1. **谈谈软件全面质量管理的思想体系。**

全面质量管理(Total Quality Management, TQM)被认为是一种全员、全过程、全企业的品质经营。它指一个组织以质量为中心，以全员参与为基础,目的在于通过让顾客满意和本组织所有成员及社会受益而达到永续经营的管埋途径。全面质量管理通常包括以下4个步骤。

1. 第1步是指一个连续的过程改进系统。其目标在于开发一个看得见的、可重复的和可度量的软件过程。
2. 第2步只有在第1步完成之后才可启动。这一步将检查影响过程的其他因素，并优化这些因素对过程的影响。例如,软件过程可能受到高层职员流动的影响，而该影响则是由于公司内部不断重组而引起的。一个稳定的公司组织会对软件质量的提高带来很大的帮助,所以第2步可以帮助管理者对公司重组方式提出建议。
3. 前面两个步骤关注的是过程，第3步(或称为“第5感觉”)则关注软件产品的用户，它是通过检查用户使用产品的方式，而促使产品本身的改进和潜在地改进产品的生产过程。
4. 第4步将管理者的注意力从当前的产品上移开并拓宽。作为一个面向商业的步骤，该步骤通过观察产品的市场用途,来寻找产品在相关领域中的发展机会。在软件领域，第4步可以被视为一种发现有利可图的新产品,或视做寻找当前计算机系统的副产品用途的努力。

对于大多数公司而言,都应该关心第1步。在建立一个成熟的软件过程之前公司进人后面步骤的意义不大。

具体来讲,全面质量管理蕴涵着如下含义。

1. 强烈关注顾客。从现在和未来的角度看,顾客已成为企业的衣食父母。“以顾客为中心”的管理模式正逐渐受到企业的高度重视。全面质量管理注重顾客价值,其主导思想就是“顾客的满意和认同是长期赢得市场，创造价值的关键”。因此,全面质量管理要求必须把以顾客为中心的思想贯穿到企业业务流程的管理中,即从市场调查、产品设计、试制、生产、检验、仓储、销售,以及到售后服务的各个环节都应该牢固树立“顾客第一”的思想。不但要生产物美价廉的产品，而且要为顾客做好服务工作,最终让顾客放心满意。
2. 精确度量。全面质量管理采用统计度量组织作业中人的每一个关键变量,然后与基准进行比较来发现问题,从而追踪问题的根源,达到消除问题、提高品质的目的。
3. 坚持不断地改进。全面质量管理是一种水远不能满足的承诺，“非常好”还不够,质量总能得到改进。在“没有最好，只有更好”观念的指导下,企业持续不断地改进产品或服务的质量与可靠性,以确保企业获取差异化的竞争优势。
4. 向员工授权。全面质量管理吸收生产线上的工人加人改进过程,广泛地采用团队形式作为授权的载体,依靠团队发现和解决问题。
5. 改进组织中每项工作的质量。全面质量管理采用广义的质量定义。它不仅与最终产品有关,并且还与组织如何交货、如何迅速地响应顾客的投诉、如何为客户提供更好的售后服务等都有关系。
6. **谈谈6σ在软件设计和编程活动中的一些具体实践。**

20世纪90年代中期，6σ开始被GE(通用电气公司，是全球最大的跨行业经营的科技、制造和服务型企业之一)从一种全面质量管理方法演变成为一种高度有 效的企业流程设计、改善和优化的技术,并提供了一系列同等地适用手设计、生产和服务的新产品开发工具。继而与GE的全球化、服务化、电子商务等战略齐头并进，成为全世界追求管理卓越性的企业最为重要的战略举措。在此之后,6σ逐步发展成为以顾客为主体来确定企业战略目标和产品开发设计的标尺,追求持续进步的一种管理析学。

6σ管理是总结了全面质量管理的成功经验.提炼了其中流程管理技巧的精华和最行之有效的方法，成为一种提高企业业绩与竞争力的管理模式。该管理法在Motorola、GE、Dell、惠普、西门子、索尼、东芝等众多跨国企业中的实践证明是卓有成效的。

1. **什么是软件评审？为什么需要进行软件评审？**
2. 软件评审:评审是一些用于开发过程早期检查和纠正缺陷的有效方法。它们可以用来检查未形成执行代码的文档的缺陷。
3. 在开发过程中，评审可以让我们获得以下收益
   1. 提高项目的生产率。这是由于早期发现了错误，因而减少了返工时间，还可能减少测试时间。
   2. 改善软件的质量。
   3. 在评审过程中，使开发团队的其他成员更熟悉产品和开发过程。
   4. 通过评审，标志着软件开发的-一个阶段的完成。
   5. 生产出更容易维护的软件。主要原因是:对于被评审的软件，评审者必须是非常熟悉的；同时，在评审过程中，一定会产生并利用很多证明文档，于是评审就迫使开发者产生出许多有用的文档，而这些文档如果不是因为评审，则在整个项目期间可能都不会生产。此外，评审过程也将增加对所开发软件的理解。
4. **软件评审包括哪些内容？**
5. 管理评审

一个组织之所以需要管理，是为了能更好地进步和发展。为了达到这个目的,通常需要对原来的发展状况进行回顾,分析并总结出存在的问题和改进的措施。这也就是进行管理评审的原因。

管理评审就是最高管理者为评价管理体系的适宜性、充分性和有效性所进行的活动。管理评审的主要内容是组织的最高管理者就管理体系的现状、适宜性、充分性和有效性以及方针和日标的贯彻落实及实现情况进行正式的评价,其目的就是通过这种评价活动来总结管理体系的业绩,并从当前业绩上考虑找出与预期目标的差距,同时还应考虑任何可能改进的机会，并在研究分析的基础上,对组织在市场中所处地位及竞争对手的业绩了以评价,从

而找出自身的改进方向。

1. 技术评审

技术评审(Technical Review)是一种同行审查技术。其主要特点是由一组评审者按照规范的步骤对软件需求、设计、 代码或其他技术文档进行仔细检查,以找出和消除其中的缺陷。技术评审的目的是确保需求说明、设计说明书与最初的说明书保持一致， 并按照计划对软件进行了正确的开发。技术评审后,需要以书面形式对评审结果进行总结。技术评审会分为正式和非正式两种,通常由技术负责人(技术骨干)制定详细的评审计划,包括评审时间、地点以及定义所需的输入文件。

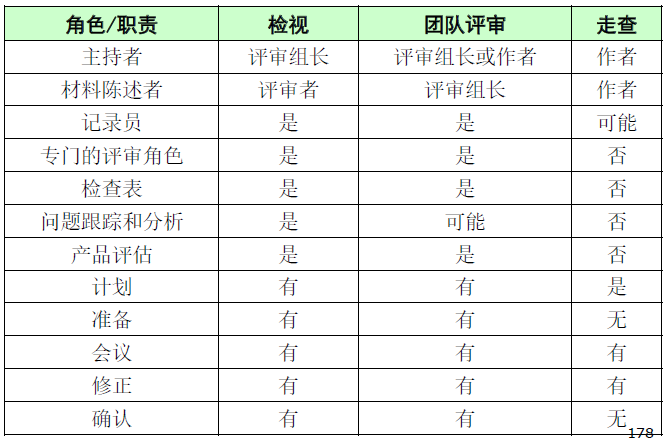
1. 文档评审

在软件开发的每个阶段,对该阶段所形成的文档进行评审,尽早发现问题,并及时采取措施予以解决，确保文档的内容准确,为软件产品的质量提供保障。

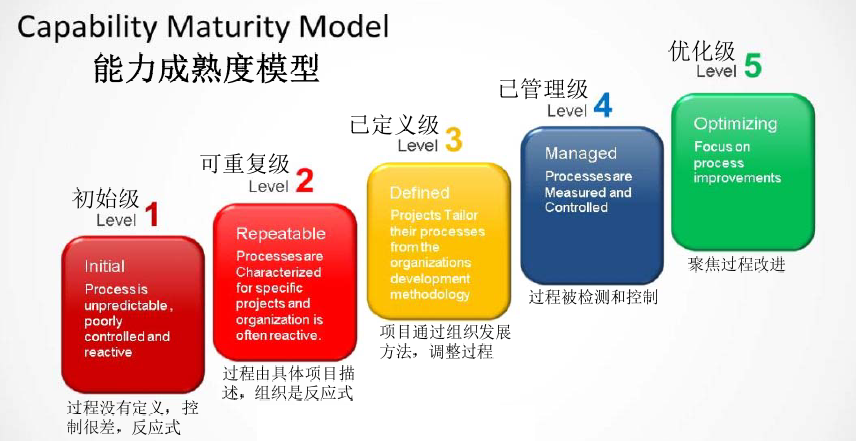
1. 过程评审

过程评审是对软件开发过程的评审,其主要任务是通过对流程的监控，保证SQA组织定义的软件过程在项月中得到了遵循，同时保证质量保证方针能更快更好地执行。过程评审的评审对象是质量保证流程,而不是针对产品质量或者其他形式的工作产出。

1. **软件评审主要有哪些方法？他们的异同点是什么？**
2. 方法：
   1. 特别检查
   2. 轮查
   3. 走查
   4. 团队评审
   5. 检视
3. 异同点：



1. **简述CMM的概念和等级，以及CMM与CMMI的关系**
2. 概念：能力成熟度模型(CMM)的本质是软件管理T.程的一个部分。它是对于软件组织在定义,实现,度量,控制和改善其软件过程的进程中各个发展阶段的描述。它通过5个不断进化的层次来评定软件生产的历史与现状。
3. 等级：
4. 初始级：过程没有定义，控制很差，反应式
5. 可重复级：过程由具体项目而定，组织是反应式
6. 已定义级：项目通过组织发展方法，调整过程
7. 已管理级：过程被检测和监控
8. 优化级：聚焦过程改进



1. CMM与CMMI的关系
2. 软件能力成熟度集成模型(CMMI)是CMM模型的最新版本。早期的能力成熟度模型是一种单一的模型,其英文缩写为CMM,较多地用于软件工程。随着应用的推广与模型本身的发展,该方法演绎成为一-种被广泛应用的综合性模型,因此改名为CMMI模型。
3. CMMI与CMM最大的不同点在于: CMMISM-SE/SW/PPD/SS 1.1 版本有4个集成成分,即系统工程(SE)和软件工程(SW)是基本的科目,对子有些组织还可以应用集成产品和过程开发方面(IPPD)的内容，如果涉及供应商外包管理可以相应地应用SS(Supplier Sourcing)部分。
4. **谈谈软件质量标准之间的关系及其优缺点。**
5. 关系：根据软件工程标准制定机构和标准适用的范围,将软件质量标准分为5个级别，即国际标准、国家标准、行业标准、企业标准和项目规范。很多标准的原始状态可能是项目标准或企业标准,但随着行业发展与推进,它的权威性可能促使它发展成为行业、国家或国际标准，因此这里所说的层次具有一定的相对性。
6. 优缺点；
7. 国际标准：由国际机构指定和公布供各国参考的标准称为国际标准。国际标准化组织(ISO)具有广泛的代表性和权威性,它所公布的标准也具有国际影响力。
8. 国家标准：由政府或国家级的机构制定或批准，适用于本国范围的标准,称为国家标准。
9. 行业标准：行业标准是由一些行业机构、学术团体或国防机构制定,并适用于某个业务领域的标准。
10. 企业标准：一些大型企业或公司,由于软件工程工作的需要，制定适用于本部门的规范。
11. 项目规范：项目规范是为一些科研生产项目需要而由组织制定一些具体项目的操作规范,此种规范制定的目标很明确,即为该项任务专用。项目规范虽然最初的使用范围小,但如果它能成功指导一个项目的成功运行并重复使用，也有可能发展为行业规范。
12. **简述论述软件可靠性和硬件可靠性的区别**

软件和硬件在可靠性特征上的差异，主要有以下几点:

1. 最明显的是硬件有老化损耗现象，硬件失效是物理故障，是器件物理变化的必然结果，有浴盆曲线现象；软件不发生变化，没有磨损现象，有陈旧落后的问题，没有浴盆曲线现象。
2. 硬件可靠性的决定因素是时间，受设计、生产、运用的所有过程影响，软件可靠性的决定因素是与输入数据有关的软件差错，是输入数据和程序内部状态的函数，更多地决定于人。
3. 硬件的纠错维护可通过修复或更换失效的系统重新恢复功能，软件只有通过重设计。
4. 对硬件可采用预防性维护技术预防故障，采用断开失效部件的办法诊断故障，而软件则不能采用这些技术。
5. 事先估计可靠性测试和可靠性的逐步增长等技术对软件和硬件有不同的意义。
6. 为提高硬件可靠性可采用冗余技术，而同一软件的冗余不能提高可靠性。
7. 硬件可靠性检验方法已建立，并已标准化且有一整套完整的理论，而软件可靠性验证方法仍未建立，更没有完整的理论体系。
8. 硬件可靠性已有成熟的产品市场，而软件产品市场还很新。
9. 软件错误是永恒的，可重现的，而一些瞬间的硬件错误可能会被误认为是软件错误。
10. **简述主要的软件可靠性参数**
11. ETF:在软件中的固有差错总数。这个数假设是固定的和有限的。
12. ETV:在软件中的固有差错总数。这个数假设是变化的,因为在开发或维护过程中,随时有可能将-一个新的差错引入到编码中。
13. EC(t):在某一时刻t已纠正的差错总数,或在使用或测试t时间后已纠正的差错总数。
14. ED(t):在某一时刻t已检测出的差错总数，或在使用或测试t时间后已检测出的差错总数。
15. P:在差错纠正活动期间所发生的测试循环次数或测试间隔数。
16. EC(p):直到第p次测试循环才被纠正的差错数。它不包括在第p次循环期间纠正的差错。
17. θ:差错加速度，或故障率的变化。
18. t:累积到当前时间或累积到当前差错数的测试或运行时间。
19. τ:累积执行时间。
20. k:常系数。
21. λo:起始故障率。
22. λp:当前故障率。
23. a:增长率。
24. N:测试用例运行的总数。
25. S:成功的测试用例运行的总数。
26. **论述提高软件可靠性的方法和技术**
27. 建立以可靠性为核心的质量标准
28. 在软件项目规划和需求分析阶段就要建立以可靠性为核心的质量标准。这个质量标准包括实现的功能、可靠性、可维护性、可移植性、安全性、吞吐率等等，
29. 虽然还没有一个衡量软件质量的完整体系，但还是可以通过一定的指标来指定标准基线。
30. 软件质量从构成因素上可分为产品质量和过程质量。
31. 确定划分的各开发过程的质量度量：需求分析质量度量、设计结果质量度量、测试结果质量度量、验收结果质量度量。
32. 选择开发方法
33. 软件开发方法对软件的可靠性也有重要影响。
34. 目前的软件开发方法主要有Parnas方法、Yourdon方法、面向数据结构的Jackson方法和Warnier方法、PSL/PSA方法、原型化方法、面向对象方法、可视化方法、ICASE方法、瑞理开发方法等，其他还有BSP方法、CSF方法等。这里特别要提一下的是Parnas方法。
35. 软件重用

软件重用不仅仅是指软件本身，也可以是软件的开发思想方法、文档，甚至环境、数据等，包括三个方面内容的重用：

1. 开发过程重用，指开发规范、各种开发方法、工具和标准等。
2. 软件构件重用，指文档、程序和数据等。
3. 知识重用，如相关领域专业知识的重用。
4. 使用开发管理工具

开发一个大的软件系统，离不开开发管理工具，作为一个项目管理员，仅仅靠人来管理是不够的，需要有开发管理工具来辅助解决开发过程中遇到的各种各样的问题，以提高开发效率和产品质量。

如Intersolv公司的PVCS软件开发管理工具，在美国市场占有率已超过70％，使用PVCS可以带来不少好处：

规范开发过程，缩短开发周期，减少开发成本，降低项目投资风险；自动创造完整的文档，便于软件维护；管理软件多重版本；管理和追踪开发过程中危及软件质量和影响开发周期的缺陷和变化，便于软件重用，避免数据丢失，也便于开发人员的交流，对提高软件可靠性，保证质量有很大作用。

1. 加强测试
2. 测试规范包括以下三类文档：

测试设计规范：详细描述测试方法，规定该设计及其有关测试所包括的特性。还应规定完成测试所需的测试用例和测试规程，规定特性的通过/失败判定准则。

测试用例规范：列出用于输入的具体值及预期输出结果。规定在使用具体测试用例时对测试规程的各种限制。

测试规程规范：规定对于运行该系统和执行指定的测试用例来实现有关测试所要求的所有步骤。

1. 测试的方法多种多样：

走查（Walk-through），即手工执行，由不同的程序员（非该模块设计者）读代码，并进行评论。

机器测试，对给定的输入不会产生不合逻辑的输出。

程序证明或交替程序表示。

模拟测试，模拟硬件、I/O设备等。

设计审查，关于设计的所有各方面的小组讨论会，利用所获得的信息，找出缺陷及违反标准的地方等。

1. 容错设计

提高可靠性的技术一般可以分为两类，一类是避免故障，在开发过程中，尽可能不让差错和缺陷潜入软件，这类常用的技术有：

算法模型化，把可以保证正确实现需求规格的算法模型化。

模拟模型化，为了保证在确定的资源条件下的预测性能的发挥，使软件运行时间、内存使用量及控制执行模型化。

可靠性模型，使用可靠性模型，从差错发生频度出发，预测可靠性。

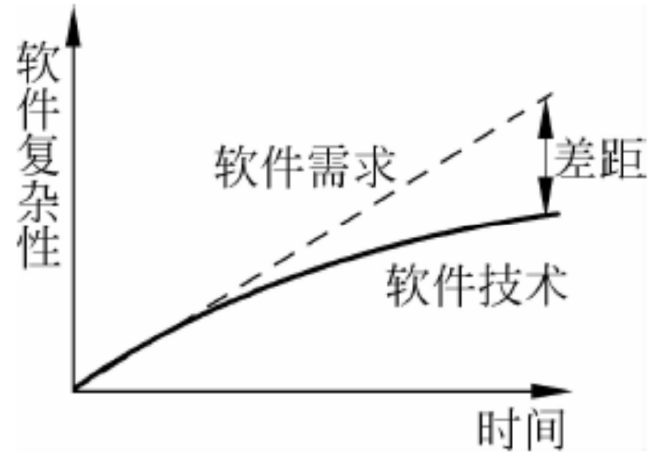
正确性证明，使用形式符号及数学归纳法等证明算法的正确性。

软件危险分析与故障树分析：从设计或编码的结构出发，追踪软件开发过程中潜入系统缺陷的原因。

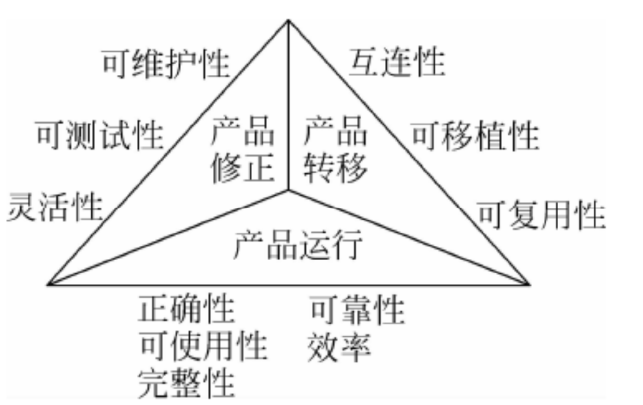
分布接口需求规格说明：在设计的各阶段使用形式的接口需求规格说明，以便验证需求的分布接口实现可能性与完备性。

1. **简述影响软件质量的因素**

软件业通过多年的实践，总结出软件质量是人、过程和技术的函数，即Q={M，P，T}。其中，Q表示软件质量，M表示人，P表示过程，T表示技术。



1. **论述几种常见的质量保证模型**
2. McCall模型
3. 正确性：一个程序满足她的需求规约和实现用户任务目标的程度。
4. 可靠性：一个程序满足一所需的精确度完成它的预期功能的程度
5. 效率：一个程序完成其功能所需的计算资源和代码的度量。
6. 完整性：对未授权人员访问软件或数据的可控制程度。
7. 可用性：学习、操作、准备输入和解释程序输出所需的工作量。
8. 可维护性：定位和修复程序中一个错误所需的工作量。
9. 灵活性：修改一个运行的程序所需的工作量。
10. 可测试性：测试一个程序以确保她完成所期望的功能所需的工作量。
11. 可移植性：把一个程序从一个硬件和或软件系统环境移植到另一个环境所需的工作量。
12. 可复用：性一个程序可以在另外一个应用程序中复用的程度
13. 互连性：连接一个系统和另一个系统所需的工作量。



1. Boehm模型
2. Boehm模型着手于软件总体的功效，也就是说，对于一个软件系统而言，除了有用性以外，它的开发过程必定是一个时间，金钱和能量的消耗过程。考虑到系统交付时使用它的用户类型，Boehm模型从几个维来考虑软件的效用。
3. 总功效可以被分解成可移植性，有效性，可维护性。
4. 其中，有效性可以细分为可靠性，效率，运行工程可维护性可以细分为测试性，可理解性，可修改性。
5. FURPS模型
6. 功能性：通过评价特征集和程序的能力、交付的函数的通用性和整体系统的安全性来评估。
7. 可用性：通过考虑人的因素、整体美学、一致性和文档来评估。
8. 可靠性：通过度量错误的频率和严重程度、输出结果的准确度、平均失效间隔时间、从失效恢复的能力、程序可预测性等来评估。
9. 性能：通过侧度处理速度、响应时间、资源消耗、吞吐量和效率来评估。
10. 支持度：包括扩展程序的能力可扩展性、可适应性和服务性这三个属性代表了一个更一般的概念—可维护性、以及可测试性、兼容度、可配置性组织和控制软件配置的元素的能力、一个系统可以被安装的容易程度、问题可以被局部化的容易程度。
11. ISO 9126
12. 功能性

适合性、准确性、互操作性、依从性、安全性。

1. 可靠性

成熟性、容错性、可恢复性。

1. 可用性

可理解性、易学性、可操作性。

1. 效率

时间特性、资源特性。

1. 可维护性

可分析性、可改变性、稳定性、可测试性。

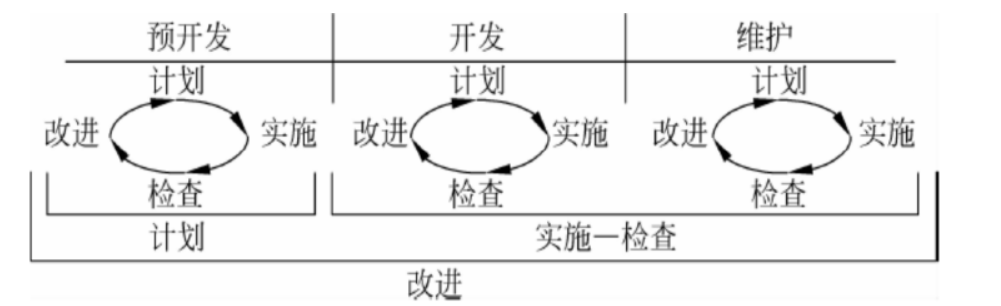
1. 可移植性

适应性、可安装性、一致性、可替换性

1. **论述软件过程度量的目标、对象、方法和结果**
2. 目标：软件过程度量的目标是为了对软件过程的行为进行目标管理,并在度量的基础上对软件过程进行控制、评价和改善。软件过程度量最终为项目管理和软件过程管理服务。
3. 对象：主要包括三个:工作产品、软件项目和过程。工作产品是指软件项目执行过程中产生的交付的和不交付的过程产品,其规模度量主要包括软件系统规模度量和软件产品文档规模度量。软件项目的度量主要集中在项目度量、成本、进度等方面。过程度量主要从组织的角度考虑，指对项目定义软件过程和组织标准软件过程的度量。
4. 方法:主要包括常用的采集方法(在不同项目阶段针对不同类型内容的数据采集)，常用的数据分析方法(多种数据表示方法和分析方法)。
5. 结果:对过程度量的结果是模型、关系和由过程量化特征组成的过程基线。
6. **论述软件配置管理关键活动**

任何关于SCM的讨论均涉及一系列复杂问题：

1. 一个组织如何标识和管理程序（及其文档）的很多现存版本，以使得变化可以高效地进行？
2. 一个组织如何在软件被发布给客户之前和之后控制变化？
3. 谁负责批准变化，并给变化确定优先级？
4. 我们如何保证变化已经被恰当地进行？
5. 采用什么机制去告知其他人员已经实行的变化？
6. **论述软件质量控制的基本概念和方法，以及实施过程**
7. 基本概念：软件质量控制是一组由开发组织使用的程序和方法，使用它可在规定的资金投入和时间限制的条件下,提供满足客户质量要求的软件产品并持续不断地改善开发过程和开发组织本身，以提高将来生产高质量软件产品的能力。
8. 方法：
9. 目标问题度量法
10. 风险管理法
11. PDCA质量控制法
12. 实施过程



1. **论述软件质量保证体系的目标**

软件质量保证的目标是以独立审查的方式，从第三方的角度监控软件开发任务的执行，就软件项目是否正确遵循已制定的计划、标准和规程给开发人员和管理层提供反映产品和过程质量的信息和数据，提高项目透明度，同时辅助软件工程取得高质量的软件产品。

1. **谈谈对软件质量保证的理解，并举例说明软件质量与测试对企业的正面影响和负面影响。**
2. 软件质量保证(SQA, Software Quality Assurance)是建立一套有计划、有系统的方法，来向管理层保证拟定出的标准、步骤、实践和方法能够正确地被所有项目所采用。
3. 举例：
4. 1994年12月30日,美国弗吉尼亚州林渠堡大学的Tomsas R. Nicely博士在他的奔腾PC上做除法实验时记录了一个没想到的结果。他把发现的问题放到因特网上，随后引发了一场风暴，成千上万的人发现了同样的问题,以及其他得出错误结果的情形。即,英特尔.奔腾浮点除法软件缺陷在计算机的“计算器”程序中输入算式: (4 195 835/3 145 727) X 3145727-4195835。如果答案足0,就说明计算机没问题。如果得出别的结果,就表示计算机使用的是带浮点除法软件缺陷的老式英特尔奔腾处理器。该软件缺陷被刻录在一个计算机芯片中，并在生产过程中反复制造。不过万幸的是.这种情况很少见,仅仅在进行精度要求很高的数学、科学和工程计算中才导致错误，而大多数进行财会管理和商务应用的用户根本不会遇到此类问题。
5. Windows的输入法漏洞

Windows2000中文简体版的输入法漏洞,可以使木地用户绕过身份验证机制进入系统内部。实验表明,Windows2000中文简体版的终端服务在远稈操作时仍然存在这一漏洞，而且危害更大。Windows2000的终端服务功能，能使系统管理员对Windows2000进行远程操作，采用的是图形界面,能使用户在远程控制计算机时功能与在本地使用一样，其默认端口为3389,用户只要装了Windows 2000的客户端连接管理器就能与开启了该服务的计算机相连。因此这.漏洞使终端服务成为Windows2000的合法木马。

1. **软件测试和软件开发的关系是怎样的？常用的软件测试方法有哪些？**
2. 软件开发是生产制造软件;软件测试是验证开发出来软件的质量。关系应该是:
3. 没有软件开发就没有测试，软件开发提供软件测试的对象。
4. 软件开发和软件测试都是软件生命周期中的重要组成部分
5. 软件开发和软件测试都是软件过程中的重要活动。
6. 软件测试是保证软件开发产物质量的重要手段。
7. 软件测试方法:
8. 静态测试和动态测试
9. 黑盒测试、白盒测试和灰盒测试
10. 基于软件开发阶段的测试方法:①需求测试②单元测试③集成测试④性能测试⑤压力测试⑥容量测试⑦配置测试⑧回归测试⑨安装测试回安全性测试。