软件质量保证与测试

Software Quality Assurance and Testing

第 7 章 测试组织和管理

金陵科技學院

主要内容

- 软件测试团队管理
- 项目管理
- 测试用例管理
- 缺陷管理

1.软件测试团队管理

团队成员类型

1）具有创新精神的测试人员

2）有测试欲望并能够持之以恒的测试人员

3）富有经验的软件测试人员

4）具有远见性的测试人员

测试经理

其职责是计划并协调项目的测试流程，并负责以下内容：

- 测试团队进行部门间的交互；
- 根据实际需要，与客户和供应商进行交互；
- 招聘、监管以及培训员工；
- 创建测试计划；
- 创建测试流程的预算和日程计划，包括估算测试工作量；
- 获取测试环境所需的硬件和软件；
- 确保测试环境和测试产品的正确配置管理；
- 定义测试流程；
- 跟踪测试流程进度；
- 召开测试前和测试后会议。

测试经理

测试经理需要具备以下技能：

(1)了解测试流程和方法；

(2)熟悉包括测试环境和数据管理、错误报告和解决方案以及测试设计和开发在内的相关测试程序；

(3)了解应用程序业务范围和应用程序需求；

(4)熟悉开发测试目的、目标和策略；

(5)熟悉各种测试工具、缺陷跟踪工具以及其他测试支持工具；

测试经理需要对任何偏离计划工作量、成本、日程安排和最后可交付成果质量的情况负责。

测试组长

指导测试团队，其职责包括：

- 为测试程序提供技术指导；
- 提供对客户界面、测试工具介绍、测试计划执行、员工监管和进度状态报告的支持；
- 验证需求的质量；
- 与测试工具供应商进行交互，以确定针对测试项目使用的测试工具的最佳方法；
- 接受关于最新测试方法和工具的信息，并且将这些知识传达给测试团队；
- 进行测试设计和测试流程走查和检视；
- 根据进行的调查，实现测试流程改进；
- 通过使用需求可追溯性矩阵来跟踪针对测试需求的测试流程；
- 实现测试流程；
- 确保测试产品文档完整。

测试组长

测试组长需要具备以下技能：

- 了解应用程序业务范围和应用程序需求；
- 熟悉包括数据管理、错误报告和解决方案以及测试设计和测试开发在内的相关测试程序；
- 精通各种技术技能，包括编程语言、数据库技术和计算机操作系统；
- 熟悉各种测试工具
- 缺陷跟踪工具以及其他支持测试生命周期的工具。

测试环境专员

专注于设置测试环境，其职责包括：

① 安装测试工具并建立测试工具环境；

② 通过使用环境设置脚本，创建并控制测试环境；

③ 创建并维护测试数据库；

④ 在测试工具环境中，维护需求层次结构。

测试环境专员

测试环境专员需要具备以下技能：

① 精通网络、数据库和系统管理；

② 精通各种技术技能，包括编程和脚本语言、数据库技术和计算机操作系统；

③ 精通测试工具和数据库；

④ 精通产品评估和整合。

测试员

软件测试员测试软件产品。软件测试员在测试流程中进行下列活动：

(1)开发测试用例和制定测试流程；

(2)创建测试数据；

(3)审核分析和设计工件；

(4)执行测试

(5)使用自动化工具来测试；

(6)准备测试文档；

(7)跟踪缺陷；

(8)报告测试结果。

测试员

软件测试员需要具备下述技术技能：

(1)了解软件开发、操作和维护流程：测试员必须了解组织中用于开发、操作和维护软件系统的流程；

(2)了解应用程序：测试员需要详细了解用户将如何使用应用程序的功能以及用户可能导致的错误的类型；

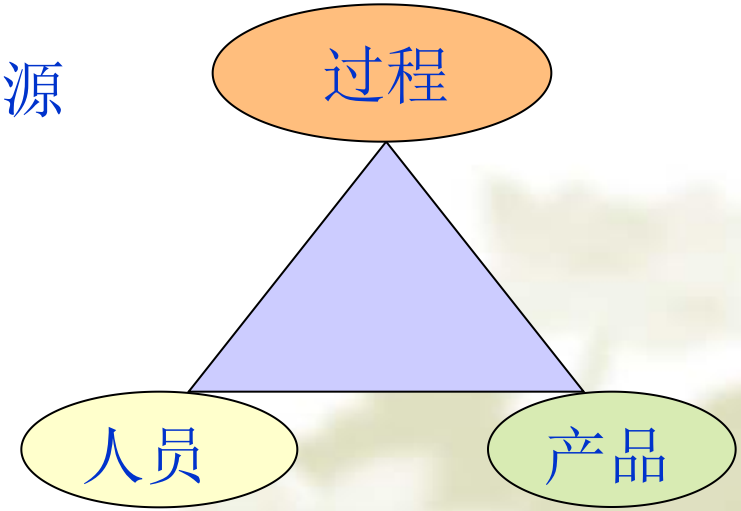
(3)了解用于软件开发的工具：测试员必须了解有助于计划、分析和开发及维护运行的工具和方法；

(4)了解测试流程：测试员必须了解测试内容、测试方法、测试时间、测试执行人员以及测试停止时间；

(5)了解测试流程文档：测试员必须了解并熟悉软件开发生命周期的每个阶段中组织开发的所有文档。

软件测试项目管理

- 过程：进度安排
- 人员：任务、责任、资源
- 产品：达到的目标



测试阶段的信息流

测试阶段的输入信息有两类：

- 软件配置：这是测试的依据和测试的对象，包括
  需求说明书、规格说明书、软件设计书
  被测的源程序等。

- 测试配置：包括
  测试计划　　测试步骤
  测试用例（测试数据）　　具体实施测试的测试脚本
  测试工具等

测试过程安排

测试准备：

测试需求分析和测试计划

测试环境搭建

测试用例设计

测试实施：

执行、记录、沟通、验证

测试总结：

测试结果汇总分析、评价报告

测试需求分析

1. 什么是测试需求？

- 确切地讲，所谓的测试需求就是在项目中要测试什么。我们在测试活动中，首先需要明确测试需求（What），才能决定怎么测（How），测试时间（When），需要多少人（Who），测试的环境是什么（Where），测试中需要的技能、工具以及相应的背景知识，测试中可能遇到的风险等等，以上所有的内容结合起来就构成了测试计划的基本要素。

2. 为什么要做测试需求分析

- 测试需求越详细精准，表明对所测软件的了解越深，对所要进行的任务内容就越清晰，就更有把握保证测试的质量与进度。

软件测试计划

- 测试计划就是描述所有要完成的测试工作，包括被测试项目的背景、目标、范围、方式、资源、进度安排、测试组织，以及与测试有关的风险等方面。
- 制定软件测试计划可以在以下几方面帮助我们:

1. 使软件测试工作进行更顺利
2. 促进项目参加人员彼此的沟通
3. 及早发现和修正软件规格说明书的问题
4. 使软件测试工作更易于管理

编写测试计划

序号	条目	内容
1	测试概要	摘要说明所需测试的软件、名词解释以及所参考的相关文档。
2	测试目标	对测试目标进行简要的描述。
3	测试范围	测试计划所包含的测试软件需测试的范围和优先级，哪些需要重点测试、哪些无需测试或无法测试或推迟测试。
4	测试策略	针对每个范围内的要素和内容，制定测试策略、选择所使用的测试技术和方法。
5	重点事项	列出需要测试的软件的所有的主要功能和测试重点，这部分应该和测试用例设计相对应，并互相检查。
6	测试配置	测试所需要的软硬件、测试工具、必要的技术资源、培训、文档等。
7	人员组织	需要什么样的人进行测试，多少人测试，各自的角色和责任，需要进行哪些学习和培训。
8	沟通方式	测试组内部、测试与开发组之间使用什么样的工具、标识和频度进行沟通和确认。
9	测试进度	将测试计划合理分配到不同的测试人员，并注意先后顺序。对于长期大型的测试计划，可以使用里程碑来表示进度的变化。
10	测试标准	测试开始、完成、延迟以及继续的标准，包括制定测试开始和完成的标准；某些时候，测试计划会因某种原因（过多阻塞性的Bug）而导致延迟，问题解决后测试继续。
11	发布提交	在按照测试计划进行测试发布后需要交付的软件产品、测试用例、测试数据及相关文档。
12	风险分析	需要考虑测试计划中可能的风险和解决方法。

测试环境搭建

- 搭建良好的测试环境是执行测试用例的前提，也是完成测试任务顺利完成的保证。

- 测试环境大体可分为硬件环境和软件环境，硬件环境包括测试必须的PC机，服务器，设备，网线，分配器等硬件设备；软件环境包括数据库，操作系统，被测试软件，共存软件等；特殊条件下还要考虑网络环境，比如网络带宽，IP地址设置等。

3.测试用例管理

- 什么是测试用例

  测试用例（Test Case）通常是指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求。

测试用例设计

- 测试用例设计应包括：

1.基本事件：参照规格说明书，按照需要实现的所有功能编写，覆盖率100%。

2.备选事件：设计过程中的备选情况，按照功能点编写。

3.异常事件：出错处理的路径，按照功能点编写。

在实际中，备选事件和异常事件的测试用例往往比基本事件的测试用例要多。

测试用例的评审

- 是否覆盖测试需求上的所有功能点？
- 用例编号是否和需求对应？
- 非功能测试需求或不可测试需求是否在用例中列出并说明？
- 用例设计是否包含了正面、反面的用例？
- 每个测试用例是否清楚填写了测试特性、步骤、预期结果？
- 步骤/输入数据部分是否清晰，是否具备可操作性？
- 测试用例是否包含测试数据、测试数据的生成办法或者输入的相关描述？
- 优先级安排是否合理？
- 是否已经删除了冗余的用例？
- 是否简洁，复用性强？例如，可将重复度高的步骤或过程抽取出来，定义为一些可复用标准步骤。

测试用例的优化

① 利用设计测试用例的八种方法不断地对测试用例进行分解与合并；

② 采用遗传算法理论进化测试用例；

③ 在测试时利用发散思维构造测试用例。

测试用例的更新

- 测试用例在形成文档后也还需要不断完善。主要起因于三个方面：

  第一、在测试过程中发现设计测试用例时考虑不周，需要完善；

  第二、在软件交付使用后反馈的软件缺陷，而缺陷又是因测试用例存在漏洞造成的；

  第三、软件自身的新增功能以及软件版本的更新，测试用例也必须配套修改更新。

测试用例的作用

- 错误跟踪
- 更准确地反映软件的某一特性
- 全面地反映软件的性能和质量
- 明确故障责任

测试用例执行

（1）测试开始标准

测试计划评审通过；

测试用例已编写完成，并已通过评审；

测试环境已搭建完毕。

测试用例执行

（2）测试停止标准

满足以下条件，测试可以正常停止。

- 测试用例全部通过；
- 需求覆盖率达到100%；
- 确认系统满足产品需求规格说明书的要求。
- 缺陷状态为“关闭”（Closed）或“推迟”（Later）状态；
- 在系统测试中发现的错误已经得到修改，各级缺陷修复率达到规定的标准；
- 缺陷密度（n个/KLOC）需要符合软件要求的范围；

测试用例执行

- 在执行测试时需要注意以下几点：

（1）确认搭建的测试环境和用例执行的环境需要一致，否则将严重影响测试用例的执行。

（2）注意测试用例中的前提条件和特殊规程说明。因为有些测试软件是有顺序性的，相应地测试用例会有一些执行前提或特殊说明。如果忽略这些内容，可能会导致测试用例的无法执行。

（3）测试用例要按步骤全部执行，每条用例至少执行一遍，才能保证待测试软件能正确满足用户需求。在用例设计时就要求做到覆盖所有需求，每个用例都和某个需求相对应。

测试用例执行

（4）执行测试用例时，要详细记录软件系统的实际输入输出，仔细对比实际输入和测试用例中的期望输入是否一致。如果不一致，要从多个角度多测试几次，尽量详细地确定软件出错的位置和原因，并测试确定该错误会不会导致更严重的错误出现。最后，把详细的输入和实际的输出以及描述写到测试报告中。

（5）不要放过任何偶然现象。测试时可能会发现某用例执行软件会出错，但是再次执行时该错误又不再重现。其实，这种错误才是隐藏最深的、最难发现的错误。发现时，要仔细分析这种情况，不要放过任何小的细节，多测试几次，准确找出问题的原因。

测试记录、沟通、验证

(1) 全方位的观察测试用例执行结果
(2) 加强测试过程记录
(3) 及时确认发现的问题
(4) 与开发人员良好的沟通
(5) 及时更新测试用例
(6) 提交一份优秀的问题报告单
(7) 测试结果分析

软件测试的评测

测试的评测主要方法包括覆盖评测和质量评测。

测试覆盖评测是对测试完全程度的评测，它建立在测试覆盖基础上，测试覆盖是由测试需求和测试用例的覆盖或已执行代码的覆盖表示的。

质量评测是对测试对象的可靠性、稳定性以及性能的评测。质量建立在对测试结果的评估和对测试过程中确定的缺陷及缺陷修复的分析基础上。

覆盖评测

- 覆盖评测指标是用来度量软件测试的完全程度的，所以可以将覆盖用做测试有效性的一个度量。最常用的覆盖评测是基于需求的测试覆盖和基于代码的测试覆盖，它们分别是指针对需求（基于需求的）或代码的设计/实施标准（基于代码的）而言的完全程度评测。
- 在执行测试过程中，经常使用两个测试覆盖度量指标，一个是确定已执行的测试覆盖率，另一个是确定成功的测试覆盖率，即执行时未出现失败的测试覆盖率。

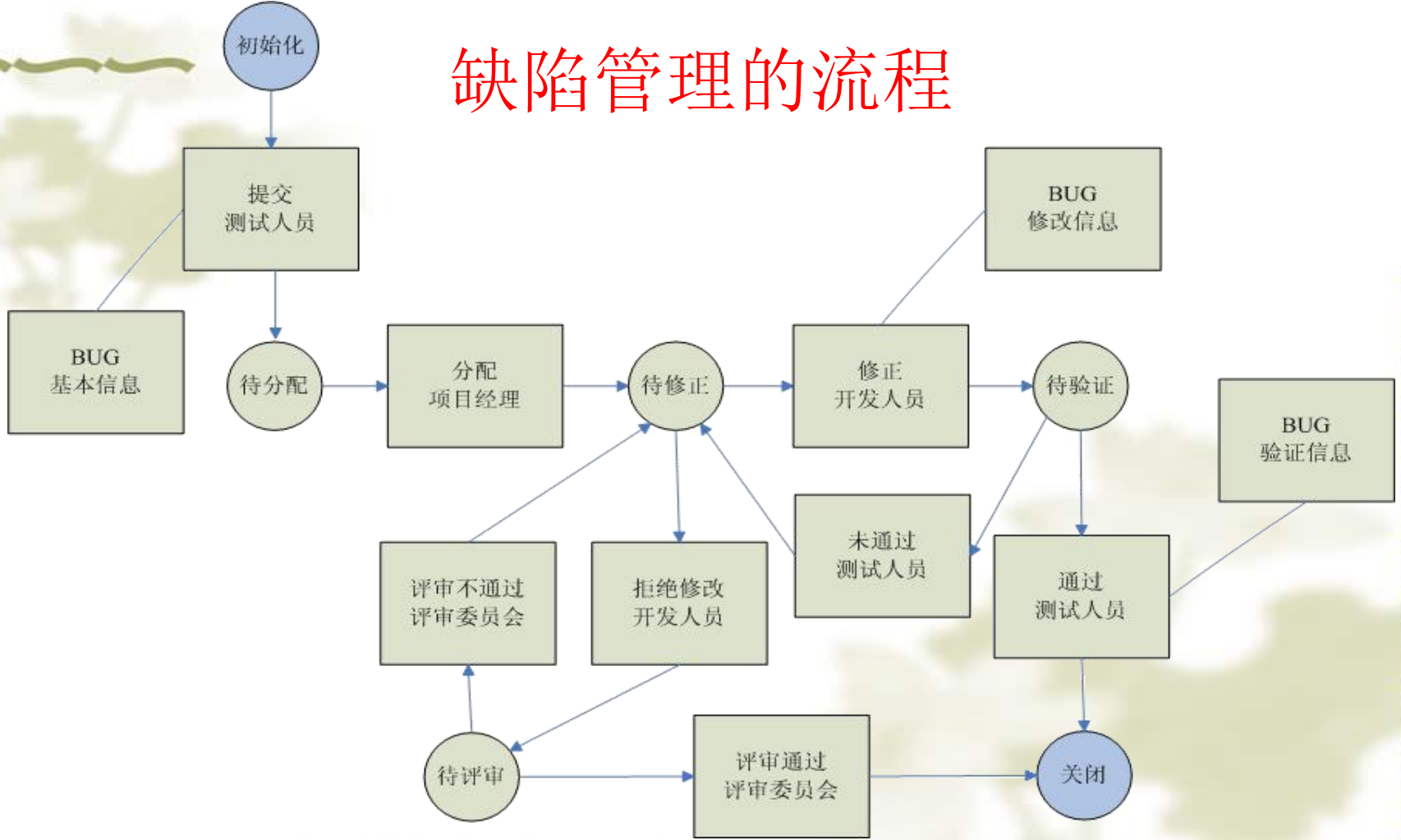
4.缺陷管理

- 软件中的缺陷（Defect或Bug）是软件开发过程中的“副产品”。通常，缺陷会导致软件产品在某种程度上不能满足用户的需要。

- 一般而言，缺陷的跟踪和管理需要达到以下两个目标：一是确保每个被发现的缺陷都能够被解决，二是收集缺陷数据并根据缺陷趋势曲线识别和预防缺陷的频繁发生。

可追踪信息	缺陷ID	缺陷ID唯一，可以根据该ID追踪缺陷。
缺陷基本信息	缺陷状态	缺陷的状态，分为“待分配”、“待修正”、“待验证”、“待评审”、“关闭”。
	缺陷标题	描述缺陷的标题。
	缺陷的严重程度	描述缺陷的严重程度，一般分为“致命”、“严重”、“一般”、“建议”。
	缺陷的紧急程度	描述缺陷的紧急程度，从1到4，1是优先级最高的等级，4是优先级最低的等级。
	缺陷类型	界面缺陷、功能缺陷、安全性缺陷、接口缺陷、数据缺陷、性能缺陷等。
	缺陷提交人	缺陷提交人的姓名和邮件地址。
	缺陷提交时间	缺陷提交的时间。
	缺陷所属项目/模块	缺陷所属的项目和模块，最好能精确到模块。
	缺陷指定解决人	缺陷指定的解决人，在缺陷“提交”状态为空，或在缺陷“分发”状态下，由项目经理指定相关开发人员修改。
	缺陷指定解决时间	项目经理指定的开发人员修改此缺陷的期限。
	缺陷处理人	最终处理缺陷的处理人。
	缺陷处理结果描述	对处理结果的描述，如果对代码做了修改，要求在此处体现出修改。
	缺陷处理时间	缺陷处理的时间。
	缺陷验证人	对被处理缺陷验证的验证人。
	缺陷验证结果描述	对验证结果的描述（通过、不通过）。
	缺陷验证时间	对缺陷验证的时间。
缺陷详细描述		对缺陷的详细描述；对缺陷描述的详细程度直接影响开发人员对缺陷的修改，描述应尽可能的详细。
测试环境说明		对测试环境的描述。
必要的附件		对于某些文字很难表达清楚的缺陷，使用图片等附件是必要的。

缺陷管理的流程


初始化
提交
测试人员
BUG
基本信息
待分配
分配
项目经理
待修正
修正
开发人员
BUG
修改信息
待验证
BUG
验证信息
未通过
测试人员
通过
测试人员
评审不通过
评审委员会
拒绝修改
开发人员
待评审
评审通过
评审委员会
关闭

缺陷管理系统

- 记录缺陷
- 统计缺陷
- 跟踪缺陷的状态
- 监控软件的特性

缺陷统计分析

（1）比较常用的缺陷分析报告，是按照时间段来分析。表明Bug发现随时间的分布情况，一般来说在测试前期和后期发现Bug会比较少，这样的分布是比较正常的。相反，如果是测试后期发现的Bug比较多的话，说明质量控制做的不到位。

（2）按照软件开发的各个阶段，分为需求阶段、设计阶段、代码走查、手动测试、自动化测试。从测试缺陷统计分析中，你能了解到每个阶段出现问题多少，以便以后有针对性的改善。

（3）按照Bug产生的原因，分为需求设计缺陷、功能性缺陷、性能缺陷、安全性缺陷、易用性缺陷及文字界面缺陷。

软件测试成熟度模型

- 随着软件产业界对软件过程的不断研究，美国工业界和政府部门开始认识到，软件过程能力的不断改进才是增进软件开发组织的开发能力和提高软件质量的第一要素。在这种背景下，由美国卡内基-梅隆大学软件工程研究所(SEI)研制并推出了软件能力成熟度模型SW-CMM，CMM逐渐成为了评估软件开发过程的管理以及工程能力的标准。
- 但是令人遗憾的是，CMM 没有充分的定义软件测试，没有提及测试成熟度的概念，没有对测试过程改进进行充分说明。仅在第三级的软件产品工程(SPE)KPA中提及软件测试职能，但对于如何有效提高机构的测试能力和水平没有提供相应指导

软件测试成熟度模型

- 研究机构和测试服务机构从不同角度出发提出有关软件测试方面的能力成熟度模型，作为SEI-CMM的有效补充。
- 美国国防部：CMM软件评估和测试KPA建议;
- Burnstein博士提出了测试成熟度模型(TMM)，依据CMM的框架提出测试的5个不同级别。它描述了测试过程，是项目测试部分得到良好计划和控制的基础。

软件测试成熟度模型

TMM 测试成熟度分解为 5 级别，关注于 5 个成熟度级别递增:

0 :测试和调试没有区别，除支持调试外测试没有其他目的

1 :测试的目的是为了表明软件能够工作

2 :测试的目的是为了表明软件不能够能够正常工作

3 :测试的目的不是要证明什么，而是为了把软件不能正常工作的预知风险降低到能够接受的程度

4 : 测试不是行为，而是一种自觉的约束 (mental discipline) ，不用太多的测试投入，即可产生低风险的软件