ISTQB Certified Tester

ISTQB初级认证

第3章 静态技术

作者：郑文强

Email: zwqwwuy@163.com

博客：http://blog.csdn.net/Wenqiang_Zheng

声明

➔ 本课件的开发基于ISTQB Foundation Level Syllabus (Version 2007)。

➔ 感谢ISTQB和大纲作者的努力，对应的大纲可以从www.istqb.org下载获得。

➔ 本课件为个人开发，只能用于个人学习目的，不能用于任何商业活动。

➔ 更多ISTQB初级认证资料，参考：
http://blog.csdn.net/Wenqiang_Zheng/archive/2011/04/09/6311523.aspx

课程内容

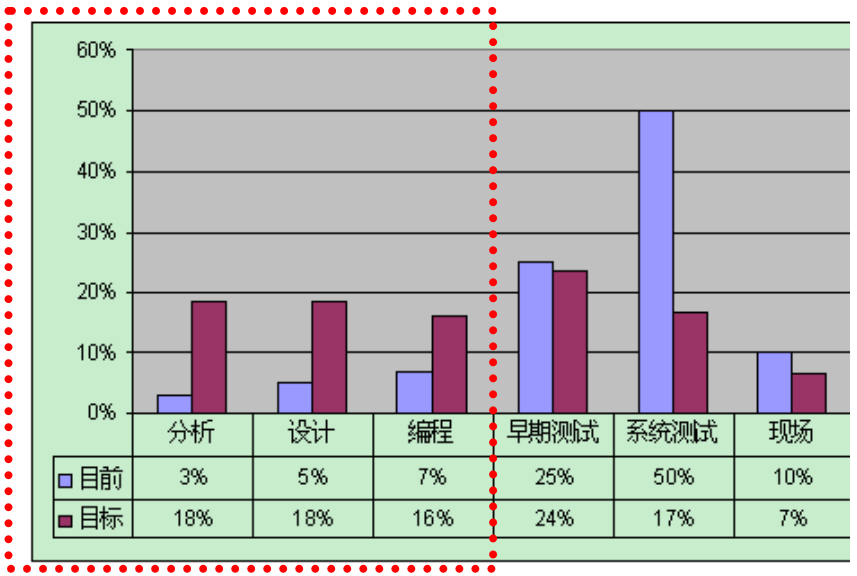
1. 静态技术和测试过程
2. 评审过程
3. 静态分析的工具支持

静态技术和测试过程

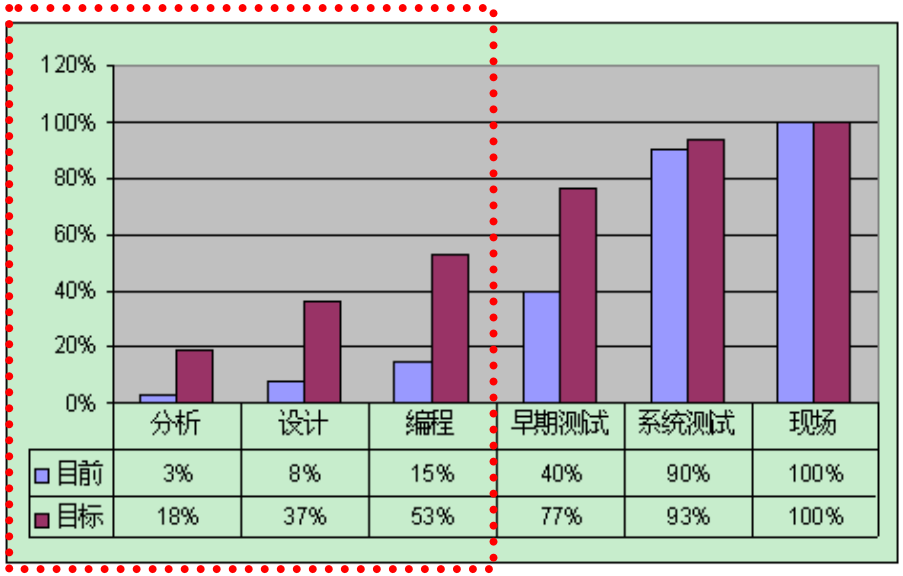
ISTQB考试知识点

- ★ 了解可以通过不同的静态技术来检查并确认软件工作产品的质量（K1）；
- ★ 描述了在评估软件工作产品中运用静态技术的重要性和它的价值（K2）；
- ★ 解释静态技术和动态技术之间的区别（K2）；
- ★ 描述静态分析和评审的目标，并且和动态测试进行对比（K2）；

静态技术



阶段缺陷的分布



阶段缺陷的累计分布

为了降低缺陷的成本，应该在开发活动的早期进行静态测试，尽早在动态测试之前去发现缺陷并修改缺陷；

----软件测试是控制成本的关键

静态技术

基本定义

★ 静态测试是广义测试概念中的重要组成部分。与动态测试技术需要运行软件不同，静态测试技术通过手工检查（评审）或自动化分析（静态分析）的方式对代码或者其他的项目文档进行检查：

- ✦ 手工检查
- ✦ 静态分析

★ 在早期通过人工检查和静态分析发现和修改缺陷，其成本会比通过动态测试发现和修改缺陷成本低的多；

静态技术

★静态测试可以完全以人工的方式进行（评审），也可以通过工具支持的方式来进行（静态分析）；

★静态测试的主要活动是检查工作产品，并对工作产品做出评估；

★静态测试的结果可以优化开发过程，并达到缺陷预防的目的；

静态技术

静态测试的对象

- ★ 需求规格说明
- ★ 设计规格说明
- ★ 代码
- ★ 测试规格说明
- ★ 测试用例
- ★ 测试脚本
- ★ 用户指南或WEB页面
- ★ ……

软件工作产品

静态技术

静态测试的优点

★ 尽早发现和修改缺陷，降低测试成本和产品生命周期成本；

★ 改善测试能力和开发能力；

★ 缩短和减少测试和开发时间；

★ 减少缺陷和改善沟通；

★ 静态测试也可以在工作产品中发现一些遗漏的内容，例如发现需求有遗漏，而这在动态测试中是很难被发现的；

★ ……

静态技术

静态测试的优点

质量提高

成本降低

进度提前

能力提升

静态测试

静态测试发现的缺陷类型

★ 与标准之间的偏差

★ 需求内的错误

★ 设计错误

★ 可维护性不足

★ 错误的接口规格说明

★ ……

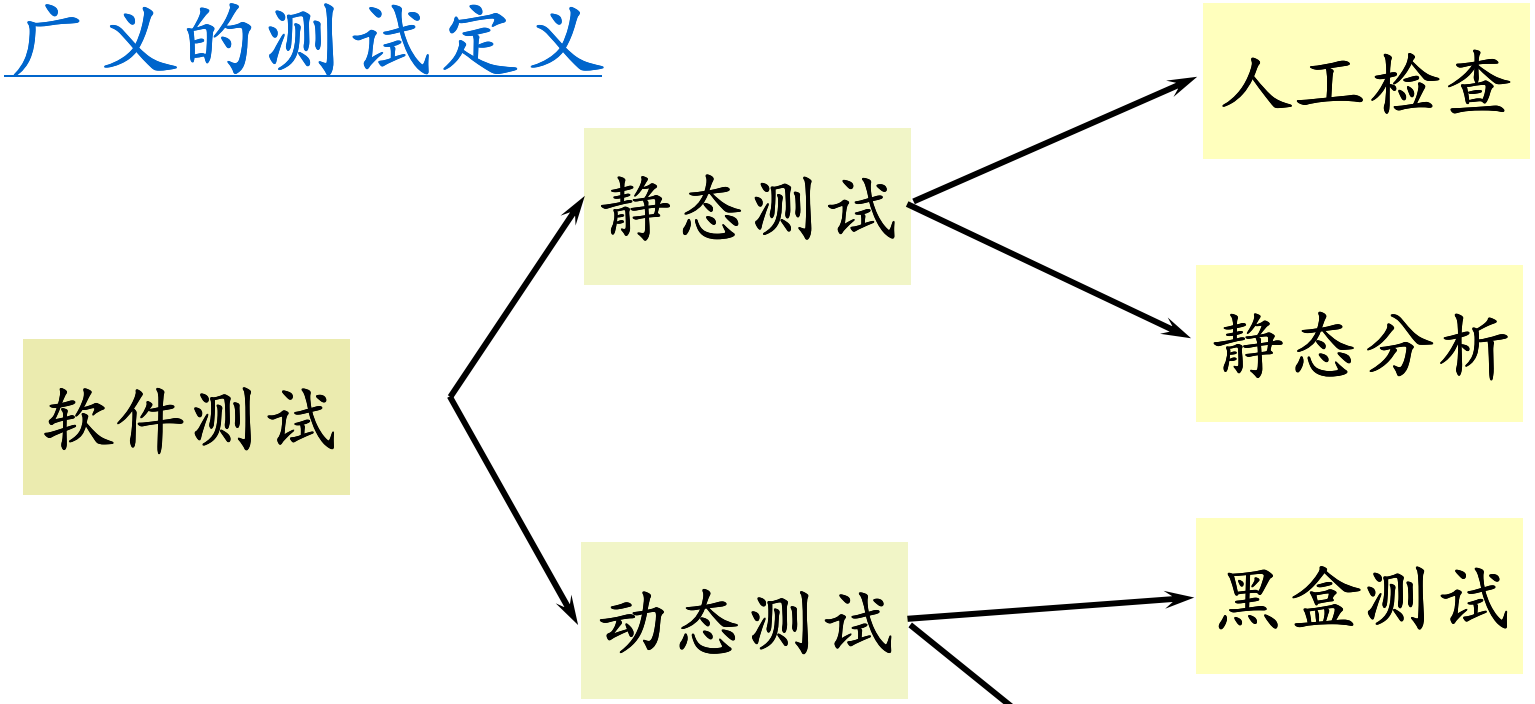
静态测试

静态测试和动态测试异同点

- ★静态测试不用运行测试对象，而动态测试需要；
- ★静态分析和动态测试具有共同的目标：识别和发现缺陷；
- ★静态测试和动态测试是互补的：不同的技术可以有效和高效地发现不同类型的缺陷；
- ★与动态测试相比，静态技术发现的是软件失效的原因而不是失效本身；

静态测试

广义的测试定义



不同的软件测试方法的组合可以发现不同类型的错误。

课程内容

1. 静态技术和测试过程
2. 评审过程
3. 静态分析的工具支持

评审过程

ISTQB考试知识点

- ★理解典型的正式评审过程中的阶段、角色和职责定义（K1）；
- ★解释不同类型评审的区别：非正式评审(informal review)、技术评审(technical review)、走查(walkthrough)和审查(inspection)（K2）；
- ★解释影响评审成功的主要因素（K2）；

评审过程

基本含义

- ★ 评审，也就是所谓的人工检查，是对所有不同文档或者代码检查技术的通称；
- ★ 评审可以是非正式的评审，也可以是正式的评审。评审过程的形式和以下因素相关：
  - ✦ 开发过程的成熟度
  - ✦ 法律法规方面的需求
  - ✦ 审计跟踪的需要
  - ✦ ……

评审过程

评审的目标

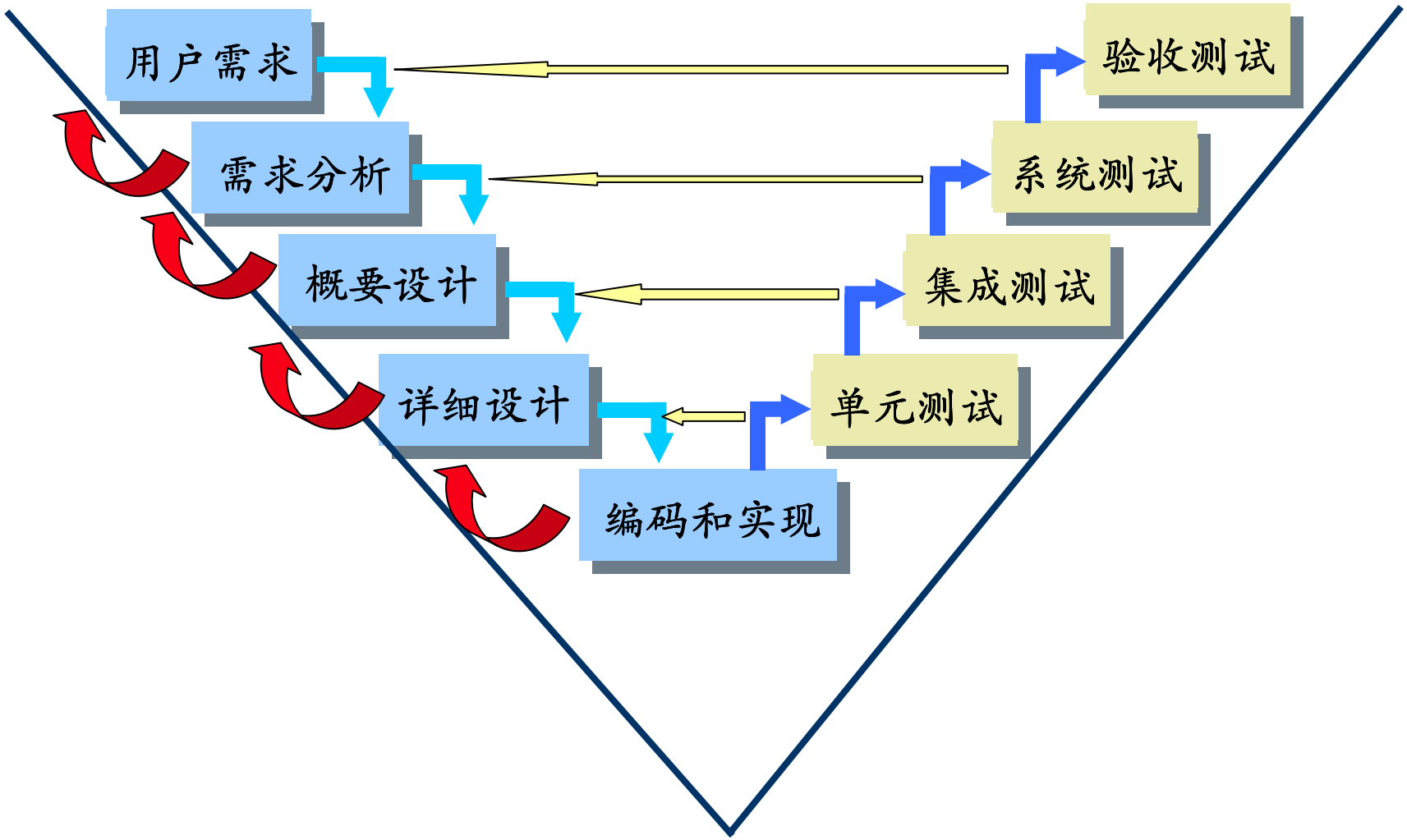
★ 查找和发现缺陷

★ 增加对评审对象的理解

★ 讨论和确定评审对象采用的技术和方法

★ ……

评审过程



评审过程

评审的优点

★ 评审可以使移除缺陷的成本更低；

★ 评审可以缩短开发时间。如果在早期识别和修正了缺陷，用来执行动态测试的成本和时间就可以减少；

★ 由于存在缺陷的数量减少，产品整个生命周期的成本就会降低；

★ 降低系统运行的故障率。通过评审在产品开发早期发现尽量多的缺陷，并进行修改，可以提高产品的质量，从而减少系统运行的故障；

评审过程

评审的优点（续）

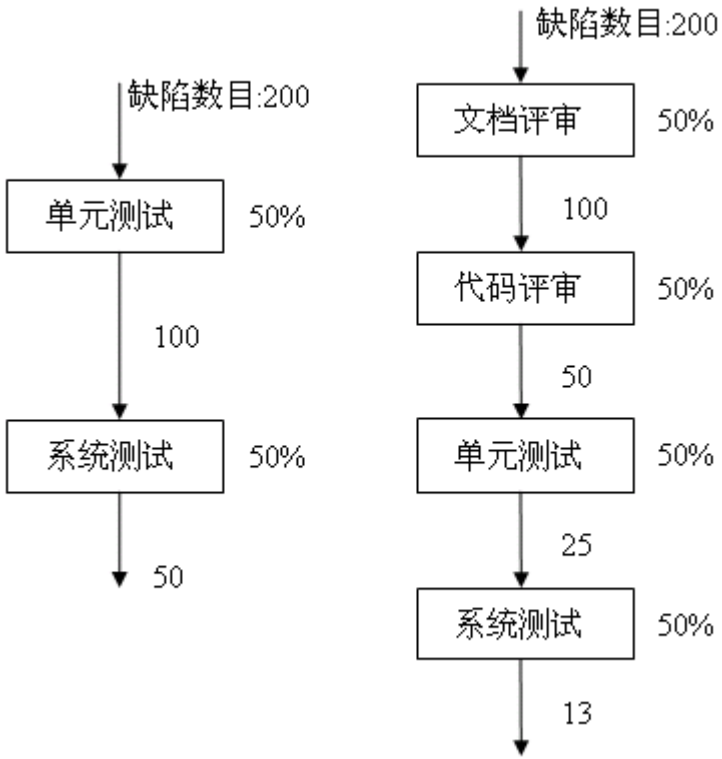
★ 评审可以提高后续发布的产品的质量。评审是通过一组人来进行的，因而评审过程中可以相互学习。通过改进每个人的工作方法，从而提高个人的技能水平；

★ 评审可以提高文档编写的质量。通常一个明确的文档要求，可以使作者发现一些遗忘的问题，即文档要进行评审，可以迫使文档相关人更加关注文档相关的标准和要求；

★ 整个团队对被检查目标的质量负责，因此，整个小组的成员可以对文档的内容理解达成共识；

评审过程

评审的优点举例：提高质量



评审过程

评审的优点举例：降低成本

阶段	缺陷修改成本
用户需求和分析	$1
概要设计	$2
详细设计	$5
编码和实现	$10
单元测试	$15
集成测试	$22
系统测试	$50
用户验收和使用	$100+

评审过程

评审过程的组成

★ 计划阶段

★ 预备会阶段

★ 个人准备阶段

★ 评审会议阶段

★ 返工阶段

★ 跟踪结果阶段

评审计划阶段

计划阶段的任务

- ★什么文档需要进行评审？定义文档评审的目标；
- ★评审对象是否准备完毕？
- ★选择文档评审的方式：结对评审还是团队评审；
- ★评审人员和评审主持人是谁？任命文档评审的主持人；
- ★什么时候进行评审工作？
- ★需要花费的工作量是多少？
- ★……

评审计划阶段

计划阶段出口准则

- ★ 评审对象package已经准备好，包括评审对象、和评审对象相关的文档，比如所有的输入文档、评审的指南、规则和检查表等；
- ★ 评审相关的参与人员已经指定，并且进行了相关评审方面的培训；
- ★ 评审计划和时间、评审目标等已经定义；

评审的主持人负责来检查文档评审阶段的出口准则是否满足！

评审预备会阶段

预备会阶段的活动

★ 分发文档；

★ 参与人员的角色和负责的职责，比如是负责技术领域还是流程方面；

★ 解释评审的目标和过程；

★ 建议的评审速度，和缺陷的记录形式；

★ 检查入口准则（针对更正式的评审类型）；

预备会的主要目的是共享评审对象的信息，以及评审的重点和目的！

个人准备阶段

个人准备阶段的活动

★ 每个评审人员各自准备评审工作：目的是发现文档中可能存在的缺陷、也可以是疑问或者改进的建议；

★ 每个评审人员各自记录发现的缺陷，其中的疑问等，并全部记录在评审日志文件中；

★ 完成后，将评审日志文件发给评审主持人；

评审人员的充分准备是成功进行评审的前置条件！

个人准备阶段

个人准备阶段的出口准则

★ 主持人负责收集和分析评审日志文件；

★ 主持人决定结束还是继续个人准备阶段；

★ 主持人和评审参与人员确认评审会议的时间；

评审会议阶段

评审会议阶段的活动

- ★ 讲解评审对象文档，并反馈在准备阶段发现的缺陷和疑问给作者；
- ★ 发现和记录新发现的缺陷和悬而未决的问题；
- ★ 识别和制订应对措施来跟踪处理悬而未决的问题；
- ★ 讨论和评估文档评审结果：通过、有条件通过、还是未通过；
- ★ 记录评审过程中的经验教训，以及一些改进建议等；

评审会议阶段

缺陷的严重程度

- ★严重缺陷（评审对象不能满足它设计的目的，在批准评审对象之前必须修正相关缺陷）；
- ★重要缺陷（影响评审对象的可用性，批准评审对象之前应该修改相关缺陷）；
- ★一般缺陷（小的偏差，基本不影响使用）；
- ★好的（没有缺陷，返工时无需修改）；

缺陷的严重程度和数目决定了评审的结果！

评审会议阶段

评审的结果

★ 通过：无需进行修改；

★ 有条件通过：需要修改，但不需要进一步评审；

★ 未通过：需要进一步的评审或其它的检查措施；

评审会议阶段

评审的通用准则

- ★评审会议的时间尽量限制在2个小时内。如果需要，可以在当天再发起另外一个评审会议；
- ★如果一个或多个专家（评审人员）没有出席，或者他们准备不充分的话，主持人有权取消或中止会议；
- ★评审过程中不讨论开发方案、设计方法以及其他常见的风格问题；
- ★主持人一般情况下不做为评审人员；

评审会议阶段

评审的通用准则（续）

★提交讨论的是被评审的文档（检查对象），而不是作者：

★评审人员必须要注意他们的言语以及表达的方式；

★作者不应该为自己或文档辩护。（这意味着：作者不应该被攻击或者被迫处于一个防御的位置。但是作者对他们决定的辩护或解释，应该视为合理和有帮助的）；

返工阶段

返工阶段的活动

★ 评审主持人检查文档评审报告，并重新分发文档评审报告；

★ 对所有发现的缺陷，进行修改工作，或者涉及的条目并不是缺陷（比如是功能），可以拒绝或者推迟到下个版本进行修改等；

★ 悬而未决的问题：按照应对计划进行了处理，和主持人一道，通过信息的收集和分析作出决定；

出口准则：文档评审报告中的所有事项都进行了处理！

跟踪结果阶段

跟踪结果阶段的活动

★ 检查缺陷是否已解决；

★ 检查出口准则；

★ 将文档评审过程中的数据记录到评审数据库中；

★ 记录、计算和分析得到的数据；

★ 分析其中的经验和教训，并定义可能的改进计划，比如针对检查表、评审过程等的改进建议；

评审过程

评审失败的原因

- 需要的人员没有空，或者不具备必需的资格或技术能力。这一点对于主持人尤为重要，因为相对技术技能，他们必须具备更多的心理上的技能。这可以通过培训或者使用咨询公司的有资质的人员来解决。
- 管理层在资源计划时不准确的估计可能导致评审的时间压力，进而导致令人不满意的评审结果。有时候一个较低成本的评审类型能够缓解这个问题。
- 评审由于准备不足而失败。这种情况大部分是因为选择了不合适的评审人员造成的。如果评审人员没有认识到评审的重要性以及对质量改进的巨大影响，并且评审因此失败，那么有必要通过演示图形来证明评审是如何提高生产率的。

评审过程

评审失败的原因

- 评审也可能因为没有文档或者文档不足而失败。评审之前，必须进行检查以验证所有需要的文档已经存在，并且已经描述充分了。只有这个时候，才能够进行评审。
- 如果没有管理层的支持，评审过程是无法成功的，因为这样无法提供必需的资源，且评审的结果也不会用于过程改进。不幸的是，评审经常会出现这种情况。

角色和职责

评审过程角色

- ★ 经理
- ★ 主持人
- ★ 评审员
- ★ 记录员
- ★ 观察员
- ★ 作者
- ★ ……

角色和职责

经理

- ★ 选择评审的工作产品，并定义评审的目标；
- ★ 选择评审的方式：结对评审还是团队评审；
- ★ 在计划中预留和分配评审的时间，以及其他的相关资源；
- ★ 选择合适的评审主持人；
- ★ 确保评审按照流程进行，并且验证评审结果和评审的输出内容；

对于技术文档，我们不建议经理参与评审！

角色和职责

主持人

★ 协助经理和作者定义评审目标、选择评审人员和计划评审工作；

★ 在各个评审阶段，检查评审的入口准则和出口准则是否满足；

★ 领导评审会议的顺利进行，确保评审过程中，关注点在定义的评审目标上面；

★ 记录文档评审相关数据和度量到评审数据库，比如评审的时间、参与的人员等；

评审主持人是评审成功的关键！

角色和职责

评审员

★ 参加评审预备会和正式的文档评审会议;

★ 查找和发现文档中的缺陷;

★ 检查文档和要求的标准之间的符合程度;

★ 对文档内容进行评估;

角色和职责

记录员

★记录员也可以是评审人员之一；

★记录员的主要职责是完整记录在评审会议中识别的缺陷和其他所有的不确定的事件；

角色和职责

观察员

- ★ 获取评审的一些信息，或者是为了熟悉文档评审的流程，起到培训的作用；
- ★ 观察员也可能是过程改进组成员，用来确保评审过程是否满足组织内定义的流程和准则；
- ★ 直接参与收集评审的信息，负责分析文档评审过程，以及在此基础上进行评审过程的改进等方面的活动；

角色和职责

作者

- ★向项目负责人申请安排文档评审；
- ★和协助理，选择合适的评审人员和评审主持人；
- ★准备被评审的文档，使之达到评审的入口准则；
- ★在评审开工会上，将被评审文档以及评审的一些要求，简单介绍给评审参与人员；
- ★修改在评审中发现的缺陷，以及其他一些需要修改的建议和意见；

评审的类型

评审的主要类型

★ 技术评审：技术相关的工作产品的评审；

- ✦ 走查（Walkthrough）
- ✦ 审查（Inspection）
- ✦ 技术评审（Technical Review）
- ✦ 非正式评审（Informal Review）

评审的类型

评审的主要类型

★ 管理评审：针对项目计划和开发过程相关的分析和评审；

- 目的是监控过程、判定计划和进度的状态，或评估为达到特定目的管理方法的有效性；

- 管理评审一般在项目得到里程碑、完成项目主要阶段，或者针对项目进行经验总结的时候开展；

走查

走查的目的

- ★ 走查是一种非正式评审，作者在评审会议上向评审人员介绍文档的内容；
- ★ 主要目的：
  - ✦ 评审人员之间相互学习和增进评审对象的理解；
  - ✦ 评估文档内容和标准规格之间的符合程度；
  - ✦ 发现文档中的缺陷、含糊的表达和问题等，从而来改进产品质量；
  - ✦ ……

走查

走查的特点

★ 走查通常由作者召集开评审会；

★ 以情景、演示的形式方式和同行进行评审；

★ 开始-结束会议模式；

★ 评审会议之前的准备、评审报告、发现的问题列表和记录人员（不是作者本人）都不是必需的；

★ 在实际情况中，走查可以是非常正式的，也可能是非正式的；

作者主导评审会议，因此对走查过程影响力最大！

审查

审查的目的

★ 审查是最正式的评审之一，遵循正式严谨的一个评审过程。通常每个评审人员都是从作者的直接同事中选出，并且具有固定的角色；

★ 主要目的：发现文档的不清晰要点和可能的缺陷，以及度量文档质量、改进产品质量和开发过程；

审查

审查的特点

- ★ 由专门接受过培训的主持人（不是作者本人）来领导；
- ★ 根据入口准则、出口准则和检查表定义正式的评审过程；
- ★ 定义了不同的角色和职责，通常是同行检查；
- ★ 审查会议之前需要进行准备；
- ★ 审查之后出具审查报告和发现问题的列表；
- ★ 对问题列表的更新有正式的跟踪过程；

技术评审

技术评审的目的

★ 评审关注的焦点是文档和规格说明的一致性、文档目的的适用性以及和标准的一致性；

★ 技术评审有各种不同的版本：从非常不正式的到有严格定义、有正式过程定义的技术评审；

★ 主要目的：讨论、评估、发现缺陷、解决技术问题、检查和规格及标准符合程度；

技术评审

技术评审的特点

★ 定义并且文档化的缺陷发现过程，需要包含同行和技术专家。

★ 一般情况下，没有管理人员参与；

★ 理想情况下由专门接受过培训的主持人（不是作者本人）来领导。

★ 会议之前需要进行准备；

★ 可能使用检查表、评审报告、发现的问题列表；

★ 在实际情况中可以是非常正式的，也可能是非正式的；

非正式评审

非正式评审的目的

★ 非正式评审是评审的精简版。但是，它或多或少以一种简单的方式遵循评审的通用过程。比如结对编程、结对测试、代码交换 等；

★ 非正式评审的主要目的：以较低的成本发现问题；

非正式评审

非正式评审的特点

★ 没有正式的评审过程；

★ 对设计文档和代码以技术评审为主；

★ 评审的过程和结果可能是文档化的；

★ 根据不同的评审人员，评审的有效性不同；

如何选择不同的评审类型

不同类型的选择

- ★ 评审结果的表现形式有助于选择评审类型。是否需要具体的文档，或者通过非正式的方式进行检查就足够了？
- ★ 寻找合适的时间来进行评审的难易程度？把5或7个技术专家召集起来参加一个会议或多个会议是否非常困难？
- ★ 是否需要有不同领域的技术知识？
- ★ 需要多少有资质的评审参与人员？评审人员有激励机制吗？

如何选择不同的评审类型

不同类型的选择（续）

- ★评审的收益（预期结果）和投入评审相关的工作量是否相一致，或者说是否值得?
- ★评审对象是否需要正式的记录？是否可以通过工具支持来开展分析活动?
- ★管理层是否支持评审活动？在项目工作面临时间压力的时候，管理层是否会缩减评审工作量?
- ★……

评审的注意事项

评审的注意事项

- ★ 评审的每一种类型，没有统一的描述，每种评审类型也没有明确的界限，且经常在不同场合使用不同的术语；
- ★ 评审的类型更多的是由公司的组织结构来决定，根据项目的需要进行相应的裁减，以提高评审的效率；
- ★ 参与评审的项目人员的协同合作有利于项目质量的提高；
- ★ 评审的方式可以是多种多样：面对面方式、电话会议方式、视频会议方式等；

如何成功开展评审活动

评审成功的因素

- ★ 每次评审都有明确的目的，比如评审的主要目的是提高被评审文档的质量、发现其中的问题和模糊的描述观点、偏差；
- ★ 对发现的缺陷持欢迎态度，并且发现的问题必须以中性和客观的方式进行记录；
- ★ 能够正确处理不同人员之间的想法和心理方面的问题，个人的特性和心理都会对评审产生很强的影响；评审文档的作者应该将评审作为一种正面的经历；

如何成功开展评审活动

评审成功的因素（续）

- ★使用检查表和指南来提高评审过程中发现问题的效率；
- ★管理层可以为软件开发过程中的文档评审计划足够的资源（时间和人力），来支持一个有效的评审过程；
- ★成功运用评审的一个重要方面是不断从评审过程本身来学习经验教训，比如评审过程的持续改进；

如何成功开展评审活动

评审成功的因素（续）

★ 提供评审技术方面的培训，特别是针对正式的评审技术，比如审查技术的培训；

★ 管理层对评审过程的支持（在项目计划中安排足够的时间来进行评审活动）；

★ 强调学习和过程的改进；

课程内容

1. 静态技术和测试过程
2. 评审过程
3. 静态分析的工具支持

静态分析的工具支持

ISTQB考试知识点

- ★ 理解通过静态分析能够识别的典型缺陷和错误，并与评审和动态测试之间进行比 较（K1）；
- ★ 列出静态分析的典型优点（K1）；
- ★ 列出通过静态分析工具识别的典型的代码缺陷和设计缺陷（K1）；

静态分析的工具支持

静态分析的定义

★ 静态分析指的是不需要通过运行程序代码，就可以对测试对象进行检查的技术。静态分析并不需要真正运行软件，而是通过工具来检查软件；

★ 和评审一样，静态分析目的是发现文档或者代码中的缺陷或者可能存在缺陷的地方。不同的是，静态分析经常是通过工具的支持来进行的，一般是自动化的方式来实现；

★ 静态分析的另外一个目的是得到度量数值，从而对测试对象的质量进行测量和验证；

静态分析的工具支持

静态分析和评审的关系

★ 静态分析与评审一样，通常发现的是软件的缺陷而不是软件运行的失败；

★ 静态分析一般是工具支持的自动化的过程，而评审一般是团队通过人工的方式进行；

★ 静态分析的对象一般需要以特定的格式和标准来进行组织，才能工具支持的自动化检查，而评审不一定；

★ 静态分析和评审是紧密联系的。假如在评审之前进行了静态分析，可以发现很多的缺陷，则评审需要检查的地方就可以明显的减少；

静态分析的工具支持

静态分析的优点

★ 在评审和动态测试之前尽早发现缺陷和问题；

★ 通过度量的计算（比如高复杂性测量），早期警惕可能存有问题的代码和设计；

★ 可以发现在动态测试过程不容易发现的一些缺陷；

★ 可以发现软件模块之间的相互关联性和不一致性；

★ 改进代码和设计，增强可维护性；

★ 在开发过程中学习经验教训，从而预防缺陷；

静态分析的工具支持

静态分析发现的缺陷类型

★ 引用一个没有定义值的变量;

★ 模块和组件之间接口不一致;

★ 从未使用的变量;

★ 不可达代码或死代码;

★ 违背编程规则;

★ 安全漏洞;

★ 代码和软件模型的语法错误;

静态分析的工具支持

编译器分析工具

★ 可以发现编程语言语法错误，并且以缺陷或告警的方式进行报告。主要检查：

- 产生不同程序元素的交叉引用列表（比如变量、函数）；
- 检查编程语言中数据和变量的正确数据类型；
- 检查没有定义的变量；
- 检查不可达代码；
- 检查域边界的上限或下限（静态选择）；
- 检查接口的一致性；
- 检查所有作为跳转开始或跳转结束标签的使用；

静态分析的工具支持

规范标准一致性检查

★ 通过工具也可以来检查测试对象是否与规范、标准相一致。比如是否遵循了大部分的编程规范和标准。这种检查方式几乎不需要花费多少时间和人力成本；

★ 工具检查常常还有一个优点：假如编程人员知道代码需要和编程规范进行一致性检查，他们会比没有这种自动测试的人员更乐于按照编程规范来工作；

静态分析的工具支持

数据流分析

- ★ 数据流分析是另一种发现缺陷的手段。这种方法是通过检查程序代码中使用的数据来进行检查的；
- ★ 变量的三种状态：
  - ✦ 已定义的（D）：变量已经赋值；
  - ✦ 已引用的（R）：读取或使用变量的值；
  - ✦ 没定义的（U）：变量没有定义具体的数值；

静态分析的工具支持

数据流分析（续）

★ 区分数据流异常的三种情况：

- UR-异常：程序路径上读取了没有赋值的变量；
- DU-异常：变量赋值，但是这个变量已经是无效或者没有定义，同时没有被引用；
- DD—异常：变量赋了第二个值，同时第一个值不在使用；

静态分析的工具支持

数据流分析例子

```
void exchange (int& Min,   int& Max) {
        int Help;
        if (Min > Max) {
            Max = Help;
            Max = Min;
            Help = Min;
            }
}
```

静态分析的工具支持

数据流分析例子（续）

- ★ 变量Help的UR-异常：变量的范围是局限在函数内。变量的第一次使用是在赋值语句的右面部分。这个时候，变量还没有赋值，而这里进行了直接的引用；
- ★ 变量Max的DD－异常：在赋值的左边，Max变量连续使用了两次，因此赋值赋了两次；
- ★ 变量Help的DU-异常：函数的最后一个赋值，变量help被赋了一个任何地方都不能用的数值，因为变量只有在函数内是有效的；

静态分析的工具支持

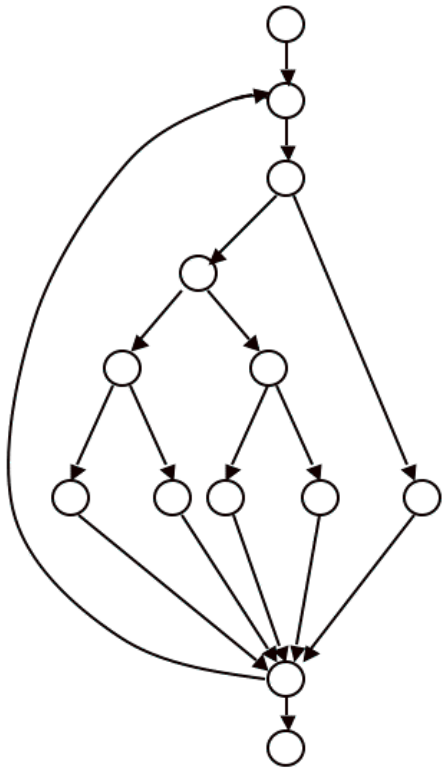
确定度量

★ 静态分析工具也可以提供度量值。软件系统的质量属性可以通过度量值来进行度量；

★ 圈数（McCable数[McCable 76]）。圈数可以用来测量程序代码的结构复杂程度。圈数计算的根据是控制流图；

静态分析的工具支持

确定度量的例子



★ 对于程序或程序模块的控制流图G，它的圈数可以通过下面的公式计算得到：

$$v(G) = e - n + 2$$

★ 其中：

- v（G） = 控制流图G的圈数6
- e = 控制流图中的边数17
- n = 控制流图的节点数13

静态分析的工具支持

圈数的作用

- ★ 圈数可以用来估算程序代码的可测试性和可维护性；
- ★ 圈数描述了程序模块中独立路径的数目。假如需要达到100%的分支覆盖，则控制流图中所有这些独立的路径至少需要执行一次。因此，圈数提供了关于测试数目的重要信息；
- ★ 程序圈数的数值越高，理解程序模块的难度越高；

答 疑