重要意义。

软件文档可以分开发文档、管理文档和用户文档三大类。

开发文档包括：《功能要求》、《投标方案》、《需求分析》、《技术分析》、《系统分析》、《数据库文档》、《功能函数文档》、《界面文档》、《编译手册》、《QA文档》、《项目总结》等。

管理文档包括：《产品简介》、《产品演示》、《疑问解答》、《功能介绍》、《技术白皮书》、《评测报告》等。

用户文档包括：《安装手册》、《使用手册》、《维护手册》、《用户报告》、《销售培训》等。

根据开发文档可知，A、B、D是开发时的产生文档，而C是用户档。所以本题的答案C选项。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

例题46

例题46（2008年5月试题35-36）

软件测试是软件开发中不可缺少的活动，通常（35）在代码编写阶段进行。检查软件的功能是否与用户要求一致是（36）的任务。

（35）A. 验收测试 B. 系统测试 C. 单元测试 D. 集成测试

（36）A. 验收测试 B. 系统测试 C. 单元测试 D. 集成测试

试题分析

本题考查软件工程的几种测试的定义，是常考的知识点。

按照开发阶段划分软件测试可分为：单元测试、集成测试、系统测试、确认测试和验收测试。

单元测试又称模块测试，是针对软件设计的最小单位--程序模块进行正确性检验的测试工作。其目的在于检查每个程序单元能否正确实现详细设计说明中的模块功能、性能、接口和设计约束等要

求，发现模块内部可能存在的各种错误。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

集成测试也叫做组装测试。通常在单元测试的基础上，将所有的程序模块进行有序的、递增的测试。集成测试是检验程序单元或部件的接口关系，逐步集成为符合概要设计要求的程序部件或整个系统。

确认测试是通过检验和提供客观证据，证实软件是否满足特定预期用途的需求。确认测试是检测与证实软件是否满足软件需求说明书中规定的要求。

验收测试是按照项目任务书或合同、供需双方约定的验收依据文档进行的对整个系统的测试与评审，决定是否接收或拒收系统。

所以在本题当中第（35）空应选择C答案，单元测试是在代码编写阶段进行的。第（36）空应选择A答案，验收测试阶段就检查软件的功能是否与用户要求一致。

试题答案

C A

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第4章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题47

例题47（2008年12月试题16-17）

若一个项目由9个主要任务构成，其计划图（如下图所示）展示了任务之间的前后关系以及每个任务所需天数，该项目的关键路径是（16），完成项目所需的最短时间是（17）天。

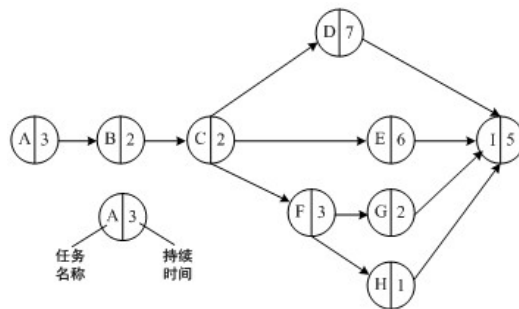


图4-7 任务计划图

(16) A. A→B→C→D→I B. A→B→C→E→I

C. A→B→C→F→G→I D. A→B→C→F→H→I

(17) A. 16 B. 17 C. 18 D. 19

试题分析

本题考点是关键路径。关键路径是计划图中的“最长路径”而非“最短路径”。很明显，图中的A→B→C→D→I最长，长度为：19。

试题答案

A D

例题48

例题48 (2008年12月试题18)

RUP (Rational Unified Process) 分为4个阶段，每个阶段结束时都有重要的里程碑，其中生命周期架构是在_____结束时的里程碑。

- A. 初启阶段 B. 精化阶段 C. 构建阶段 D. 移交阶段

试题分析

请参看例题14分析。

试题答案

B

例题49

例题49 (2008年12月试题19)

在软件工程环境中进行风险识别时，常见的、已知的及可预测的风险类包括产品规模、商业影响等，与开发工具的可用性及质量相关的风险是_____。

- A. 客户特性 B. 过程定义 C. 开发环境 D. 构建技术

试题分析

该题考查内容为风险的类型，风险类型主要包括以下三类：

(1) 项目风险：指潜在的预算、进度、人力（工作人员及组织）、资源、客户和需求等方面的问题以及它们对软件项目的影响。例如：项目复杂性、规模和结构不确定性等都是项目风险。项目风险威胁到项目计划，也就是说，如果项目风险变成现实，有可能会拖延项目的进度，增加项目的成本。

(2) 技术风险：指潜在的设计、实现、接口、验证和维护等方面的问题。此外，规约的二义性、技术的不确定性、陈旧的技术和"先进的"技术也是技术风险因素。技术风险威胁到要开发软件的质量及交付时间，如果技术风险变成现实，则开发工作可能变得很困难或根本不可能。

(3) 商业风险：在信息系统项目中，商业风险威胁到要开发系统的生存能力。一般主要有5类商业风险：

- 开发了一个没有人真正需要的优秀产品或系统（市场风险）。
- 开发的产品不再符合公司的整体商业策略（策略风险）。

- 开发了一个销售部门不知道如何去卖的产品（销售风险）。
- 由于重点的转移或人员的变动而失去了高级管理层的支持（管理风险）。
- 没有得到预算或人力上的保证（预算风险）。

题目中的产品规模属项目风险，商业影响属商业风险，而与开发工具的可用性及质量相关的属于技术风险。但技术风险中又有几个方面，包括团队对开发技术的把握程度（如应用新技术），开发环境是否满足开发的需要，接口方面是否能处理好，维护方面等，开发工具显然是属于开发环境范畴。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题50

例题50（2008年12月试题29）

软件能力成熟度模型（CMM）将软件能力成熟度自低到高依次划分为初始级、可重复级、定义级、管理级和优化级。其中_____对软件过程和产品都有定量的理解与控制。

- A. 可重复级和定义级 B. 定义级和管理级
C. 管理级和优化级 D. 定义级、管理级和优化级

试题分析

请参看例题29分析。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题51

例题51（2008年12月试题30）

极限编程（XP）包含了策划、设计、编程和测试四个活动，其十二个最佳实践中的"持续集成"实践在_____活动中进行。

- A. 策划和设计 B. 设计和编程
C. 设计和测试 D. 编程和测试

试题分析

XP中的12个最佳实践是：

- (1) 计划游戏。
- (2) 小型发布。
- (3) 隐喻。
- (4) 简单设计。
- (5) 测试先行。
- (6) 重构。
- (7) 结对编程。
- (8) 集体代码所有制。
- (9) 持续集成。
- (10) 每周工作40小时。
- (11) 现场客户。
- (12) 编码标准。

其中的持续集成是要求XP团队每天尽可能多次地做代码集成，每次都在确保系统运行的单元测试通过之后进行。这样，就可以及早地暴露、消除由于重构、集体代码所有制所引入的错误，从而减少解决问题的痛苦。要在开发过程中做到持续集成并不容易，首先需要养成这个习惯。所以持续集成是在"编程和测试"活动中进行的。

试题答案

D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题52

例题52 (2008年12月试题31)

ISO/IEC 9126 软件质量模型中第一层定义了六个质量特性，并为各质量特性定义了相应的质量子特性。子特性_____属于可靠性质量特性。

- A. 准确性 B. 易理解性 C. 成熟性 D. 易学性

试题分析

ISO/IEC 9126 软件质量模型请参看例题16分析。

在该模型中，可靠性有：成熟性、容错性、易恢复性这三个子特性。所以本题选C.

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

例题53

例题53 (2008年12月试题32)

UP (统一过程) 是用例驱动的、以架构为核心、迭代和增量的软件过程框架，它提供了一种_____的特性。

- A. 演进 B. 敏捷 C. 测试驱动 D. 持续集成

试题分析

UP中的软件过程在时间上被分解为四个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和

版本，但这个软件版本并非最终版，它只是

化阶段、构建阶段和交付阶段，从而产生一段，最终产生软件产品。使用这种方法进行

软件开发，软件的功能是一代一代逐步完善的，所以包含演进的特性。

试题答案

A

版权方授权希赛网发布，侵权必究

上一节

本书简介

下一节

例题54

例题54 (2008年12月试题33)

系统测试人员与系统开发人员需要通过文档进行沟通，系统测试人员应根据一系列文档对系统进行测试，然后将工作结果撰写成_____,交给系统开发人员。

- A. 系统开发合同 B. 系统设计说明书
C. 测试计划 D. 系统测试报告

试题分析

本题考查基本文档的内容及用途。系统开发合同是在系统开发以前完成的文档记录。系统设计说明书是在系统设计阶段完成的文档，这都与测试没什么关系。软件项目的测试计划是描述测试目的、范围、方法和软件测试的重点等的文档。对于验证软件产品的可接受程度编写测试计划文档是一种有用的方式。详细地测试计划可以帮助测试项目组之外的人了解为什么和怎样验证产品。而系统测试报告是对测试后的结果进行记录的文档，把该文档反馈给系统开发人员，能让他们得知自己开发的模块有什么样的问题，以便进行及时的修正，所以该题选D。

试题答案

D

版权方授权希赛网发布，侵权必究

例题55

例题55 (2008年12月试题34)

系统的可维护性可以用系统的可维护性评价指标来衡量。系统的可维护性评价指标不包括_____。

A. 可理解性 B. 可修改性 C. 准确性 D. 可测试性

试题分析

软件可维护性是指纠正软件系统出现的错误和缺陷，以及为满足新的要求进行修改、扩充和压缩的容易程度。目前广泛用来衡量程序可维护性的因素包括可理解性、可测试性、可修改性。

试题答案

C

[版权方授权希赛网发布，侵权必究](#)

例题56

例题56 (2008年12月试题35)

某程序根据输入的三条线段长度，判断这三条线段能否构成三角形。以下6个测试用例中，_____两个用例属于同一个等价类。

①6、7、13; ②4、7、10; ③9、20、35;

④9、11、21; ⑤5、5、4; ⑥4、4、4.

A. ①② B. ③④ C. ⑤⑥ D. ①④

试题分析

本题考黑盒测试的等价类划分。判断三条线段是否构成三角形的条件是：任意两边之和大于第三边。

按此原则，可将6个测试用例进行等价类划分为：

不能构成三角形：①、③、④

对于能构成三角形的，进一步可划分为：

两边之和等于第三边：①

两边之类小于第三边：③、④

能构成三角形：②、⑤、⑥

对于能构成三角形的，进一步可划分为：

普通三角形：②

等腰三角形：⑤

等边三角形：⑥

所以正确答案为：B.

试题答案

B

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第 4 章：软件工程

作者：希赛教育软考学院 来源：希赛网 2014年02月08日

例题57

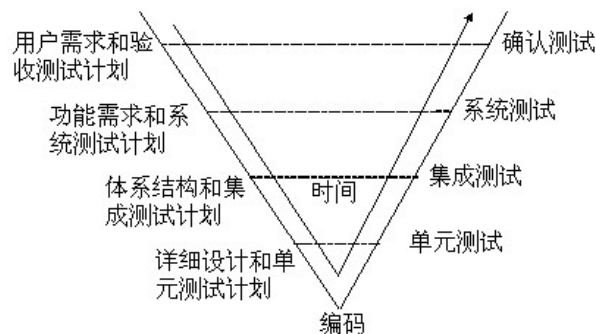
例题57（2008年12月试题36）

在模拟环境下，常采用黑盒测试检验所开发的软件是否与需求规格说明书一致。其中有效性测试属于____中的一个步骤。

A. 单元测试 B. 集成测试 C. 确认测试 D. 系统测试

试题分析

软件测试主要有：单元测试、集成测试、系统测试、确认测试。这几个步骤与软件开发的对应如下图所示：



单元测试也称模块测试，通常可放在编程阶段，由程序员对自己编写的模块自行测试，检查模块是否实现了详细设计说明书中规定的功能和算法。单元测试主要发现编程和详细设计中产生的错误，单元测试计划应该在详细设计阶段制定。单元测试期间着重从以下几个方面对模块进行测试：模块接口、局部数据结构、重要的执行通路、出错处理、边界条件等。

集成测试也称组装测试，它是对由各模块组装而成的程序进行测试，主要目标是发现模块间的接口和通信问题。集成测试主要发现设计阶段产生的错误，通常采用黑盒测试。集成测试计划应该在概要设计阶段制定。集成的方式可分为非增殖式（非渐增方式、一次性组装方式、整体拼装方式）和增殖式（渐增式组装方式）。

系统测试的任务是把软件放在实际的硬件和网络环境中进行测试，主要测试软件的非功能需求和质量属性是否得到满足。系统测试是根据系统方案说明书来设计测试用例，通常采用黑盒测试。常见的系统测试主要有恢复测试、安全性测试、强度测试、性能测试、可靠性测试和安装测试。

（1）恢复测试：恢复测试检测系统的容错能力。

（2）安全性测试：检测系统的安全机制、保密措施是否完善，主要是为了检测系统的防范能

力。系统安全性设计的准则是使非法入侵者所花费的代价比进入系统后所得到的好处要大，这样，非法入侵者就已经无利可图了。

(3) 强度测试：对系统在异常情况下的承受能力的测试，是检查系统在极限状态下运行时，性能下降的幅度是否在允许的范围内。

(4) 性能测试：检查系统是否满足系统方案说明书对性能的要求。性能测试要覆盖软件测试的各阶段，而不是等到系统的各部分组装之后。

(5) 可靠性测试：主要测试系统的MTBF和MTTR。

(6) 安装测试：检测在安装过程中是否有错误、是否容易操作等。

确认测试的任务是检查软件的功能、性能和其他特征是否与用户的需求一致，它是以需求规格说明书作为依据的测试，通常采用黑盒测试。软件确认测试首先要进行有效性测试以及软件配置审查，然后进行验收测试。经过管理部门的认可和专家的鉴定后，软件即可交给用户使用。

(1) 有效性测试。在模拟环境下，通过黑盒测试检验所开发的软件是否与需求规格说明书一致。测试人员可以由开发者的内部人员组成，但最好是没有参加该项目的有经验的软件设计人员。在所有测试用例完成后，要列出缺陷清单。

(2) 软件配置审查。主要是检查软件（源程序、目标程序）和文档（包括面向开发和用户的文档）是否齐全以及分类是否有序。确保文档、资料的正确和完成，以便维护阶段使用。

(3) 验收测试。以用户为主的测试，软件开发人员和质量保证人员也应该参加。在验收测试之前，要对用户进行培训，以便熟悉该系统。验收测试的用例由用户参与设计，主要验证软件的功能、性能、可移植性、兼容性、容错性等。测试时一般采用实际数据。

在确认测试时，如果一个软件是为某个客户定制的，那么由客户实施验收测试，以便确认该软件是他所需要的。但是，对于那些作为产品被众多客户使用的软件，就不可能为每个客户做验收测试。大多数软件生产商使用一种 α 测试和 β 测试的过程。

(1) α 测试：在开发者的现场由客户来实施的，被测试的软件是在开发者从用户的角度进行常规设置的环境下运行的。 α 测试在开发者的指导下进行测试，开发者负责记录发现的错误和使用中遇到的问题。也就是说， α 测试是在“受控的”环境中进行的。

(2) β 测试：在一个或多个客户的现场由该软件的最终用户实施的。与 α 测试不同的是，进行 β 测试时开发者通常是不在场的。用户负责记录发现的错误和使用中遇到的问题并把这些问题的报告给开发者。

从以上的描述可以看出有效性测试是确认测试的一个步骤。

试题答案

C

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

例题1