




SOFTWARE TESTING

苏临之

sulinzhi029@nwu.edu.cn



α - and β -Testing

- ❖ α 测试是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试，不能由程序员或测试员完成。
- ❖ β 测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，不能由程序员或测试员完成。所以 β 测试是在开发者无法控制的环境下进行的软件现场应用。



Differences

- ❖ 两者主要区别是测试的场所、环境和时序不同：
 - α 测试可以是指把用户请到开发方的场所来测试， β 测试是指在一个或多个用户的场所进行的测试。
 - α 测试可以的环境是受开发方控制的，用户的数量相对比较少，时间比较集中。而 β 测试的环境是不受开发方控制的，且用户数量相对比较多，时间不集中。
 - 一般 α 测试可以先于 β 测试执行。



Defect Management

- ❖ 软件缺陷管理是在软件生命周期中识别、管理、沟通任何缺陷的过程。该过程从缺陷的识别开始，直到到缺陷的解决关闭为止，确保缺陷被跟踪管理而不丢失。一般的，需要跟踪管理工具来帮助进行缺陷全流程管理。
- ❖ 每一个软件组织需要明确必须妥善处理软件中的缺陷，因为这关系到软件组织机构生存、发展的质量根本。但是并非所有的软件组织都明确应当如何有效地管理自己软件中的缺陷。



Description of a Defect

- 缺陷ID
- 缺陷标题
- 缺陷详细描述
- 缺陷提交人和提交时间
- 缺陷状态
- 缺陷严重程度
- 缺陷紧急程度
- 缺陷等级
- 缺陷所属项目/模块
- 缺陷指定解决人和解决时间
- 缺陷复核情况
- 测试环境
- 必需附件



Seven Defect Statuses

- New (发现错误 • 新建)
- Open (打开)
- Fixed (已修复)
- Reopen (再次打开)
- Closed (关闭)
- Rejected (拒绝)
- Delayed (延迟)



Process of Defect Management

- 缺陷管理的流程可以概括为：测试人员提交新的错误入库，缺陷状态为“发现错误·新建”；项目经理将缺陷分配给相应的开发人员，缺陷状态为“打开”；开发人员查重现缺陷，做如下处理：如果不是缺陷，则置状态为“拒绝”；如果是缺陷则修复并置状态为“已修复”；如果不能解决的错误，要留下文字说明并保持错误为“拒绝”状态。测试人员查询状态为“已修复”的错误，验证错误是否已解决，做如下处理：如问题解决了置错误的状态为“关闭”，如果问题得不到立即解决则需要置状态为“延迟”。

Five Defect Levels

等级	说明	举例
低	可在发布后再商量是否改进	A. 某些测试的建议
中	不影响功能的正常使用,可以在时间和资源允许的情况下再解决	A. 辅助性说明描述不清楚 B. 显示格式不规范 C. 长时间操作未给用户进度提示 D. 提示窗口文字未采用行业术语 E. 可输入区域和只读区域没有明显的区分标志 F. 系统处理未优化
高	事件是重要的,但是由于解决问题需要花费一定的时间,所以可以用较长的时间解决,如果时间紧可以留下个版本解决	A. 界面文字等错误 B. 打印内容、格式错误 C. 简单的输入限制未放在前台进行控制 D. 删除操作未给出提示



Five Defect Levels

等级	说明	举例
很高	事件是重要的，并且应该在紧急的事件处理之后尽快得到解决，必须在发布前解决	A. 功能不符 B. 缺少功能，与需求不符 C. 数据流错误 D. 程序接口错误 E. 轻微的数值计算错误
紧急	事件非常重要，测试工作无法继续进行，需要马上给予关注解决。	A. 由于程序所引起的死机，非法退出，运行中断，应用程序崩溃 B. 死循环 C. 导致数据库发生死锁 D. 数据通讯错误 E. 严重的数值计算错误



Planning a Testing Effort

- ❖ 软件测试中最后一个要点就是把先前的软件测试知识（去哪里找、如何测试、如何有效测试）联系起来，说明和软件测试有关的所有工作如何计划、如何组织。
- ❖ 通过利用精心组织的测试计划、测试用例和测试报告，可以对测试工作进行正确的记录以及交流，将使达到目标变得更加可能。



Goals

- ❖ 如果测试员之间不交流计划测试的对象，需要什么资源，进度如何安排，整个项目就很难成功。
- ❖ 软件测试计划是软件测试员与产品开发小组交流意图的主要方式。
- ❖ 测试计划的目标就是：规定测试活动的范围、方法、资源和进度；明确正在测试的项目、需要测试的特性、要执行的测试任务、每个任务的负责人，以及与计划相关的风险。

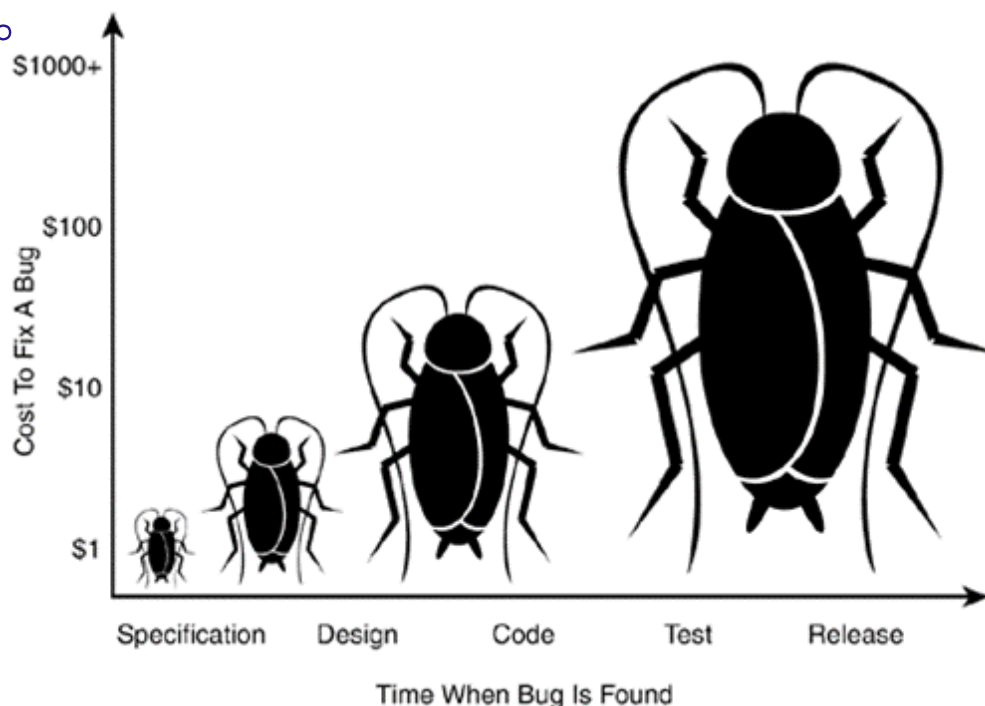


Main Theme

- ❖ 测试计划主题是一套清单，该清单应该在整个项目小组中全面讨论、相互沟通并达成一致。这比测试计划文档模版更加重要。
- ❖ 测试计划的主题主要包括以下方面：高级期望、人员（包括地点和事物）、责任、测试必要性、测试阶段、测试策略、资源需求、任务分配、测试进度、测试用例、缺陷报告、度量统计、风险问题等。

Why the Quality is Free?

- ❖ 软件缺陷发现得越晚，则维护成本就越高。
- ❖ 制造高质量的产品和制造低质量的产品相比，其额外开销可能会小。





Quality Cost

- ❖ 一致性费用（Cost of Conformance, COC）指的是一次性计划和执行测试所相关的全部费用，用于保证软件按照预期方式去运行。
- ❖ 非一致性费用（Cost of Non-Conformance, CONC）既包含在开发中由质量缺陷产生的费用（特别是因缺陷导致的返工），又包含了产品交付后的诸多费用（例如产品召回、官司等一些巨大损失产生的费用）。



Internal & External Failures

- ❖ 内部失败（Internal Failures）指的是在发布产品之前所需要的各种缺陷。由内部失败产生的非一致性费用相对较小。
- ❖ 外部失败（External Failures）指的是在发布产品之后所发现的各种缺陷。由外部失败产生的非一致性费用相对较大，有时候甚至导致公司无法运作以至破产。
- ❖ 一致性费用和内部失败引起的非一致性费用之和往往会小于外部失败引起的非一致性费用。



Examples

❖ Intel奔腾浮点除法：1994

- $(4195835/3145727)*3145727-4195835$ 算式的结果如果为0，则计算机没有问题；如果结果不为0，则使用的是带有浮点除法软件缺陷的老Intel奔腾CPU，该软件缺陷被刻录在一个芯片中，并反复生产制造。
- 在发布芯片之前的内部测试已经发现了这个问题，不过因为只有精度要求很高的计算中才导致错误，Intel管理层认为不严重，未修正
- 更换芯片的费用：4亿美元



Tester and Quality

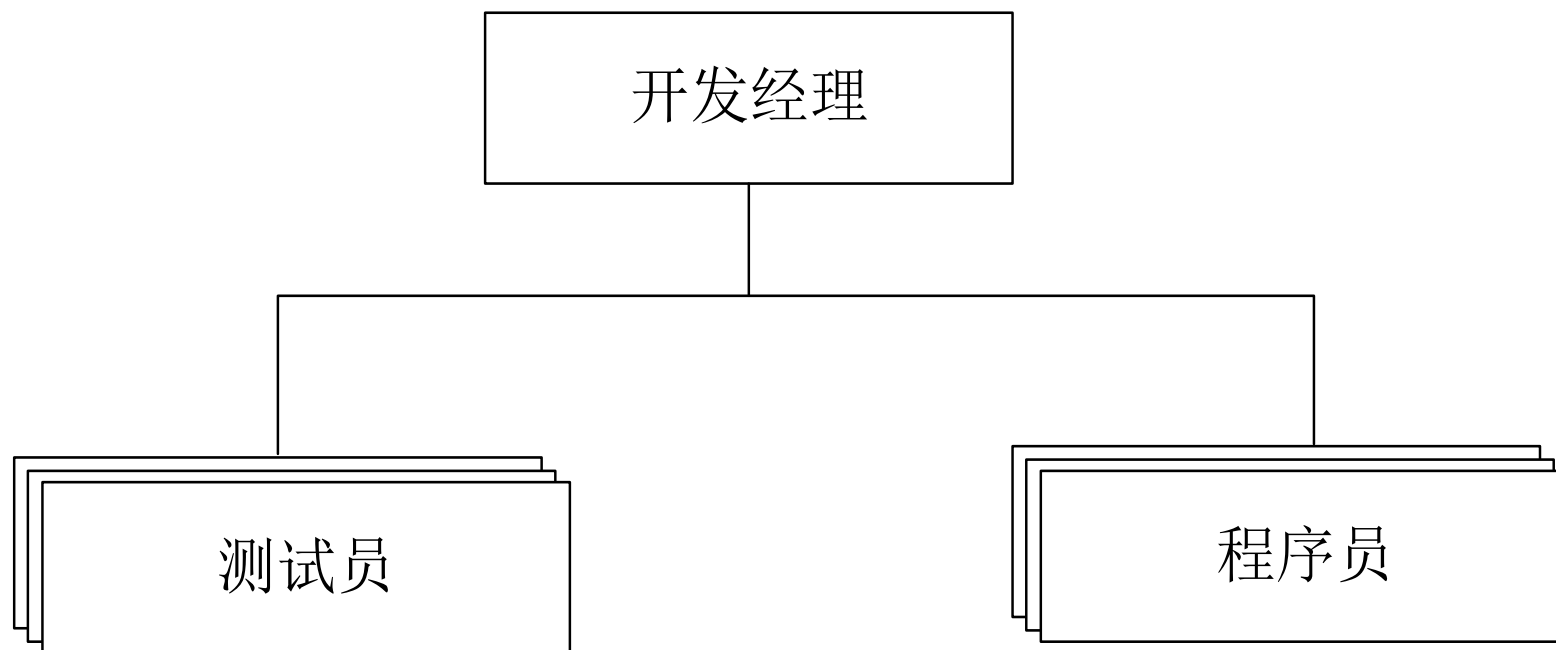
- ❖ 软件测试员的目标是尽快尽早发现软件缺陷，并确保其得以修复。
- ❖ 为了实现这个目标，需要做两方面工作：第一是发现缺陷，第二是确保其能够得以修复。
- ❖ 因此，质量并不是靠测试来解决的，因为测试的最终解决者并不是软件测试员。



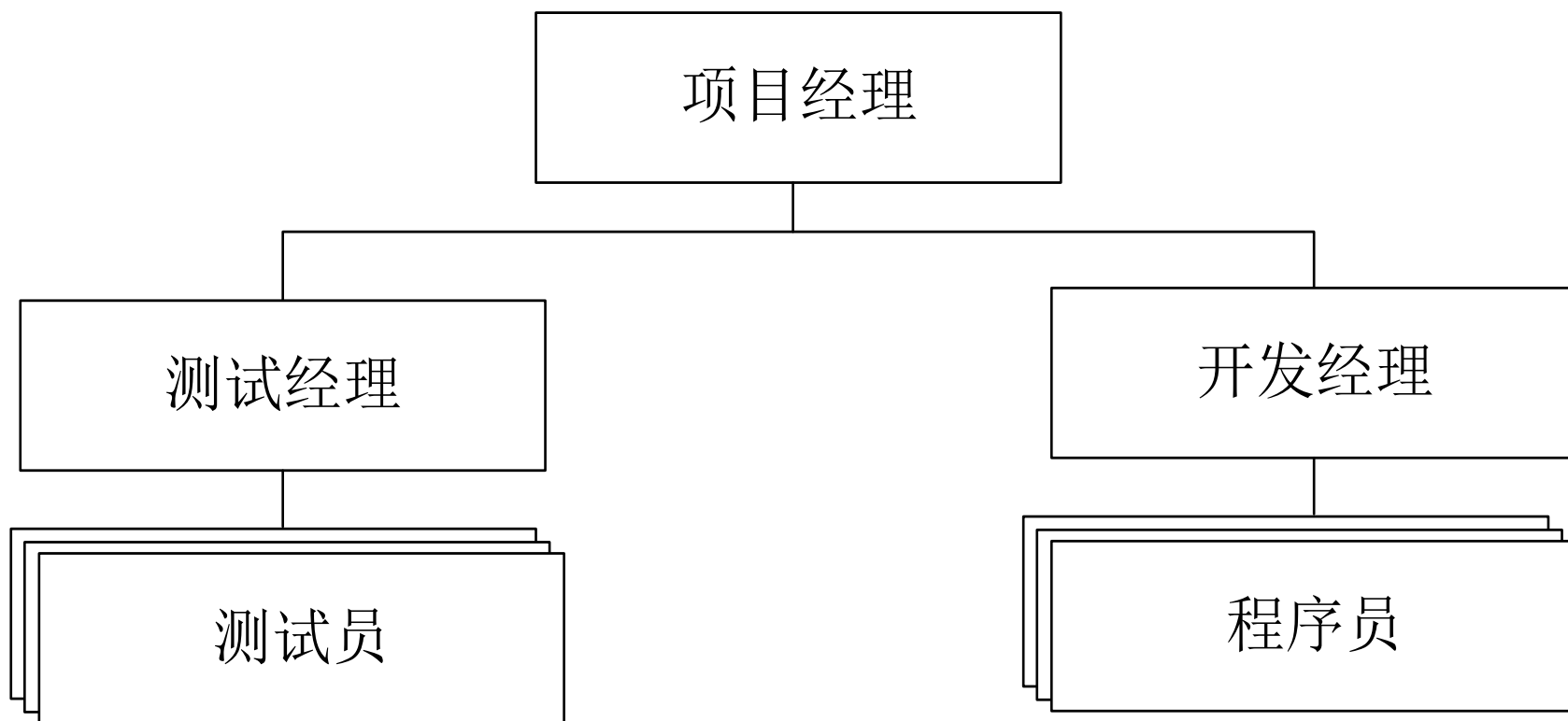
QA Personnel

- ❖ 软件质量保证（QA）人员的主要职责是检查和评价当前软件开发的过程，找出改进过程的方法，从而达到防止软件缺陷出现这一目标。
- ❖ QA团队比软件测试团队责任更大，因为他们是需要对软件缺陷加以负责。
- ❖ 项目经理需要以实现“0缺陷”为首要目标，而不是让软件如期发布，也不是让其费用低于预算。

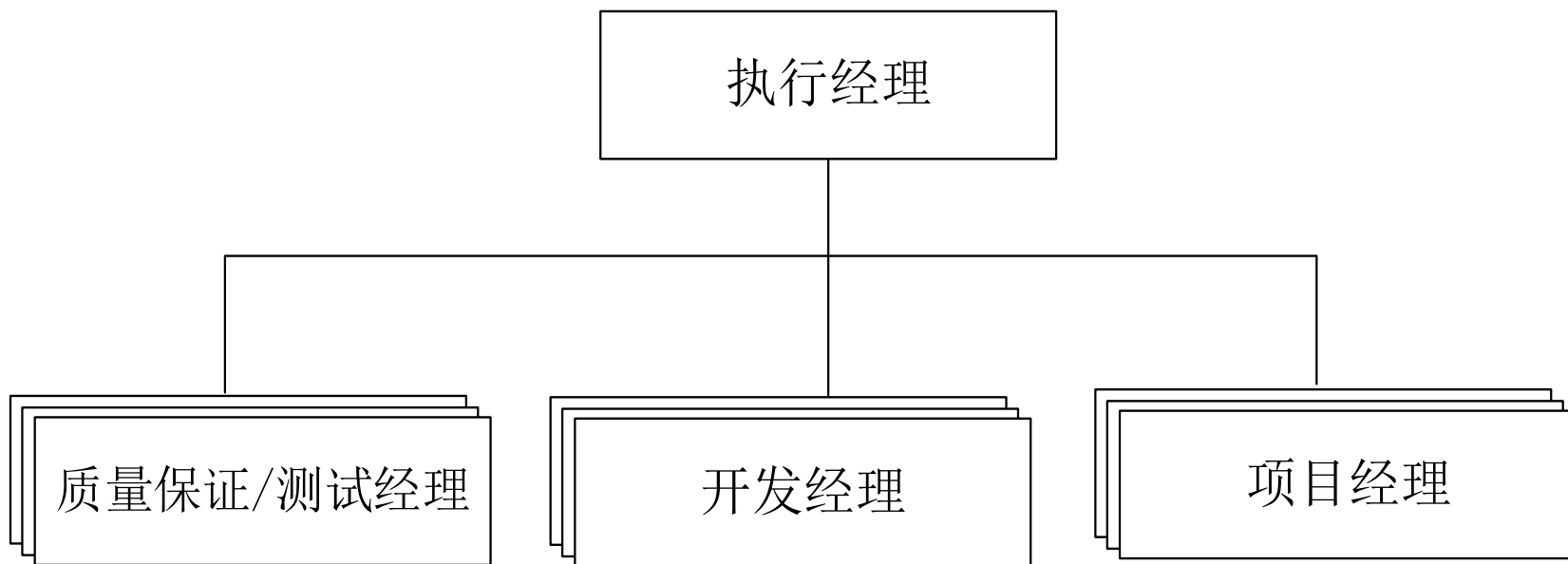
Organization Structure: Level 1



Organization Structure: Level 2



Organization Structure: Level 3





Capability Maturity Model

- 能力成熟度模型（CMM）是一个行业标准模型，用于定义和评价软件公司开发过程的成熟度。共分为5个等级。
 - 1级：初始的。随意和混乱的过程，项目成功依赖于个人英雄主义和运气。这是软件公司存在最多的级别。
 - 2级：可重复的。项目级的思想，类似的项目成功经验可以重复使用。多数的软件公司位于这个等级。
 - 3级：定义的。组织级别的思想。具有预先考虑和文档化、标准化管理和工程活动。少部分软件公司位于此等级。
 - 4级：可管理的。可控的过程，且对过程和产品质量有详细的评估和理解。极少数软件公司位于此等级。
 - 5级：不断优化的。通过量化的反馈和新途径实现持续改进。几乎没有几家公司能达到这个标准。



Work Categories

- ❖ 软件测试技术人员
- ❖ 软件测试员或软件测试工程师
- ❖ 软件测试工具开发师或软件测试开发工程师
- ❖ 软件测试负责人
- ❖ 软件测试经理
- ❖



How to Find a Suitable Job?

- ❖ 网络寻找
- ❖ 媒体寻找
- ❖ 开放日和招聘会查看
- ❖ 实习和协作中寻找
- ❖ 临时性工作
- ❖



Final Examination

- 目标1：掌握基软件测试的基本概念、基本理论、常用测试方法与测试技术。
 - 目标2：能够使用常见的测试自动化工具以提高测试的效率和准确性。
 - 目标3：能够针对实际问题选择合适的测试策略，综合应用测试理论、测试方法与技术并运用测试工具进行软件部件测试的设计与实施。
-
- 基本概念和基本理论，对应目标1（ $27 \div 0.6 = 45$ ）
 - 综合大型题目，对应目标3（ $27 \div 0.6 = 45$ ）
 - JUnit测试工具的使用，对应目标2（ $6 \div 0.6 = 10$ ）



Final Examination

- 全英文试题，务必熟悉专业术语
- 熟悉课堂上讲过的概念，尤其是近年考试反复出现的概念和多年未考的概念
- 测试技术方法必须复习全面不留死角，理解方法的本质
- JUnit测试掌握其用法，尤其是如何使用注释和断言



THANK YOU!