



Shanghai Jiao Tong University

软件过程与管理

Module: Software Quality Management

软件质量管理

上海交通大学软件工程中心

质量事故案例

- ◆ 1981年，由计算机程序改变而导致1/67的时间偏差，使航天飞机上的5台计算机不能同步运行。这个错误导致了航天飞机发射失败。
- ◆ 1986年，1台Therac25机器泄露致命剂量的辐射，致使两名医院病人死亡。原因是一个软件出现了问题，导致这台机器忽略了数据校验。
- ◆ 化学银行一个晚上从10多万位顾客账户上，错误地扣除了大约1500万美元的存款。原因是一个软件的问题使ATM的每一笔业务重复记录两次。



Software Process and Management

2

沈备军

问题

- ◆ 软件系统功能齐全是不是就是质量好？
- ◆ 用户界面友好是不是就是软件的质量好？
- ◆ 没有BUG是不是就是软件的质量好？
- ◆ 什么是用户满意的软件项目？
- ◆ 软件测试是不是软件质量的全部？
- ◆ 那么，什么是软件的质量？如何保证软件的质量？

Software Process and Management

3

沈备军

本节内容

- ◆ 软件质量管理概述
- ◆ 软件评审

@第12章. 教材
@质量管理. PMBOK

Software Process and Management

4

沈备军

现代质量观对软件质量观的影响

质量是检验出来的	统计过程控制 (SPC)
质量是制造出来的	全面质量控制 (TQC)
质量是设计出来的	综合质量管理 (TQM)

- ◆ 控制的方法：更侧重于对思想方法的控制，更偏重于对设计的控制
- ◆ 质量改进的方法：CMM、ISO9001、SPICE、Six Sigma

Software Process and Management

5

沈备军

软件质量的定义

- ◆ ANSI/IEEE Std 729-1983定义 “与软件产品满足规定的和隐含的需求的能力有关的特征或特性的全体”。
- ◆ M. J. Fisher 定义 “所有描述计算机软件优秀程度的特性的组合”。

Software Process and Management

6

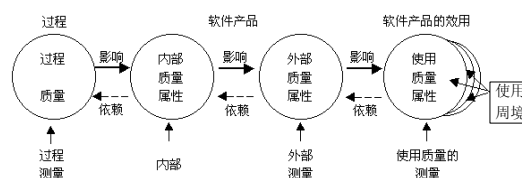
沈备军

何谓软件质量好

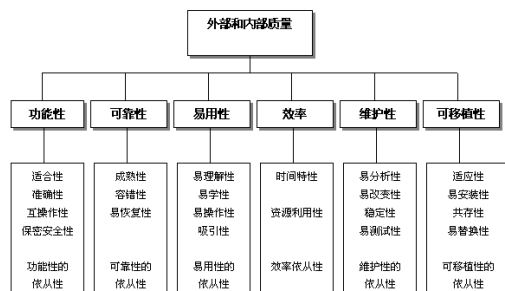
- ◆ 明确声明的功能和性能需求、明确文档化过的开发标准、以及专业人员开发的软件所应具有的所有隐含特征都得到满足。
 - 软件需求是进行质量度量的基础，与需求不符就是质量不高
 - 指定的标准定义了一组指导软件开发的准则，如果不能遵照这些准则，就极有可能导致质量不高
 - 通常有一组隐含需求是不被提及的，如易维护性，如果软件符合了明确的需求却没有满足隐含需求，软件质量仍然值得怀疑

软件的质量属性

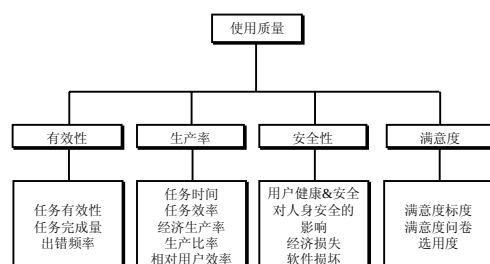
- ◆ 质量的三种视角：内部、外部、和使用质量



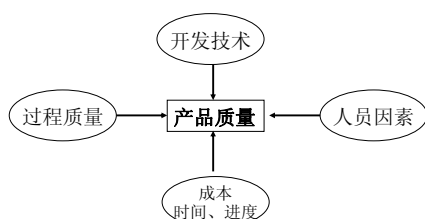
外部和内部质量模型



使用质量的模型



影响软件质量的因素



质量成本

- ◆ 预防成本：使项目的差错保持在一个可接受范围内的成本，如培训、分包商评定等
- ◆ 评估成本：评估产品或服务的成本。如产品检查、评审或测试、处理和报告测试数据
- ◆ 内部故障成本：在客户收到产品之前，纠正已识别出的一个缺陷所引起的成本
- ◆ 外部故障成本：为产品交付顾客之后发现的缺陷而支付的成本。如顾客抱怨处理、讼案、未来商务机会丧失
- ◆ 测量和测试设备成本

质量控制和管理技术

- 验证与确认 (V&V)
- 质量控制图 (Quality Control Chart)
- 鱼骨图 (Fishbone Diagram)
- Pareto 分析
- 统计抽样
- 偏差分析: 质量计划和实际执行情况的对比分析
- 趋势分析: 发现了多少缺陷, 还有多少没发现

Software Process and Management

13

沈备军

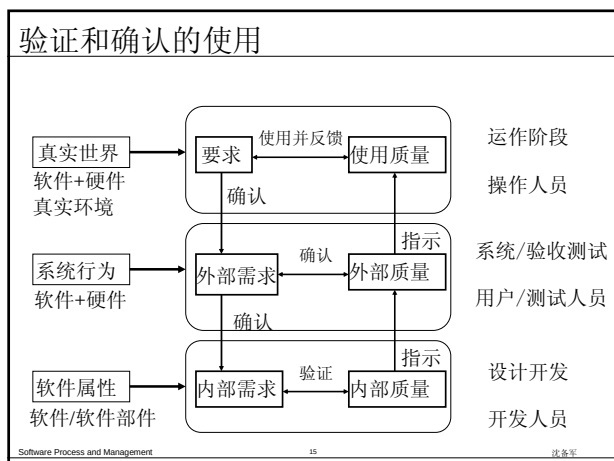
验证和确认的定义

- V&V 是一个用以分析、评价、测试系统和软件文档以及代码系统的过程, 从而尽可能地确保质量、可靠性以及系统需求和目标满意度。
[IEEE Standard Glossary]
- 验证(Verification)是“对系统或单元评价的过程, 以确定一个给定的开发阶段的产品是否满足在此阶段开始时所给定的条件”
我们是否正确地完成了产品?
- 确认(Validation)是“在软件开发过程期间或结束时评价系统或单元的过程, 以确定它是否满足给定的需求”
我们是否完成了正确的产品?

Software Process and Management

14

沈备军



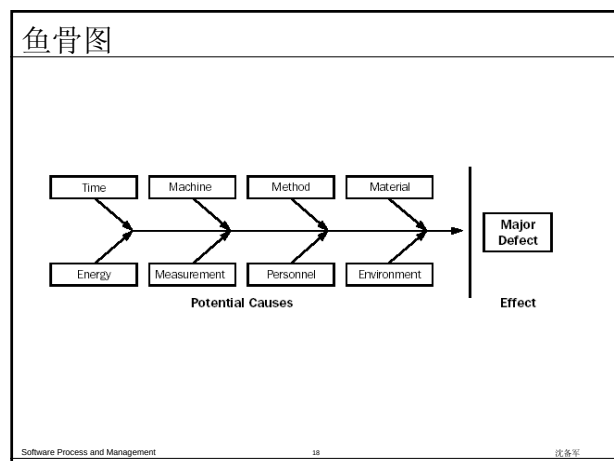
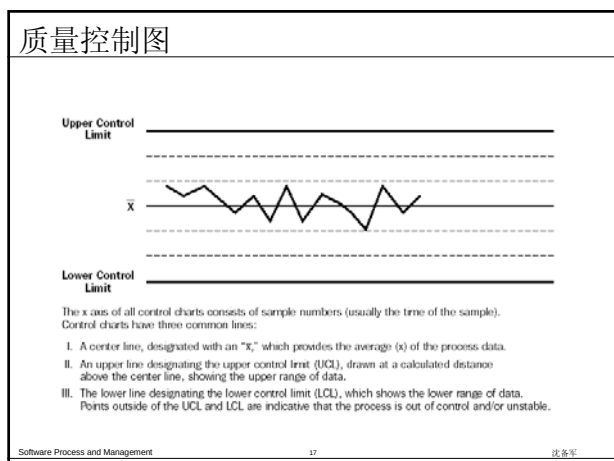
V&V 方法

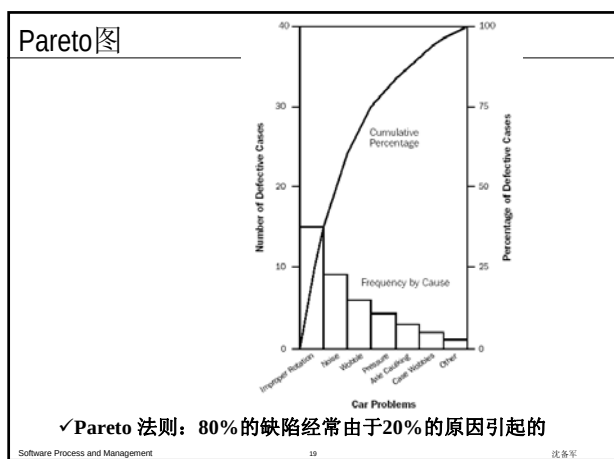
V&V方法	项目
测试	代码、最终产品
静态分析	代码
评审	中间产品
形式化证明	代码
可靠性增长模型	最终产品
模拟	原型
...	

Software Process and Management

16

沈备军





软件质量管理最佳实践经验（1）

- ◆ 从整个开发过程来保证质量，及早纠正大部分的缺陷
- ◆ 制定质量计划
- ◆ 软件复用
- ◆ 独立的QA人员
- ◆ 软件评审
- ◆ 代码走查

Software Process and Management

20

沈备军

软件质量管理最佳实践经验（2）

- ◆ 有效的软件工程技术、方法和工具
- ◆ 多层次的测试策略
- ◆ 易错模块的质量保证：20%的模块包含了80%的错误
- ◆ 对软件文档及其变更的控制
- ◆ 遵从软件开发规程
- ◆ 度量和报告机制

Software Process and Management

21

沈备军

案例讨论

- ◆ 质量问题分析



Software Process and Management

22

沈备军

本节内容

- ◆ 软件质量管理概述
- ◆ 软件评审

Software Process and Management

23

沈备军

你会有这些问题吗？

- ◆ 什么是软件评审（review）、审查（inspection）、走查（walkthrough）？
- ◆ 需要化多少时间？我的项目进度太紧了，无法安排评审。
- ◆ 如何提高评审的有效性？
- ◆ 评审过程如何进行？评审员越多越好吗？
- ◆ 可以用评审结果来评估员工的业绩吗？

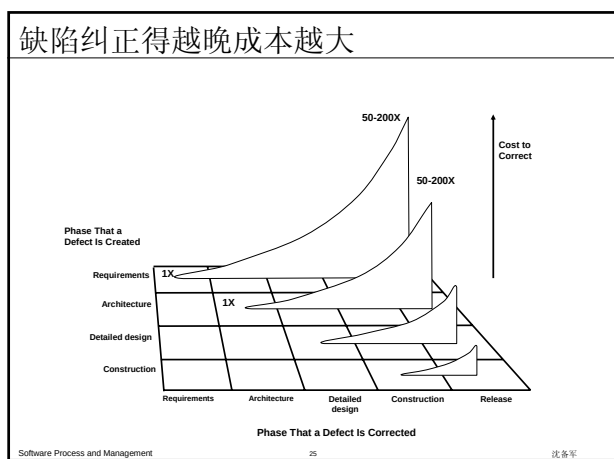


Software Process and Management

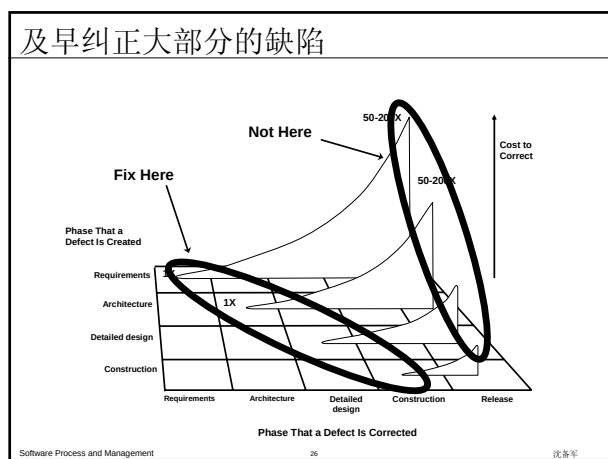
24

沈备军

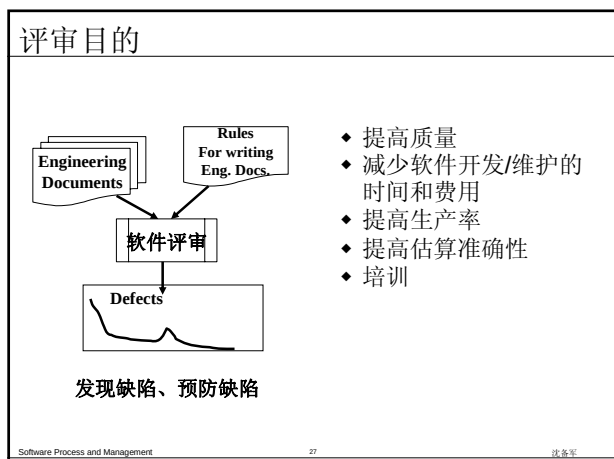
缺陷纠正得越晚成本越大



及早纠正大部分的缺陷



评审目的



评审得益

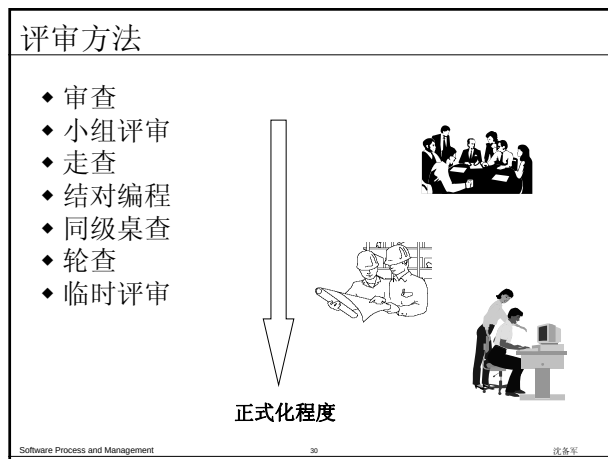
- ◆ 提高产品的质量
 - 设计阶段的评审在发现缺陷的有效性最高达到75%，比测试有效20倍以上。
 - ◆ 改进开发过程，预防缺陷发生
 - ◆ 通过减少开发后期的缺陷纠正和返工时间来加快开发进度
 - 减少50%的开发周期[Raytheon, IBM][Fagan 2.97]
 - ◆ 提高长期的软件生产率，提高1.7 倍 [Raytheon 1988-1994]
 - 300 LOC/day vs.144 LOC/day [IBM FSD 77]
 - ◆ 投资回报率从4:1 到 30:1
- Software Process and Management 28 沈备军

软件评审分类

评审类型	目的
管理评审	向上层管理者提供信息，以帮助决策
同级评审	寻找产品中的缺陷和改进契机
项目后的评审	对完成后的项目或阶段进行评审，以吸取经验
状态评审	向项目组提供最新的项目状态信息

Software Process and Management 29 沈备军

评审方法



评审方法的特点

评审类型	活 动				
	计 划	准 备	开 会	纠 错	验 证
审查	是	是	是	是	是
小组评审	是	是	是	是	否
走查	是	否	是	是	否
结对编程	是	否	持续进行	是	是
同级桌查, 轮查	否	是	可能	是	否
临时评审	否	否	是	是	否

Software Process and Management

35

沈备军

审查 Inspection

- ◆ 最系统化、最严密的、最有效的评审技术
- ◆ 严格定义的、文档化的审查过程
 - 计划、准备、开会、返工、验证
- ◆ 审查小组明确分工
 - 作者（作者不能担任审查组长、读者和记录员）
 - 审查者
 - 审查组长：主持评审会议
 - 读者：陈述产品
 - 记录员：记录下评审小组提出的问题
 - 其他审查者
- ◆ 使用缺陷检查表



Software Process and Management

32

沈备军

小组评审 Team Review

- ◆ 评审过程
 - 计划、准备、开会、返工
- ◆ 作者或评审组长主持会议
- ◆ 读者这个角色被省略了，改由评审组长询问其他评审者这一部分是否有问题
- ◆ 使用记录员
- ◆ 使用缺陷检查表

Software Process and Management

33

沈备军

走查 Walkthrough

- ◆ 评审过程
 - 计划、开会、返工
- ◆ 作者主持会议，起主导作用，陈述产品
- ◆ 常用走查方法
 - 使用一些样品数据一步步执行一个模块，和同事一道检查以确保正确的逻辑和行为。
 - 使用交互式调试器
 - 按脚本执行，脚本描述了一项具体的任务或场景，用以说明系统如何在用户会话中发挥功能

Software Process and Management

34

沈备军

审查、小组评审和走查的特性比较

特 性	审查	小组评审	走查
主持	评审组长	评审组长或作者	作者
材料陈述者	读者	评审组长询问问题	作者
所交材料的粒度	一小部分	分页或分段	根据作者的判断
使用记录员	是	是	可能
文档化的评审过程	是	可能	可能
设置专门的评审角色	是	是	否
使用缺陷检查表	是	是	否
数据收集和分析	是	可能	否
产品评估	是	是	否

Software Process and Management

35

沈备军

结对编程Pair Programming

- ◆ 极值编程 XP 中的一个实践
- ◆ 两个开发者在一个工作站上同时编写同一个程序，进行实时的、持续的、非正式的评审。司机和搭档的角色还要不时地交换。
- ◆ 由于搭档的实时评审，结对者可以迅速纠正错误。快速的迭代能使设计和程序更加强壮。
- ◆ 结对编程技术除了能应用于编码，还能应用于需求、设计、测试等文档。

Software Process and Management

36

沈备军

同级桌查 Peer Deskcheck

- ◆ 在两次编译之间仔细地检查源代码以保证程序正确执行，这就是桌查。桌查是PSP的组成部分,是一种自评审，不属于同级评审。
- ◆ 在同级桌查中，除作者外的一位评审者对工作产品进行检查。评审者可以和作者坐在一起讨论，也可以独立检查。
- ◆ 评审完成后，评审者把错误表交给作者，或者两人一起坐下来共同准备错误表，或者简单地将做过标记的工作产品交给作者。
- ◆ 要寻找一位足够专业且值得信赖的人担任评审者。

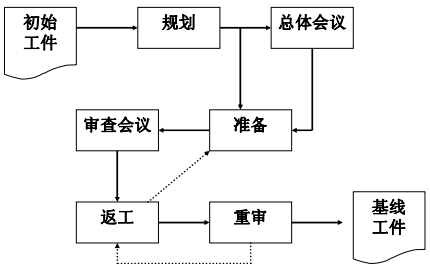
轮查 Passaround

- ◆ 轮查是由多人组成的并行同级桌查
- ◆ 轮查有助于缓和同级桌查的两个主要风险
 - 评审者不能及时提供反馈
 - 评审效果太糟

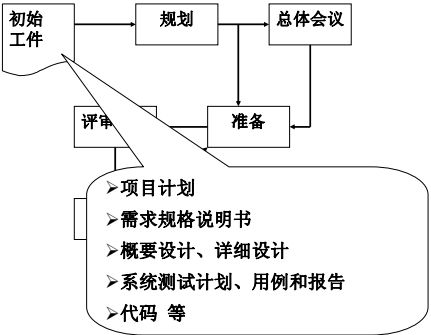
选择合适的评审方法

评审目标	审查	小组评审	走查	结对编程	同级桌查	轮查
查找产品缺陷	×	×	×	×	×	×
检查规范的一致性	×	×			×	×
检查是否符合标准	×				×	×
检查完整性/正确性	×		×			
评估可理解性/可维护性	×	×		×		×
证实关键构件的质量	×					
过程改进	×	×				
测量文档质量	×					
培训其他组员熟悉产品		×	×	×		×
对方法达成共识		×	×	×		
确保修改和纠错正确		×	×		×	
寻找可替换的方法			×	×		
模拟执行程序			×			
评审开销最小化					×	

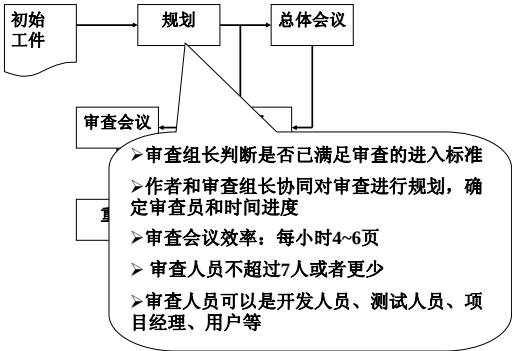
审查过程



受审查的工件



审查的规划



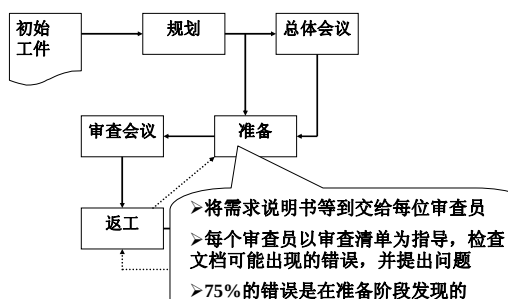
进入审查的标准

- ◆ 文档符合标准模板
- ◆ 文档已经做过拼写检查和语法检查
- ◆ 作者已经检查了文档在版面安排上所存在的错误
- ◆ 已经获得了审查员所需要的先前或参考文档
- ◆ 在文档中打印了行序号以方便在审查中对特定位置的查阅
- ◆ 所有未解决的问题都被标记为TBD（待确定）
- ◆ 包括了文档中使用到的术语词汇表

练习

- ◆ 如果要评审交大选课系统的需求文档，共20页，你是评审组长，会安排哪些人作为评审人员？安排几次会议？每次会议开几小时？

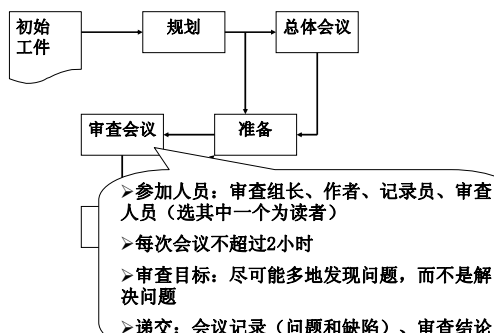
审查的准备



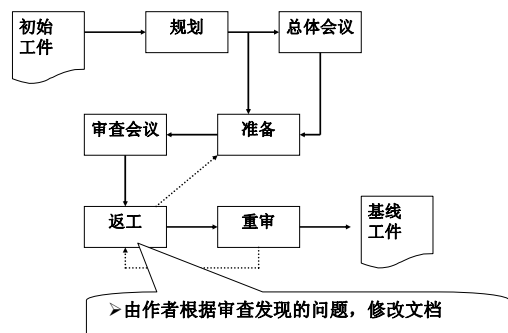
练习

- ◆ 一个有效的技术评审会议必须让每个人事先都有充分的准备，您如何识别哪个评审者没有做准备工作？您若是评审组组长，您怎么做？

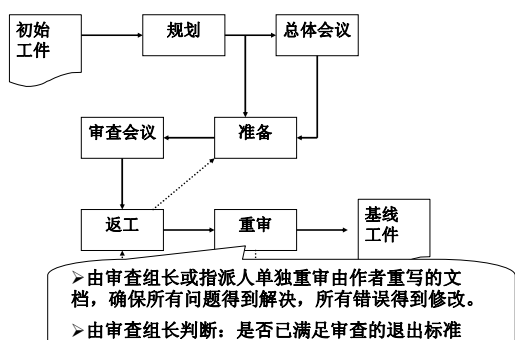
审查会议



重写



复审



Software Process and Management

49

沈备军

退出审查的标准

- ◆ 已经明确阐述了审查员提出的所有问题
- ◆ 已经正确修改了文档
- ◆ 修订过的文档已经进行了拼写检查和语法检查
- ◆ 所有TBD的问题已经全部解决，或者已记录下每个待确定问题的解决过程，目标日期和提出问题的人
- ◆ 文档已经登记入项目的配置管理系统

Software Process and Management

50

沈备军

评审的关键成功因素

- ◆ 建立成功的软件评审程序
- ◆ 分析早期的评审
- ◆ 尽早并经常进行正式或非正式评审
- ◆ 评选一名评审冠军，树立榜样
- ◆ 为评审制定目标
- ◆ 在项目计划中为评审分配资源和时间
- ◆ 培训评审者和评审组长
- ◆ 获得管理层的支持



Software Process and Management

51

沈备军

评审的最佳实践准则

- ◆ 制订审查计划，明确项目目标和审查目标
- ◆ 使用严格的，量化的准入和准出条件
- ◆ 在文档生成的初始阶段就开始对其进行审查
- ◆ 核对源代码和相关的文档
- ◆ 在最适当的时机进行准备和审查
- ◆ 把焦点放在主要缺陷上
- ◆ 测量审查的投资回报率
- ◆ 重视缺陷预防和过程改进



Software Process and Management

52

沈备军

评审的陷阱

- ◆ 参与评审的人不了解评审过程
- ◆ 评审过程没有被遵循
- ◆ 适合的人没有参加评审
- ◆ 评审会议陷入到对问题的解决上
- ◆ 评审的焦点放在文档形式而不是内容本身上
- ◆ 评审结果被用于作者的业绩评估



Software Process and Management

53

沈备军

陷阱1：参与评审的人不了解评审过程

- ◆ 症状
 - 没有掌握各种类型的评审
 - 组织内多个评审小组不能遵循一致的评审过程
- ◆ 对策
 - 培训、实践和文档化的评审过程



Software Process and Management

54

沈备军

陷阱2：评审过程没有被遵循

- ◆ 症状
 - 放弃，或采用其它的方法
 - 随意简化，或有时做有时不做
 - 走过场，时间和资源安排不足
- ◆ 原因
 - 评审过程过于复杂
 - 没有建立相关政策
 - 质量不是项目成功的驱动力
- ◆ 提示
 - 在一个已经陷入进度超期、需求混乱、人员疲惫之困境的项目中引入评审是很艰苦的，但是如果评审能将项目引入正常轨道，那么这是完全值得的。

Software Process and Management

55

沈备军

如何评审大型文档

- ◆ 问题
 - 审查员们可能会认真地检查第一部分，一些意志坚定的人可以检查到中间部分，没有人坚持到最后部分
- ◆ 建议
 - 在将成为基线时再审查
 - 在开发过程中采用非正式的、渐增式的评审
 - 不同审查员从文档的不同位置开始检查
 - 如果有足够的审查员，可分成几个小组分别审查文档的不同部分

Software Process and Management

56

沈备军

陷阱3：适合的人没有参加评审

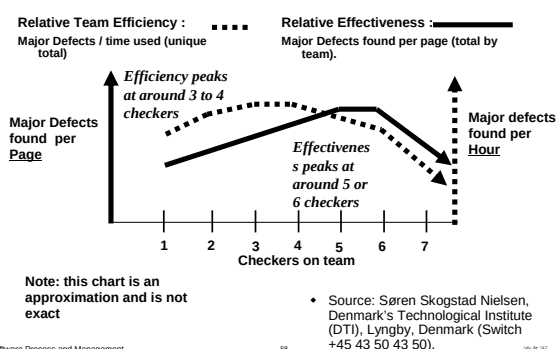
- ◆ 症状
 - 一些评审员的资质不够，抱着学习的目的
 - 评审员所代表的角色不全
 - 一些评审员没有明确的评审目的
 - 人员过多，导致过多的争议
- ◆ 对策
 - 列出各类文档或产品的评审角色
 - 评审会议不宜超过7人

Software Process and Management

57

沈备军

评审小组人数对效率的影响



Software Process and Management

58

沈备军

如何面对庞大的审查小组

- ◆ 确保每个参与者都是为了寻找错误，而不是为了了解需求，或维护行政上的位置；让只想了解审查内容的人参加总体会议，而不是审查会议。
- ◆ 理解审查员所代表的观点，委婉拒绝以相同的观点看待问题的参与者。在准备阶段可收集多人的信息，但只派一个人作为代表参加会议。
- ◆ 把审查组分成若干小组并行审查，再把结果合并起来。

Software Process and Management

59

沈备军

陷阱4：评审会议陷入到对问题的解决上

- ◆ 后果
 - 停止了对产品的检查
 - 对所讨论的问题不感兴趣的评审者停止了思考
 - 当会议结束的时间快到了，评审者会很快翻过剩下的材料，宣布评审的成功，而这些材料中很可能隐藏了主要缺陷
- ◆ 对策
 - 评审组长要控制会议的焦点，集中在发现问题上
 - 除非特别要求评审成为一次头脑风暴会议

Software Process and Management

60

沈备军

陷阱5：评审的焦点放在文档形式而不是内容本身上

- ◆ 原因
 - 缺乏充足的准备，只做了表面的检查
- ◆ 对策
 - 事先定义编码标准和文档标准模板
 - 将产品是否符合模板和标准作为评审准入条件



Software Process and Management

61

沈备军

陷阱6：评审结果被用于作者的业绩评估

- ◆ 后果
 - 为了避免惩罚，开发者可能不会将他们的产品交付评审，同时也会拒绝评审同事的工作以避免使别人受到惩罚
 - 在评审过程中，评审者可能不会直接指出错误，相反会事后告诉作者，这样就不会得罪作者
 - 开发者可能在这种带惩罚性质的评审之前会事先进行一次“预评审”以非正式地过滤出一部分错误，从而降低评审效率
 - 评审小组可能会老在争论是否算一个缺陷
 - 尽量少地找出错误成为评审心照不宣的目标，从而减少评审的投资回报。
- ◆ 对策
 - 评审时对事不对人，评审的是产品，而不是评审作者
 - 依据产品最终的结果来考核开发者

Software Process and Management

62

沈备军

讨论

- ◆ 一个程序是正确的，但质量不好，有没有可能性？
- ◆ 设立独立的质量保证小组有什么好处？它们和软件工程组的关系和合作应该是如何的？
- ◆ 如果您被委任来负责提高公司的质量，您第一件应做的事是什么？第二件？
- ◆ 查找一段代码的bug, 可以采用什么方法？哪种最有效？

Software Process and Management

63

沈备军

代码bug的排除效益

- ◆ 代码自查 70-80%
- ◆ 代码互查 50-70%
- ◆ 编译 50%
- ◆ 单元测试 40-50%
- ◆ 集成测试 45%
- ◆ 系统测试 45%

[Humphrey 97 Introduction to the PSP]

Software Process and Management

64

沈备军