

单选题

- 1.劣质成本关注故障、过程、机会和顾客，下面(D)可能造成劣质成本。
A、纠正单元测试中的错误;
B、编写软件质量计划;
C、各种技术评审和管理评审;
D、为修正客户发现的问题，紧急发布程序补丁;
- 2.评审软件需求是为了确保(A)
A、完整性、正确性、一致性、可行性
B、可测性、可靠性、充分性、完整性
C、完整性、正确性、可靠性、充分性
D、完整性、正确性、可靠性、可行性
- 3、下列关于软件可靠性测试的说法中，错误的是(A)
A、发现软件缺陷是软件可靠性测试的主要目的
B、软件可靠性测试通常用于有可靠性要求的软件。
C、在一次软件可靠性测试中，执行的测试用例必须完全符合所定义的软件运行剖面 D、可靠性测试通常要对测试结果进行分析才能获得测试结论
- 4、在同行评审的过程中，要记录同行评审的实现和结果数据，数据的实例有(D)
A、软件工作产品规模
B、评审小组的规模和组成
C、每个人评审人员的准备时间
D、以上都是

判断题

- 1、面向对象程序与结构化程序的复杂性度量方法不同。(√)
- 2、过程管理是项目管理的工作。(√)
- 3、同行评审的主要目标在于检测错误、核对与标准的偏离。(√)
- 4、在任何软件机构中，定期、不定期的培训、再培训都是必须而且是必要的。(√)

填空题

- 1、软件缺陷是由很多原因造成的，包含软件产品（市场需求文档、规格说明书、系统设计文档、程序代码、测试用例等）归类、统计会发现，**规格说明书**是软件缺陷出现最多的地方。
- 2、质量管理体系中使用的文件类型：质量手册、**质量计划**、规范、指南、图样、作业指导书、程序、记录。
- 3、质量控制的工具和技术包含检查、控制图表、**排列图**、统计取样、流程图、趋势分析。
- 4、在质量控制、质量保证和质量管理体系基础之上建立**质量管理体系**，在质量方针指导下，质量管理指挥和控制组织的质量活动，协调质量的各项工作，包括质量控制、质量保证、全面质量管理和质量改进。
- 5、审核的目的：评价质量管理体系的**符合程度**，评价质量管理体系的有效性，识别改进的机会。

简答题

1、 关于软件质量标准，在软件开发事件中起什么作用，具体怎么实施。

根据 PPT 内容，软件质量标准在软件开发中扮演着“有章可循”的指导性角色，其核心作用是为软件质量管理工作提供一个统一的、公认的框架和准则，从而提高软件产品的质量、可靠性和可维护性。

具体作用如下（PPT 第 3 章，第 14-29 页）：

1. 提供统一框架和语言：标准为项目团队、管理者和客户之间提供了共同的语言和期望基准，减少了因理解不一造成的混乱和返工。
2. 指导质量管理活动：标准详细规定了在软件开发、供应和维护等各个阶段需要进行的质量活动，如 ISO 9001 的 20 条要求和 IEEE 系列标准，覆盖了从管理职责、设计控制到评审、测试的方方面面。
3. 保证过程和产品的一致性：通过遵循标准，可以确保开发过程是可控和可重复的，最终产出的软件产品（包括文档和代码）也符合预定的规格和质量要求。
4. 作为评估和改进的依据：标准为质量审计（Audit）和评审（Review）提供了客观的衡量尺度，便于发现过程中的不符合项，并为持续的质量改进指明方向。

具体实施方法如下：

1. 选择与分层（PPT 3-15, 3-16）：
 - 理解标准层次：首先需要识别项目所处的环境，并确定应遵循的标准层次。PPT 将标准分为国际标准、国家标准、行业标准、企业规范、项目规范五个级别。通常，一个组织会基于国际或行业标准，制定自己的企业级规范。
 - 选择合适的标准体系：根据项目类型和客户要求，选择一个或多个具体的标准体系作为指导，例如：
 - 面向过程管理的 ISO 9000 族标准。
 - 面向软件工程具体实践的 IEEE 系列标准（如 IEEE 730 SQA 流程，IEEE 1012 V&V）。
 - 面向过程能力成熟度改进的 CMMI 质量框架。
2. 融入开发流程（PPT 3-11, 3-21）：
 - 将标准的要求具体化为企业或项目的开发流程、操作规范、模板和检查表。例如，依据 IEEE 730 标准来定义组织内的 SQA 流程，明确启动、规划、控制和执行等阶段的任务。
 - 在项目计划阶段，就要将遵循标准所需的工作（如评审、审计、文档编写）纳入计划，并分配资源。
3. 执行与监控（PPT 3-25, 3-26）：
 - 在开发过程中，严格执行标准化的流程。例如，进行管理评审来监控进展是否与需求相符；进行技术评审（Technical Review）来评估软件产品是否满足规格说明；进行审查（Inspection）来系统地检查文档和代码。
 - SQA 人员（质量保证人员）根据标准对项目进行审计和监督，确保各项活动都“有章可循”。
4. 验证与改进（PPT 3-23）：
 - 通过**验证（Verification）和确认（Validation）**活动，检查开发产品是否符合标准要求，以及最终产品是否满足用户需求。
 - 将执行过程中发现的问题和偏差记录下来，作为未来改进标准或流程的依据，形成一个持续改进的闭环。

QA、QC、QM 的概念及区别

- QM (质量管理)：是一个组织在质量方面指挥和控制的协调活动，它包括了制定质量方针、目标以及质量策划、质量控制、质量保证和质量改进等活动。PPT 第 2 章《软件质量管理》和第 3 章《软件质量工程体系》整体上描述的就是 QM 的内容。

- QC (质量控制): 根据 PPT 第 4 章, QC 是质量管理的一部分, 其活动主要是检查和验证工作成果是否符合预定的标准和要求。它关注的是**“结果”, 通过测试、评审等手段发现缺陷**。
- QA (质量保证): 根据 PPT 第 5 章, QA 也是质量管理的一部分, 其活动主要是建立和维护一套流程和标准来预防缺陷的产生。它关注的是**“过程”, 通过审计、过程改进等活动, 确保开发过程是正确的, 从而为产品质量提供信任**。

区别:

- 范围: QM > QA > QC。QM 是全面的管理体系, QA 侧重过程预防, QC 侧重产品检验。
- 目的: QC 的目的是发现错误; QA 的目的是预防错误; QM 的目的是通过策划、控制、保证和改进来系统地达成质量目标。
- 时间点: QC 是实时的、面向具体产出的检验; QA 是阶段性的、面向过程的审计和监控。

3、从 CMMI 的角度谈谈软件质量控制的主要目标及主要活动。

根据 PPT 第 3 章第 28 页的《3.4.7 CMMI 质量框架》, CMMI 为软件质量控制提供了一个结构化的框架。其主要目标和活动如下:

主要目标:

CMMI 框架下软件质量控制的主要目标是: 确保软件产品及其开发过程符合既定的标准、流程和需求, 通过一系列可度量、可验证的活动, 系统性地保证和提升软件质量, 并使整个过程对所有干系人可见和一致。

主要活动:

PPT 中明确列出了以下 8 项关键活动, 这些活动构成了 CMMI 框架下质量控制的核心实践:

1. 提出软件质量需求: 在项目初期就明确定义产品的质量属性和验收标准。
2. 确定开发方案: 规划如何通过具体的开发活动来达成这些质量需求。
3. 阶段评审: 在开发生命周期的关键节点(如需求、设计、编码后)进行正式评审, 以确保各阶段的产出物符合要求。
4. 测试管理: 对测试活动进行系统化的规划、执行和监控, 确保测试的覆盖率和有效性。
5. 文档化管理: 确保所有的需求、设计、计划和报告都得到规范化的管理, 保持版本一致和可追溯。
6. 验证产品与相应文档和标准的一致性: 这是质量控制的核心, 即通过检查、测试等手段, 确认最终产品与最初定义的需求、设计和标准是完全吻合的。
7. 建立测量机制: 定义和收集质量度量数据(如缺陷密度、评审效率等), 用于量化地评估质量状态。
8. 记录并生成报告: 将所有的质量控制活动、发现的问题和度量结果记录在案, 并生成报告, 供项目团队和管理者决策。

4、请解释因果图的作用及其在项目质量管理中的应用。

根据 PPT 第 4 章第 39、40 页内容, 因果图(又称鱼骨图或石川图)是项目质量管理中一个非常重要的分析工具。

作用:

1. 系统性地识别潜在原因: 因果图的主要作用是帮助团队系统、全面地识别导致某个问题(结果)的所有潜在原因。它通过图形化的方式, 将复杂问题分解为若干个主要类别, 再在各类别下寻找具体原因, 防止遗漏。
2. 结构化地组织和展示原因: 它将问题与原因之间的逻辑关系以“鱼骨”的结构清晰地展示出来。“鱼头”代表“结果/问题”, “大骨”代表原因的主要分类, “小骨”则代表具体的原因。这种结构有助于团队成员聚焦思考和深入分析。
3. 促进团队沟通和达成共识: 在绘制因果图的过程中, 通常采用头脑风暴法, 团队成员可以从不同角度提出看法, 这有助于促进沟通, 并最终对问题的根本原因达成共识。

在项目质量管理中的应用：

因果图在项目质量管理中被广泛应用于根本原因分析（Root Cause Analysis），特别是在处理已发生的质量问题时。

应用步骤如下：

1. 确定分析的问题（鱼头）：首先，明确要分析的质量问题，例如：“软件发布后缺陷率过高”、“用户界面体验差”或“项目延期”。
 2. 确定主要原因分类（大骨）：根据问题的性质，选择或定义几个主要的原因类别。PPT 中提到了经典的 6M 分类法，这在制造业和软件项目管理中都非常适用：
 - 人（Manpower）：开发人员技能、团队沟通、责任心等。
 - 机器（Machine）：开发/测试设备性能、软件工具等。
 - 材料（Material）：需求文档、设计文档、第三方组件等。
 - 方法（Method）：开发流程、编码规范、测试策略等。
 - 测量（Measurement）：质量度量方法、测试用例的有效性等。
 - 环境（Mother-nature）：工作环境、项目压力、市场变化等。
 3. 进行头脑风暴（长出小骨）：团队成员针对每个大类，brainstorm 可能导致问题的具体原因，并将其作为小骨画在对应的大骨上。可以继续追问“为什么”，进行更深层次的分析（类似 5Why 方法）。
 4. 分析和确定关键原因：绘制完成后，团队对图上的所有原因进行讨论和评估，找出最可能导致问题的几个根本原因。
 5. 制定改进措施：针对找到的根本原因，制定相应的纠正或预防措施。
-

5、请简述什么是软件质量，并列举至少两个影响软件质量的因素。

软件质量的定义：

根据 PPT 第 1 章第 22 页的内容，软件质量可以从多个维度进行定义，综合来说：

软件质量是指一个软件产品或系统满足其明确规定（explicit）和隐含（implicit）需求的程度。它不仅指软件没有错误（Bug Free），更是一系列与软件功能和性能相关的特征和特性的集合。

影响软件质量的因素：

PPT 在多个章节（如第 1 章第 35-36 页，第 9 章第 15 页）都提到了影响软件质量的因素，以下列举并解释其中几个重要因素：

1. 需求的不确定性和复杂性：
 - 描述：这是软件缺陷最主要的来源之一。如果需求本身不清晰、不完整、存在矛盾或者在开发过程中频繁变更，那么无论后续的设计和编码多么完美，最终的软件产品质量也很难得到保证。
 - 来源：PPT 第 1 章第 36 页明确指出“需求不清晰、系统结构非常复杂”是造成软件问题的原因之一。
2. 开发人员的素质和团队合作：
 - 描述：开发人员的技能水平、经验、责任心以及团队内部的沟通效率直接影响代码质量和设计质量。一个高水平但沟通不畅的团队，其产出质量可能不如一个技术水平中等但协作顺畅的团队。
 - 来源：PPT 第 1 章第 35-36 页提到“开发人员的疲劳、压力”、“缺乏足够的知识、技能和经验”、“沟通不够、不流畅”等都是缺陷产生的原因。第 9 章第 15 页也列出了“开发人员素质”。
3. 软件项目的管理：
 - 描述：项目管理对质量有巨大影响。例如，不切实际的项目期限会给开发团队带来巨大压力，导致为了赶进度而牺牲质量；缺乏有效的风险管理，可能使潜在的质量问题最终演变成重大缺陷。

- 来源：PPT 第 1 章第 36 页提到“质量文化的影响、开发周期短、开发流程不够完善、风险估计不足等”。

4. 软件的规模和复杂度：

- 描述：软件的规模越大，其内部的逻辑关系和模块交互就越复杂。复杂性的增加会指数级地增加潜在的缺陷数量和测试难度，从而对质量构成巨大挑战。
- 来源：PPT 第 9 章第 15 页将“软件规模”列为影响软件可靠性（质量的一个重要方面）的重要因素。

名词解释

1、 软件配置 (Software Configuration)

根据 PPT 第 7 章第 4 页：

软件配置是说明软件组成的一种术语，指在软件开发过程中，构成软件产品的各种文档、程序及其数据的优化组合。它包括了所有在软件过程中产生的信息，这些信息总称为**软件配置项 (Configuration Items)**，主要涵盖三个方面：

- **计算机程序**（如源代码和可执行文件）。
 - **描述程序的文档**（如需求文档、设计文档、用户手册）。
 - **数据**（包括程序内部和外部的数据）。
-

2、基线 (Baseline)

根据 PPT 第 7 章第 6 页：

- **概念**：基线是“已经正式通过复审核批准的某规约或产品”。它是在软件开发生命周期各阶段末尾建立的一个稳定、正式的参考点或里程碑。
 - **作用**：
 1. 作为进一步开发和修改的基准和出发点。
 2. 进入基线后，对配置项的变更将受到严格的控制。
 3. 提供重现性、可追踪性和版本隔离的能力，是软件配置管理中的核心概念（第 7 章第 21 页）。
-

3、 间隔尺度 (Interval Scale)

根据 PPT 第 8 章第 5 页：

间隔尺度，也称为**定距尺度（定距）**，是一种度量尺度。它的特点是：通过数值来表示两个邻近测量点之间的差异，但没有绝对的“零”值。

在这种尺度下，数据可以进行加减运算，但不可以进行乘除运算。例如，温度（摄氏度或华氏度）就是一个典型的间隔尺度，我们可以说 20 度比 10 度高 10 度，但不能说 20 度的温度是 10 度的两倍热。

4、 软件评审 (Software Review)

根据 PPT 第 3 章第 25 页及第 6 章第 3 页：

软件评审是对软件元素（如文档、代码、计划等）或者项目状态的一种评估手段，其目的是**确定被评审对象是否与计划、标准和需求保持一致，并促使其得到改进**。

从实践角度看，软件评审最重要的目的就是**通过系统性的检查，尽早地在软件产品中发现缺陷**，从而将昂贵的、后期的返工成本，转变为成本较低的前期缺陷发现。

5、 PDCA 控制法 (PDCA Control Method)

根据 PPT 第 4 章第 21 页和第 22 页：

PDCA 控制法，又称**戴明环（Deming Cycle）**，是一个用于质量持续改进的、动态循环的管理方法。它由四个阶段组成：

- **P (Plan - 计划)**：确定目标，分析问题，并制定达成目标的实施计划。回答“做什么，打算怎么做”。
- **D (Do - 执行)**：按照制定的计划具体实施和操作。
- **C (Check - 检查)**：检查执行过程和结果，看是否达到了预期的目标，找出实际与计划之间的差距。
- **A (Act - 处理/行动)**：对检查的结果进行处理。成功的经验要加以肯定并标准化；失败的教训要总结，并将未解决的问题转入下一个 PDCA 循环。回答“下一次怎么改进”。

PDCA 的核心思想是**不断循环、持续改进、阶梯式上升**。

6、 变更控制 (Change Control)

根据 PPT 第 7 章第 22 页和第 23 页：

变更控制是一个**正式的、系统化的过程**，用于管理和控制对已建立的**基线**或软件配置项所做的任何修改。当软件的某个部分（如需求、设计或代码）在被基线化之后需要改变时，必须启动变更控制流程。

这个过程主要包括以下几个核心活动：

1. **提交变更请求**。
2. **评估变更请求**（分析其影响、成本和必要性）。
3. **决策**（批准或拒绝变更）。
4. **实现已批准的变更**。
5. **验证变更的正确性**。
6. **完成并关闭变更请求**，更新相关文档。

其目的是确保所有变更都是经过深思熟虑、受控且可追溯的，防止随意修改对项目造成混乱和质量下降。

7、 软件质量 (Software Quality)

根据 PPT 第 1 章第 22 页：

软件质量是一个多维度的概念，其核心是**软件产品满足规定的和隐含的与需求能力有关的全部特征和特性**。它不仅仅是指软件没有错误，更是多个质量属性的综合体现。

PPT 中从以下几个关键视角对其进行了定义：

- **ANSI/IEEE STD729 定义**：软件满足用户要求的程度、软件各种属性的组合程度、以及用户对产品的综合反映程度。
- **SEI 的 Watts Humphrey 观点**：在实用性、需求、可靠性和可维护性这几个方面达到一致和优秀的水准。
- **核心准则**：最终体现在四个方面：
 1. **客户满意度**：使最终产品最大限度满足客户需求。
 2. **一致性准则**：各阶段工作产品与上一阶段保持一致。
 3. **软件质量度量**：通过设立指标体系来度量产品质量。
 4. **过程质量观**：软件的质量由其开发过程的质量决定。

应用题

1、一个信息系统有 500 个模块，其中有 20 个控制模块，240 个模块其功能依赖于前导处理，系统处理大约 100 个数

据对象，每个对象平均有 3 个属性，有 150 个独特的数据库条目和 20 个不同的数据库，300 个模块有单一的入口和出口点，请用功能点方法计算软件规模。

2、请至少用两种方法计算出如下程序片段的圈复杂度。

```
public void process(int x, int y) {  
    if(x>0){  
        System.out.println("Positive");  
    }else{  
        System.out.println(" Non-positive");  
    }  
    for(int i=0;i<y;i++){  
        if(i%2==0){  
            System.out.printtln("Even");  
        }else{  
            System.out.println("Odd");  
        }  
    }  
}
```