

ISTQB初级认证

第5章 测试管理

作者: 郑文强

Email: <u>zwqwwuy@163.com</u>

博客: http://blog.csdn.net/Wengiang 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/Wengiang Zheng/archive/2011/04/09/6311523.aspx

课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

ISTQB考试知识点

- ★认识独立测试的重要性(K1);
- ★列出在组织内进行独立测试的优点和缺点 (**K2**);
- ★考虑使用不同团队的成员来成立测试小组 (**K1**);
- ★了解测试负责人和测试员的任务(**K1**);

建立、组织和管理一支优秀的测试团队是做好软件测试工作的基础,也是测试管理的重要工作之一;

不同的组织,可能会以不同的方式组织测试团队,来达到组织和项目层面的不同的测试目标;

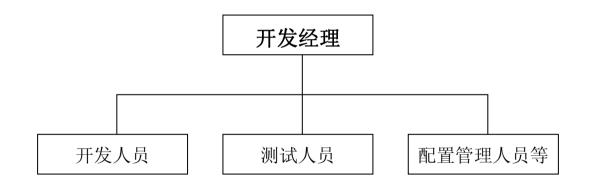
不同的测试组织架构模式

模式1: 开发团队负责测试



开发人员采用交叉测试的方式,即开发人员不对自己的程序进行测试。这种方式也叫伙伴测试或者结对测试!

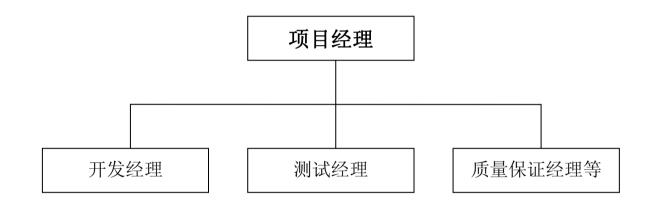
模式2: 开发团队中的测试人员



与模式1类似,但是模式2中的测试人员,负责项目的测试任务,而不仅仅是结对测试的方式。开发人员和测试人员都向开发经理汇报!

测试管理

模式3:项目团队中的测试人员



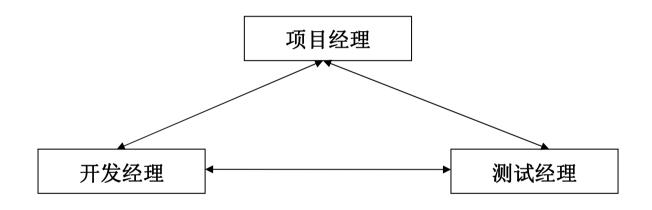
项目团队里专门配备测试团队,并且不负责开发任务,他们对整个项目的测试任务负责。开发经理和测试经理同时向项目经理汇报工作!

测试管理

模式4: 不同测试类型的测试专家

- ★性能测试
- ★可用性测试
- ★安全性测试
- **★**兼容性测试
- ★稳定性测试
- *****

模式5:独立的测试团队



项目经理、开发经理和测试经理三国鼎立。测试团队具有独立性、权威性!

独立测试的优点

- ★独立的测试团队没有偏见,能够找到开发人员未能找到的缺陷;
- ★独立的测试团队能够验证开发人员在设计和实现 系统时所作的(隐式)假定和设想;
- ★独立的测试团队可以避免来自项目经理的非正常 的干预;

独立性、权威性

独立测试的缺点

- ★由于测试团队独立于开发团队,可能会与开发团队缺乏沟通;
- ★如果测试人员没有必要的资源,那么独立测试就有可能成为瓶颈;
- ★测试如果作为项目质量的最后检查点,可能成为项目的瓶颈;
- ★开发人员可能对质量有所松懈,因为他们会认为 测试人员总会找到问题;

组件测试的模式选择

- ★建议选择模式**1**和模式**2**: 开发团队负责测试和 开发团队中的测试人员;
- ★避免开发人员自己测试自己开发的程序,因为他们容易忽视自身所犯的错误;
- ★测试人员本身是开发人员,不存在测试与开发之间的沟通问题;
- ★可能存在的风险:参与人员认为自身本质上是开发人员,因而将忽视他们所要担当的测试责任;

集成测试的模式选择

- ★假如系统集成和组件集成的团队和组件的开发团队是同一团队时,可以采用类似组件测试的模式,即模式**1**和模式**2**;
- ★假如需要集成来自多个团队开发的组件,测试团队成员应该来自多个开发团队,以避免各自对自己开发的组件的"偏爱",或者采用独立的测试团队。可以采用模式3、模式4和模式5的方式:

系统测试的模式选择

★最终产品应该从客户和最终用户的角度去衡量, 因此,和开发保持独立至关重要;

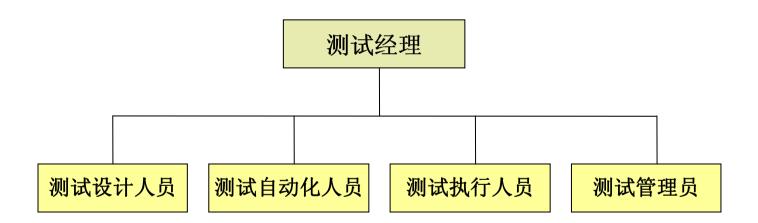
★因此,系统测试中采用模式**3**至模式**5**是专业的、可接受的选择:

验收测试的模式选择

★最终产品的验收,应该从客户和最终用户的经常 使用的场景或者用例的角度去衡量;

★因此,验收测试中采用模式**4**和模式**5**是专业的、可接受的选择;

测试组织的角色



测试经理的职责

- ★测试计划和测试控制方面的专家,具备软件测试、 质量管理、项目管理和人员管理等领域的知识和 经验;
- ★参与编写或者评审组织内部的测试方针;
- ★编写项目的测试策略和测试计划,选择合适的测试策略和测试方法;
- ★作为项目测试方面的代表,计划和获取合适的测试资源;
- ★发起和监控测试工作,即对整个测试过程中的测试活动进行监控;

测试经理的职责(续)

- ★引入合适的测试度量,以测量测试过程效率并评估测试过程和产品的质量;
- ★选择和引入合适的测试工具,并组织必要的工具 使用培训;
- ★确定测试环境和测试自动化的类型和范围;
- ★ 计划测试运行,并根据测试结果和项目测试进展,调整和更新测试计划和资源;
- ★收集和分析测试过程中的相关测试数据,为测试过程改进和测试质量改进提供信息;
- ★编写并提交测试报告;

测试设计人员的职责

- ★测试方法和测试规格说明方面的专家,具备软件测试、软件工程以及(形式化)规格说明方法等领域的知识和经验;
- ★分析、评审和评估用户需求、规格说明、设计和模型等内容,以便设计测试用例;
- ★创建规格说明,准备和获取测试数据;
- ★ 协助测试经理制订测试策略和测试计划;

测试自动化人员的职责

- ★测试自动化专家,具备测试基础知识、编程经验 以及丰富的测试工具和脚本语言知识;
- ★ 搭建和规划测试自动化平台,协助测试经理评估和引入相应的自动化测试工具;
- ★自动化测试需求分析,利用项目中提供的脚本语言等测试工具,设计自动化测试计划和自动化测试用例;
- ★对自动化测试执行情况进行分析,完善自动化测 试框架:

测试管理员的职责

- ★安装和操作测试环境方面的专家(具备系统管理员知识)。建立和支持测试环境(需要经常与系统管理员和网络管理员进行协调);
- ★规划和搭建测试实验室,建立、设置和维护测试 环境所需的应用服务器或软件平台;
- ★负责测试环境所需的网络规划和建设,维护网络的正常运行;
- ★对实验室的硬件、软件资源进行登记、分配和管理;

测试管理员的职责(续)

- ★申请所需求的、新的硬件资源、软件资源,协助有关部门进行采购、验收;
- ★对使用实验室的硬件、软件资源的权限进行设计、 设置,保证其安全性;
- ★优化测试环境,提高测试环境中网络、服务器和 其它设备运行的性能;

测试管理

测试执行人员的职责

- ★ 执行测试和事件报告方面的专家(具备**IT**基础知识、测试基础知识,会应用测试工具,熟悉测试对象);
- ★分析、评审和评估用户需求、规格说明及可测试性,以及评审测试规格说明;
- ★建立测试环境(通常需要与系统管理员,网络管理员协同完成),根据要求使用测试管理工具和测试监控工具;
- ★进行各种级别的测试,执行并记录测试日志,评估测试结果,记录和预期结果之间的偏差;

课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

测试计划和估算

ISTQB考试知识点

- ★认识测试计划的不同级别和目标(K1);
- ★根据"软件测试文档标准(IEEE 829)"总结《测试计划》、《测试设计规格说明》 和《测试规程》的目的及内容(K2);
- ★区分属于二类不同概念(预防型和应对型)的各种测试方法,如基于分析、基于模型、基于方法、符合过程/标准的、动态/启发式的、咨询式或基于面向可重用的方法(**K2**);
- ★区分为系统而做测试计划和为安排测试执行做测试计划的内容上的不同之处 (**K2**);

测试计划和估算

ISTOB考试知识点(续)

- ★在综合考虑优先级、技术和逻辑依赖后,为给定的测试用例集编写测试执行计划 (**K3**);
- ★列出在测试计划时应该考虑的测试准备和执行活动(**K1**);
- ★认识影响测试开销的主要因素(K1);
- ★ 从概念上区别两种不同的估算方法:基于度量的 方法和基于专家的方法(**K2**);
- ★理解/解释针对特定测试级别和测试用例组所定义的恰当的出口准则(例如对于集成测试、验收测试或可用性测试的测试用例)(**K2**);

<u>质量保证计划IEEE</u> 730

测试不应该是质量保证QA的唯一措施,而应该与其他质量保证 措施相结合,并在质量保证计划中体现。

质量保证计划 (依据[IEEE 730])

- 1. 目的
- 2. 参考文档
- 3. 管理
- 4. 文档化
- 5. 标准、实践、惯例和度量
- 6. 软件评审
- 7. 测试
- 8. 问题报告和修正行为
- 9. 工具、技术和方法学
- 10. 媒介控制
- 11. 供应商控制
- 12. 记录收集、维护和保存
- 13. 培训
- 14. 风险管理
- 15. 术语表
- 16. SQAP变更规程和历史

质量保证QA和质量控制QC

质量保证QA

- ★关注的是过程
- ★目标是预防缺陷
 - + 过程定义
 - + 过程改进
 - **+**



质量控制QC

- ★关注的是产品
- ★目标是发现缺陷
 - + 静态测试
 - + 动态测试
 - +

测试计划的活动

- ★确定测试的范围和风险,明确测试的目标;
- ★定义测试的整体方法(测试策略),包括测试级 别的定义、入口和出口准则的定义;
- ★把测试活动集成和协调到整个软件生命周期活动中去: 收集、准备、开发、运行和维护;
- ★决定测试什么?测试由什么角色来执行?如何进行测试?如何评估测试结果?
- ★ 为测试分析和设计活动安排时间进度;

测试计划的活动 (续)

- ★ 为测试实现、执行和评估安排时间进度;
- ★为已定义的不同测试任务分配资源;
- ★定义测试文档的数量、详细程度、结构和模板;
- ★ 为测试准备和执行的监控、缺陷解决和风险问题 选择度量项;
- ★ 确定测试规程的详细程度,以提供足够的信息支持可重复的测试准备和执行;

测试计划的持续性

测试计划对于测试经理来说是贯穿于整个生命周期的持续性活动。必须考虑测试活动的反馈并识别不断变化的风险的基础上,定期地更新测试策略和相关的计划!

测试管理

测试计划的模板

测试计划 (依据[IEEE 829])

- 1. 测试计划标识
- 2. 介绍
- 3. 测试项
- 4. 需要测试的特性
- 5. 不需测试的特性
- 6. 测试方式
- 7. 测试项通过/失败准则 (测试出口准则)
- 8. 挂起准则和恢复准则
- 9. 测试交付物
- 10. 测试任务
- 11. 环境要求
- 12. 责任
- 13. 人员配备和培训要求
- 14. 进度表
- 15. 风险和意外事件
- 16. 审批

测试计划例子描述

该示例中采用的工资系统包括以下主要功能:

- a) 维护雇员信息;
- b) 维护工资历史信息;
- c) 准备工资支付凭证;
- d) 准备工资税务报表;
- e) 准备工资支付报表。

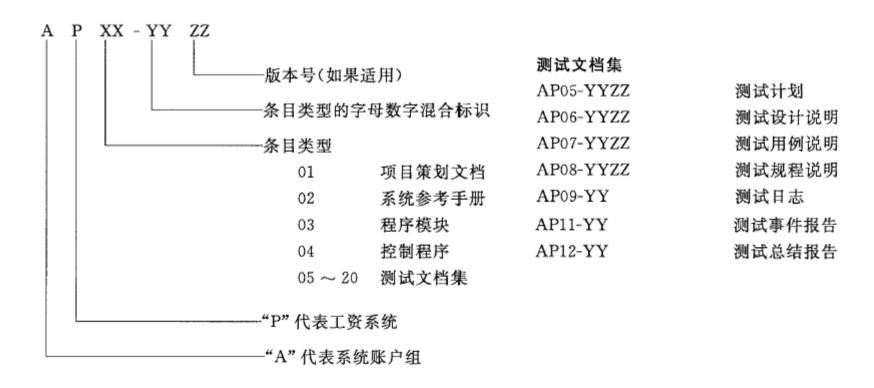
已提出用于工资系统的二期开发计划,不久后将要启动。这个阶段主要涉及人员报表系统。

在准备该示例时,曾作出以下假定:

- a) 系统测试活动假设模块测试和集成测试已经完毕。它意味着单个程序的功能已经过全面的测试。因此,系统级测试集中在对多个程序块的功能性(例如:年终处理)以及对外部接口、安全、恢复和性能的测试。此外,操作者和用户规程也经过测试;
- b) 工资系统应是仅在一个地点被测试的系统。

<u>标识符: AP05-0103</u>

★用来唯一的标识测试计划文档的名称和版本。



介绍

- ★提供了项目背景的简短概述。其目的是帮助参与项目的人员(客户、管理者、开发人员以及测试人员)能够更好地理解测试计划的内容;
- ★在测试计划的介绍部分,应该包含引用到的所有 文件列表。通常包括一些策略和标准,比如行业 标准、公司标准、项目标准、用户标准、项目授 权(有可能为合同)、项目计划及其他计划、规 格说明等;

介绍(例子)

2.1 目标

公司工资系统的系统测试计划应该支持以下目标:

- (1) 细化准备和进行系统测试所需要的活动。
- (2) 与所有负责方沟通有关他们要执行的任务以及执行任务时所安排的进度。
- (3) 确定用来准备计划的信息源。
- (4) 确定进行系统测试所需要的测试工具和环境。

2.2 背景

去年,XYZ公司系统和程序开发部门应公司会计部门的要求开发了一个新的通用总帐系统。与此同时,还提出要求要开发一个与该通用总帐系统接口的新的公司工资系统。

管理层系统评估委员会在 19 * * 年 9 月批准了开发工资系统的请求,并且指定一个工资系统顾问组来确定系统需求。顾问组于 19 * * 年 12 月完成了一份需求陈述(AP01-01)和一份初步开发计划。

介绍(例子续)

2.3 范围

该测试计划覆盖了公司工资系统的全部系统测试,包括操作者和用户规程、以及程序和作业控制。除了综合性多程序功能性测试外,还应评估外部接口、安全、恢复和性能。

2.4 引用文件

下列文档用作该测试计划的信息源:

公司工资系统初步开发计划(AP01-02)

公司工资系统授权(AP01-03)

公司工资系统最终开发计划(AP01-06)

公司工资系统质量保证计划(AP01-08)

公司工资系统配置管理计划(AP01-09)

XYZ公司系统开发标准及规程(XYZ01-0100)

公司通用总帐系统设计描述(AG01-04)

公司通用总帐系统测试计划(AG05-01)

测试对象/测试项

- ★这一部分应该包含被测产品部分及其组件的简单概述、标识包含版本和修订级别的测试条目、详细说明传输介质的特性和规格说明;
- ★为了避免误解,还应该包含不被测试的项目列表;

测试对象/测试项(例子)

组成公司工资系统的所有项在系统测试期间应予测试。待测试的版本应由配置管理员放在合适的库中。管理员还应控制对受试版本的更改,并且将可提供新版本的时间通知测试组。

以下文档为规定正确的操作建立基础:

公司工资系统需求规格说明(AP01-01)

公司工资系统设计描述(AP01-04)

公司工资系统参考手册(AP02-01)

公司工资系统模块参考手册(AP02-03)

要测试的各项列出如下:

3.1 程序模块

要测试的程序模块按以下规则来标识:

类型	<u>库</u>	成员名称
源代码	SOURLIB1	AP0302
		AP0305
可执行代码	MACLIB1	AP0301
		AP0302

AP0305

测试对象/测试项(例子续)

3.2 作业控制规程

应用程序、分类和实用程序的控制规程标识如下:

<u>类型</u>	<u>库</u>	成员名称
应用程序	PROCLIB1	AP0401
分类	PROCLIB1	AP0402
实用程序	PROCLIB1	AP0403

3.3 用户规程

公司工资系统用户事务参考手册(AP02-04)中规定的在线规程应予测试。

3.4 操作者规程

系统测试包括公司工资系统操作参考手册(AP02-02)中规定的规程。

需要测试的特征

- ★这一部分应该确定系统中所有应该被测试的功能 和特征;
- ★这一部分应该参考测试规格说明和更多相关的描述,以及相对应的测试等级或测试阶段;

需要测试的特征(例子)

以下清单列出待测试的特征:

测试设计

说明编号 描述

AP06-01 数据库转换

AP06-02 月薪雇员全面的工资处理

AP06-03 计时雇员全面的工资处理

AP06-04 所有雇员全面的工资处理

AP06-05 定期报告

AP06-06 通用总帐事务的建立

AP06-07 安全 AP06-08 恢复

AP06-09 性能

不需要测试的特征

- ★为了避免误解和防止不切实际的预期,应该定义 产品中哪些是不需要或者无法进行测试的(这可 能是由于资源限制或者技术原因所导致的);
- ★针对不同的特性,也可能需要不同的测试级别;

不需要测试的特征(例子)

下列特征不应包括在系统测试中,因为它们在系统初始安装时不会使用。

平等就业机会委员会符合性报告

内部培训进度报告

工资/业绩审查报告

二期开发阶段文档集应包含关于这些特征的一个测试计划。

测试用例将不会覆盖正在受试的事务或者报告中所有可能的选项组合。只有目前 XYZ 公司工 资处理明确需求的组合应予测试。

测试方法和策略

- ★如果可能,应根据风险分析来描述测试目标。风险分析应该显示:假如缺少测试导致错误无法发现,将会产生哪些风险。根据这些信息,就可以得到哪些测试项是必须测试的,以及相关的重要程度。这样就可以确保将测试重点放在重要的测试项上;
- ★根据上面的分析,选择和描述将要使用的测试方法。根据标识出的风险和可用资源情况,必须清楚地阐明所选择的方法是否可以达到测试目标以及为什么能够达到目标;

测试方法和策略(例子)

6 方法

测试人员应根据系统文档集准备所有的测试设计、用例以及规程说明。这种方法应验证测试所覆盖那些领域的文档集信息的准确性和综合性。

公司工资和会计部门的人员应协助开发测试设计和测试用例,这样做有助于确保测试能体现系统的实际使用。

为了确保保密性,从会计文件中选取的所有测试数据应含有已更改的保密敏感字段。

6.1 转换测试

除了计算输入和输出的记录外,转换数据库的有效性应以两种方式进行验证。第一种验证方法 涉及到使用必须由开发组建立的"数据库审核员"功能。当针对被转换数据库运行时,数据库审核员 应核对一条记录内的数值范围,以及要求的各条记录之间的关系。

第二种验证方法涉及到随机选取旧记录的一个小的子集,然后直接与新记录的相对应子集进行比较。直接比较的数目"c"和旧记录的数目"r"必须加以规定。从1到r的范围内产生由随机数字组成的c集合。在转换过程中,该集合应予以分类和应用,以驱动对直接比较记录的选择。

注:同样的两种验证方法在实际的转换期间应予采用。

测试方法和策略 (例子续)

6.2 作业流测试

月薪雇员和计时雇员的记录综合集以及这两种记录的合并集应用于测试工资处理。标准的作业 流测试方法应予采用。

每种定期报告作业流至少运行一次。

6.3 接口测试

为了测试工资系统与通用总帐系统之间的接口,工资系统应建立一个通用总帐事务综合集。这 些事务应输入到通用总帐测试系统。生成的通用总帐条目必须加以选取、打印并与由工资系统准备 的通用总帐事务的打印输出相比较。

6.4 安全测试

无妥当口令但又试图访问在线数据条目并显示事务的情况应予测试。

6.5 恢复测试

在可单独运行的时间内,通过停机且随后依照恢复规程进行恢复测试。

6.6 性能测试

依据性能要求(AP01-01),通过利用产生的数据量测量若干作业的运行时间,以此来评估性能。

验收准则

- ★当针对测试对象的所有相关测试执行完成之后,需要根据测试结果来决定是否可以发布和交付测试对象。为了达到这个目的,就必须定义验收准则或者测试出口准则;
- ★针对不同的测试对象,验收准则可以有所不同。 验收准则的完整性依赖于对风险的分析;

验收准则 (例子)

该系统必须符合 XYZ 公司系统开发标准和规程(XYZ01-0100)中陈述的系统通过/失败的标准需求。该系统还必须满足下列需求:

- ——内存需求一定不要大于真实存储量 64k。
- ——用户规程与其他会计系统的一致性必须使工资主管满意。

挂起准则和恢复准则

- ★除了验收准则之外,还需要定义挂起准则和恢复 准则;
- ★ 挂起准则:可能会导致测试执行挂起(非正常中止)的状态和事件;
- ★恢复准则:可以继续或者重新进行测试的状态和事件;

挂起准则和恢复准则(例子)

8.1 暂停准则

不能转换雇员信息数据库会导致所有测试活动的暂停。

8.2 恢复要求

出现测试暂停后,当系统的新版本向测试组传递时,6.7条中描述的回归测试应予执行。

测试交付物

- ★确定每个测试活动需要交付的数据和结果,以及 以何种方式来沟通这些测试结果;
- ★测试交付物并不仅仅指狭义上的测试结果(如事件报告和测试协议),同时也包含计划和准备方面的文档,如测试计划、测试规格说明、时间进度表、用以描述测试对象移交的文档以及测试总结报告;

测试交付物 (例子)

9 测试交付项

系统测试组应形成下列文档,这些文档在测试结束后交付给配置管理组。

测试文档

系统测试计划;

系统测试设计说明;

系统测试用例说明;

系统测试规程说明;

系统测试日志;

系统测试事件报告日志:

系统测试事件报告;

系统测试总结报告。

测试任务

- ★针对测试计划和执行的所有任务列表,包括责任 分配;
- ★跟踪所有任务的状态(开始、进行中、延迟、完成)。这是正常项目计划和跟踪的组成部分,因此需要在项目或测试状态报告中定期汇报;

测试任务 (例子)

附件 A 任务列表

任 务	前期任务	特殊技能	责 任	投入	完成日期
(1)准备测试计划	结束工资系统设计描述(AP01-04)和初步的开发计划(AP01-02)	_	测试经理; 高级测试分析员	4 (人日)	××-01-21
(2)准备测试设计说明	任务 1	通晓公司的 工资规程	高级测试分析员	9	××-04-01
(3)准备测试用例说明	结束相应的测试 设计(任务 2)		测试分析员	4	××-04-15
(4)准备测试规程说明	结束相应的测试 用例(任务 3)	_	测试分析员	6	××-05-15
(5)建立最初的雇员信息 数据库	任务 4		测试分析员	6	××-06-01
(6)结束测试项传递并向 测试组传递该公司的工 资系统	结束集成测试		开发项目经理		××-06-01
(7)检查执行该系统需要 的所有工作控制规程	任务 6	工作控制经验	测试技术员	1	××-06-08
(8)组装并链接该公司工 资系统	任务 6		测试技术员	1	××-06-08
(9) 执行数据录入测试 规程	任务 5 任务 8		测试分析员	1	××-06-22
(10)执行批测试规程	任务 5 任务 8	_	测试技术员	3	××-06-30
(11)检查批测试结果	任务 10	通晓工资报 告需求	测试分析员	1	××-07-02
(12)解决测试事件报告	任务 9 任务 11		开发组经理; 系统测试组经理; 公司的工资部门 经理	2	××-07-16
(13)重复任务(6)~(12) 直到所有测试规程成功 运行			_	2	××-07-30
(14)撰写系统测试总结 报告	任务 13	_	系统测试组经理; 公司工资部门经理	1	××-08-06
(15)将所有测试文档集 和测试数据传输给配置 管理组			系统测试组	1	××-08-06

测试环境

★列举出了执行计划中的测试所必需的测试基础设施元素。常见的有:测试平台、测试人员的工作场所以及他们的设备、测试工具、开发环境或测试人员所需的部分环境以及其他的一些工具(电子邮件、万维网WWW、Office软件包等):

测试环境(例子)

11 环境要求

11.1 硬件

测试应在 XYZ 公司的硬件配置下进行。

鉴于大多数测试必须在主要的操作时间内开展,在此期间内应向测试组提供3个在线终端。

11.2 软件

11.2.1 操作系统

该业务操作系统应用于执行这些测试。

11.2.2 通信软件

所有在线程序应在测试通信软件的控制下加以测试。

11.3 安全性

安全性应限于现有的各种控制器。

11.4 工具

开发和评估系统测试需要下列测试工具:

- (1) 测试数据生成器(UT09-0200)。该程序用于生成绝大多数的测试数据。它位于标准系统 库 SYSLIBA。
- (2) 比较器程序(UT08-0100)。该程序用于在回归测试期间比较系统结果。它位于标准系统 库 SYSLIBA 中。
- (3) 数据库审核器。该程序用于审核数据库中的数值范围及记录之间的相互关系。它须由开 发组提供。

职责

- ★关于项目的测试是如何组织的?
- ★ 谁有什么样的权限和责任?
- ★测试人员可能必须分到不同的测试小组或测试级别中,给什么样的人分配什么样的任务?

职责(例子)

12 职责

下列各组对测试各部分负有责任。

12.1 系统测试组

该组对测试及技术测试业务进行全面管理。

12.2 公司工资部门

该组是公司工资系统的终端用户,在下列各项活动中应协助系统测试组工作。

- -----审查测试设计说明;
- ---执行在线测试;
- ——校验输出屏幕和报告。

12.3 项目开发组

该组传递要测试的系统,并响应系统测试事件报告。该组对需要排错的任何程序进行调试,并提供数据库审核器。

人员和培训要求

- ★ 该部分要指定人员配备的要求(角色、资质、数量、何时需要他们以及计划假期等等)。该计划不只是为测试人员而制定的,同时也针对管理测试基础设施的人员、开发人员、每种工具和软件产品(例如:数据库系统)以及在测试过程中需要对外接口的产品的用户和咨询人员;
- ★同时也必须包含提供所需技术的培训;

人员和培训要求 (例子)

- 13 人员和培训要求
- 13.1 人员配备

需要下列人员开展该测试项目:

13.1.1 测试组

测试经理

高级测试分析员

测试分析员 2

测试技术员 1

13.1.2 工资部门

工资监管人员 1

13.2 培训

公司工资部门的人员必须经过培训,以便对数据录入事务进行处理。用户事务参考手册(AP02-04)应作为该培训的基础。

进度

- ★描述了带有主要里程碑的全面测试活动的进度表,该计划必须与项目计划中的一致,并需维护以保证相同;
- ★项目经理和测试经理之间的定期磋商是必需的。 在开发过程中,测试经理应该被告知有关项目延期的消息,并更新具体的测试计划。项目经理根据测试结果来采取措施;假如需执行额外的修改和增加测试周期,如有必要,可延迟里程碑时间点。如果有任何测试资源需要共享,比如测试实验室,则必须清晰明确地写在进度表里;

进度(例子)

附件 A 任务列表

	1	T A 11.37997			
任 务	前期任务	特殊技能	责 任	投入	完成日期
(1)准备测试计划	结東工资系统设计 描述 (AP01-04)和初步的开发计划(AP01-02)		测试经理; 高级测试分析员	4 (人日)	××-01-21
(2)准备测试设计说明	任务 1	通晓公司的 工资规程	高级测试分析员	9	××-04-01
(3)准备测试用例说明	结束相应的测试 设计(任务 2)		测试分析员	4	××-04-15
(4)准备测试规程说明	结束相应的测试 用例(任务 3)	_	测试分析员	6	××-05-15
(5)建立最初的雇员信息 数据库	任务 4	_	测试分析员	6	××-06-01
(6)结束测试项传递并向 测试组传递该公司的工 资系统	结束集成测试		开发项目经理		××-06-01
(7)检查执行该系统需要 的所有工作控制规程	任务 6	工作控制经验	测试技术员	1	××-06-08
(8)组装并链接该公司工 资系统	任务 6		测试技术员	1	××-06-08
(9)执行数据录入测试 规程	任务 5 任务 8	_	测试分析员	1	××-06-22
(10)执行批测试规程	任务 5 任务 8	_	测试技术员	3	××-06-30
(11)检查批测试结果	任务 10	通 晓 工 资 报 告需求	测试分析员	1	××-07-02
(12)解决测试事件报告	任务 9 任务 11		开发组经理; 系统测试组经理; 公司的工资部门 经理	2	××-07-16
(13)重复任务(6)~(12) 直到所有测试规程成功 运行			_	2	××-07-30
(14)撰写系统测试总结 报告	任务 13	_	系统测试组经理; 公司工资部门经理	1	××-08-06
(15)将所有测试文档集 和测试数据传输给配置 管理组	任务 14		系统测试组	1	××-08-06

风险和应急

- ★主要针对测试项目本身来描述存在的风险,比如 实现测试概念时的风险,以及在实际项目中由于 没有资源致使不能完成合理的活动所引发的风 险;
- ★ 这部分最低限度应该输出一份风险列表,并且在 某些时间点上进行监控,以便寻找方法使风险最 小化:

风险和应急(例子)

15 风险和应急

如果系统故障严重地影响测试进度,开发经理已同意分派一名全职人员到测试组做调试工作。如果一位工资监管人员对于测试工作不够用,工资经理已同意确定第二位监管人员。如果硬件出现的问题影响系统在白天的应用,则测试组应安排其夜晚的活动。

公司工资系统的第一次运行使用,在分发工资支票前必须详细地加以核查,并对任何出错的支票须用手工进行修正。

批准

★相关人员签字;

批准 (例子)

16 批准

测试经理: 日期:

开发项目经理: 日期:

质量保证经理: 日期:

在有限的测试资源和时间的情况下,尽早尽快的在测试对象中查找尽量多的缺陷。因此,如何有效的选择测试用例,是测试计划制订中面临的一个重要环节!



测试用例优先级

测试优先级的划分

- ★测试时间和资源有限,可能无法执行所有的测试 用例,穷尽测试是不可能的;
- ★ 首先执行最重要的测试用例,尽可能早的发现尽量多的缺陷,或者满足客户最需要的功能;
- ★测试优先级的划分和测试执行顺序的确定,取决于项目的特征、应用领域和客户的要求:

测试优先级的划分

★即使测试过早结束,仍然能够保证在该时刻测试 能达到最佳效果:

★最重要的测试用例将首先被执行,这样就可以尽 早发现重要的问题;

测试优先级划分准则

- ★使用频率或失效的概率:如果系统的某些特定功能经常被使用并且其中包含了故障,那么这个故障导致失效的概率就会很高。因此,用于此功能的测试用例应该比某个较少使用的功能的测试用例具有更高的优先级;
- ★失效的风险:风险是严重程度和失效概率的综合结果(数学乘积)。例如,风险可能是因使用软件的客户的业务受到了(失效的)影响,导致了财产损失。为找到高风险失效而设计的测试用例应该比为找到低风险失效而设计的测试用例具有更高的优先级:

测试管理

划分测试优先级的准则(续)

- ★失效的可见性:失效对客户的可见性,是划分测试用例优先级的更进一步的准则。这在交互式系统中尤为重要。例如,如果城市信息服务系统的用户界面上存在问题,那么用户会感到不安全,并对其他信息输出不再信任;
- ★需求的优先级:系统提供的不同功能对于客户来说其重要性也不尽相同。如果某些功能不能正常工作,客户也许能够接受这样的功能缺失,而其他部分则不可或缺;

划分测试优先级的准则(续)

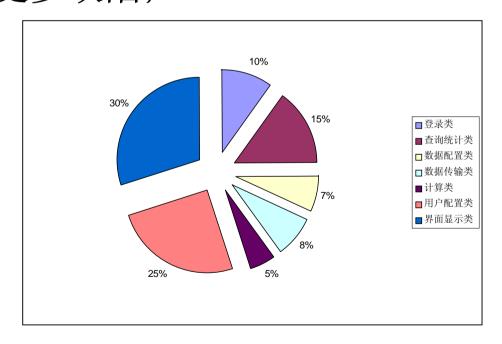
- ★ 质量特性:除了功能需求,质量特性对于客户也具有不同的重要性。必须要测试重要的质量特征是否已正确实现。用于验证与必要的质量特征是否一致的测试用例具有高优先级;
- ★开发人员角度: 划分优先级也可以从系统架构的 开发人员的角度来完成。失效时导致严重后果, 比如系统崩溃的组件需要加强测试;

划分测试优先级的准则(续)

- ★测试对象的复杂性:复杂的程序部件需要加强测试,因为开发人员可能在此引入了较多的缺陷。不过,看起来简单的程序部件也可能会包含很多缺陷,这是由于开发不够细致造成的。因此,对这个领域划分优先级时,应该参考从组织中早期项目得来的经验数据;
- ★高项目风险的失效:存在高项目风险的失效应该尽早被发现。这些失效需要做大量的修正工作,会独占资源并导致项目明显延迟;

划分测试优先级的准则(续)

★缺陷的集群效应: 在先前发现缺陷的地方,可能 会出现更多缺陷;



测试优先级的优点

- ★为每个测试用例设置合理的测试优先级,在测试 资源有限的时候,可以首先执行测试优先级高的 测试用例,从而达到成本、质量等的平衡;
- ★根据前面版本发现的缺陷的分布情况,合理调整 测试用例的优先级,可以更加高效的分配测试资 源;

测试准则

测试准则的类型

- ★测试入口准则
- ★测试出口准则
- ★测试挂起准则
- ★测试恢复准则

测试入口准则

测试入口准则定义:允许一个系统进入一个特定的测试阶段时所必须具备的条件!

测试入口准则

测试入口准则例子

- ★概要测试用例、详细测试用例文档评审通过;
- ★ 自动化测试用例在实际测试环境中验证通过;
- ★相关的测试资源和测试环境准备就绪,包括人员、 工具、实验室等;
- ★开发人员对提交的版本进行了预测试并且预测试通过:
- ★开发人员提交版本说明,包括本版本中新增加的功能特性、修改的缺陷、没有修改的缺陷、可能存在的问题以及测试重点的建议等;

测试出口准则

测试出口准则定义:允许一个系统完成一个特定的测试阶段时所必须具备的条件!

测试出口准则定义的目的在于定义什么时候可以停止测试(全部测试或针对某个测试级别);

测试管理

测试出口准则

测试出口准则考虑的因素

- ★测试覆盖率:有多少测试用例已经(成功)运行,有多少需求被覆盖,有多少代码被覆盖;
- ★产品质量: 发现的故障数、失效的危险程度、失效率、可靠性等等;
- ★尚存风险:尚未执行的测试、尚未修复的缺陷、 未完成的需求覆盖率或代码覆盖率等等;
- ★ 经济制约:容许成本、项目风险、交付日期以及 市场机会:

测试出口准则

测试出口准则例子

- ★ 所有在本测试周期所计划的测试用例全部执行完毕, 并且所有的严重程度为1的缺陷修改完毕并通过确认测试;
- ★ 计划的测试时间用完;
- ★ 软件测试达到了预先定义的测试覆盖率;

场景描述

项目经理: 你觉得这个项目的测试时间是否足够? 这个项目对我们公司

很重要,客户要求我们在下个月底之前发布并进行演示,所以

测试时间只有6个星期;

测试经理: 好的,我先看看,然后给你结果。

过了一会。。。。。。

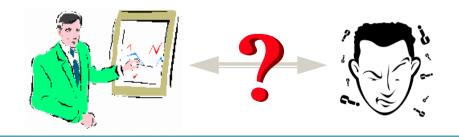
测试经理: 我根据目前我们的资源,估计了一下,这个项目的测试时间

大概需要10个星期:

项目经理: \$%&@!, 你没有听见我前面和你说的信息? 我们必须在6个

星期之内测试完成?

测试经理: \$%&@!



Estimates, Target and Commitments的 关系

- ★测试目标的确立不一定依赖于测试的估算;
- ★测试目标是我们商业目标的一种描述之一;
- ★测试目标一般是强制的,尽管它可能是不容易实现的,或者是不现实的;
- ★而承诺是是对在特定时间内,满足特定质量要求的情况下,对对相关产品或者服务的一种允诺;
- ★承诺可以是和估算结果相同的,也可以是不同的; 的; 理解这些概念,是否可以帮助测试

理解这些概念,是否可以帮助测试 经理更好的回答前面的问题?

场景描述

你作为测试估算方面的专家,参加了一个关于软件测试的年会。在你的周围,是另外3个软件测试估算方面的专家,分别是甲、乙、丙。

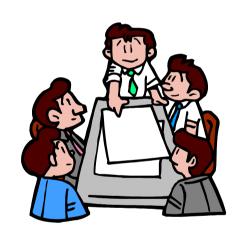
主持人突然拿起话筒说,需要为大会参与人员准备午餐,但是需要统计会议中的人数,请你和另外3个估算专家估算会议厅中大概的人数。。。

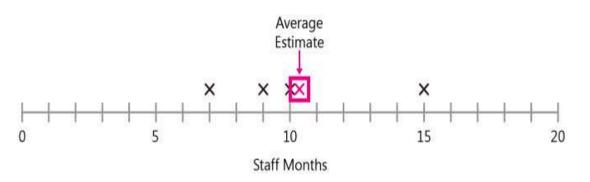


Count, Compute and Judge

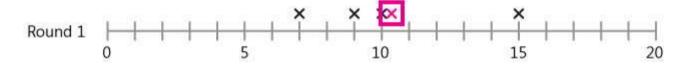
- ★甲:根据我的经验,大概是**335**人;
- ★乙:整个会议室大概有坐位500人,大概在70%
 - 80%, 因此人数为350和400人之间;
- ★丙:大概有**11**排,每排**7**张桌子,每张大概**5** 人,总共大概**385**人:
- **★**我:, 得到的数据是**407**人;

基于团队的估算: ROUND 0

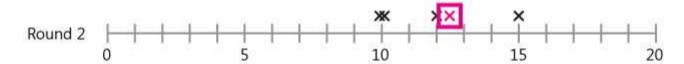


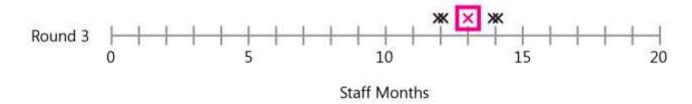


基于团队的估算: ROUND 1/2/3









基于百分比的估算

基于百分比的测试工作量估算的前提条件是得到测试对象研发的总的工作量估算(比如基于代码行KLOC或者功能点FP估算得到)。然后根据项目或者组织在每个开发活动中的经验百分比,之后得到测试相关的工作量估算。!

基于百分比估算的例子

和代码研发相关的工作量: 100PW

和代码不相关的工作量:是步骤1中的25%算为=25PW

总共的项目研发工作量: 125PW

基于百分比的估算



ID	开发活动	百分比%	工作量PW
ID1	用户需求	13.00%	16.25
ID2	概要设计	12.00%	15
ID3	详细设计	15.00%	18.75
ID4	编码和实现	10.00%	12.5
ID5	测试活动	40.00%	50
ID6	项目收尾	10.00%	12.5
Total	0	100.00%	125

测试工作量估算影响因素

- ★产品的特点:测试模型使用的规格说明和其它信息的质量、产品的大小、问题领域的复杂度、可靠性和安全性方面的需求、对文档的需求等;
- ★开发过程的特点:组织的稳定性、使用的工具、 测试过程、参与者的技能水平和时间紧迫程度;
- **★测试的输出:**发现的缺陷数量、需要返工的工作量和输出的测试文档;

测试策略的定义

测试策略定义了软件项目的测试目标和实现方法;

测试策略决定了测试的工作量和测试的成本;

选择合适的测试策略是测试经理做出的一个最重要的计划任务决定,目的在于选择一种测试方式,能够最优化测试成本和缺陷成本之间的关系!

预防性策略

- ★测试人员开始就参与项目:制订测试计划和测试设计尽早开始,测试经理可以有效的优化测试并降低测试成本;
- ★V模型中的Verification,比如评审可以在很大程度上防止缺陷的产生;
- ★尽早进行测试设计和准备,以及应用评审和静态 分析,有助于尽早发现缺陷,从而能降低测试执 行时的缺陷密度;
- ★对于安全关键的软件,预防性策略可能是强制的;

预防性策略的表现

★ 在项目过程中,测试过程应该尽早开始:测试活动应该是平行于开发活动的;

★测试应该持续地伴随项目的所有阶段同时进行, 即测试活动应该贯穿于整个软件的生命周期;

应对性策略

- ★测试人员很晚才参与到项目中,此时已经无法选择预防性方法了;
- ★制定测试计划和测试设计在软件或系统已经开发 后才开始;
- ★测试经理必须做出恰当的应对。比如采用"探索性测试"。这是一种启发式方法,测试人员"探索"测试对象和测试设计,在此过程中,测试的执行和评估几乎是同时进行的:

分析式方法和启发式方法

分析式方法

基于对数据或者数据的分析来制订测试计划!

基于以前项目的丰富的测试相关的数据来优化成本、时间和覆盖率等,比如测试准则的定义!

启发式方法

基于项目内外的专家经验和实际经验来制订测试计划!

以前项目的测试相关数据 没有有效的收集和分析, 得到合适的度量!

测试策略方法的类型

- ★基于模型的测试,使用被测软件的抽象功能模型 进行测试用例设计,以找到测试出口准则,并针 对模型来测量测试覆盖率;
- ★基于模型的统计或随机测试使用测试对象中故障 分布的统计模型、软件使用时的失效率(比如可 靠性增长模型)或用例(比如运行概况)的统计 分布;测试工作量可以根据这些数据来分配;
- ★基于风险的测试使用项目和产品的风险信息,并 将测试集中于风险最大的地方;

测试策略方法的类型(续)

- ★ 遵循过程或标准的方法使用规则、建议和标准, 比如V模型或IEEE 829,作为方法指南;
- ★面向重用的方法重用现有的测试环境和测试资料,目的在于通过最大程度的重用来快速地搭建测试;
- ★基于检查表的(系统化)方法使用早期测试周期的失效和缺陷列表、潜在的缺陷或风险列表、优 先排序的质量准则、或其他非正式的方法;
- ★面向专家的方法使用专门技术和相关专家(在应用过的技术或领域)的建议和经验;

前面介绍的测试策略方法,一般都介于分析式方法 和启发式方法之间,或多或少的综合应用了这两种 方法;

上述方法几乎不会单独使用。一般,测试经理综合使用几种方法来制定测试策略;

测试管理

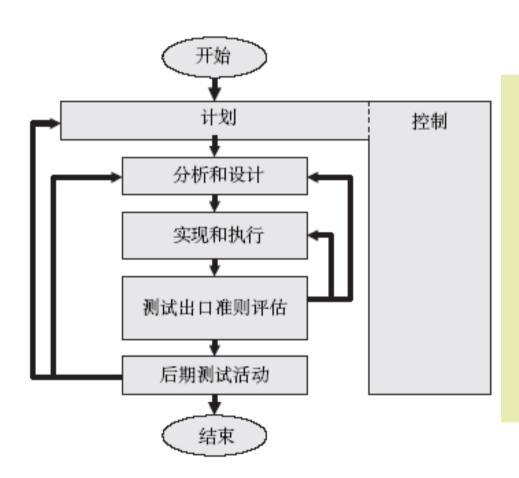
课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

测试进度监控

ISTOB考试知识点

- ★认识用于监督测试准备和执行的常见的测试度量项(**K1**);
- ★理解和解释针对测试报告和测试控制的测试度量 (例如已发现和已修复的缺陷、已通过或失败的 测试)(**K2**);
- ★根据"软件测试文档标准(IEEE 829)"总结测试报告的目的和内容(K2):



由于变更请求或者缺陷修改,使得测试过程的每个阶段出现重复或者循环执行!

测试经理需要通过合适的手段进行测试过程的管理: 计划、监管和控制等!

测试周期的计划

- ★测试计划定义了测试策略和整体进度计划,需要 在项目早期制订;
- ★针对每个测试过程,应该制订详细的测试计划来 补充;

概要测试计划 VS 详细测试计划

测试计划的更新

- ★开发状态:在测试周期开始时,开发人员提供的软件,其功能性和原计划相比可能有所限制或改变。这种情况下可能需要对测试规格说明或者测试用例进行相应的修改;
- ★测试结果: 先前测试周期揭露的问题可能迫使对测试优先级进行修改。被修正的缺陷需要额外的再测试,这也是需要进行计划的。当问题不能被完整地重现和分析时,也需要额外的测试;
- ★资源:制定当前测试周期计划时,必须与当前的项目计划保持一致,应该考虑诸如现有人员和假期计划、测试环境和特殊测试工具当前的可用性等问题;

详细测试计划的制订

测试经理需要根据开发状态、测试结果和资源等情况,为下一个测试活动估计工作量和时间需求,并详细定义应该在什么时间、由哪些测试人员、按什么样的顺序来执行哪些测试用例。这个详细策划的结果将会是下一测试周期的测试计划,比如回归测试计划!

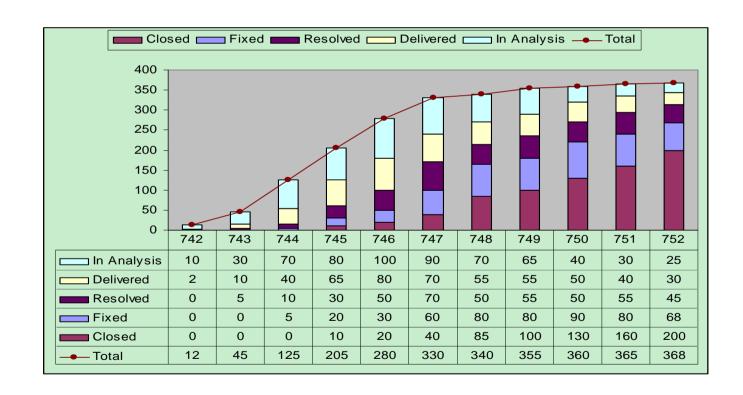
测试管理

测试周期的监测

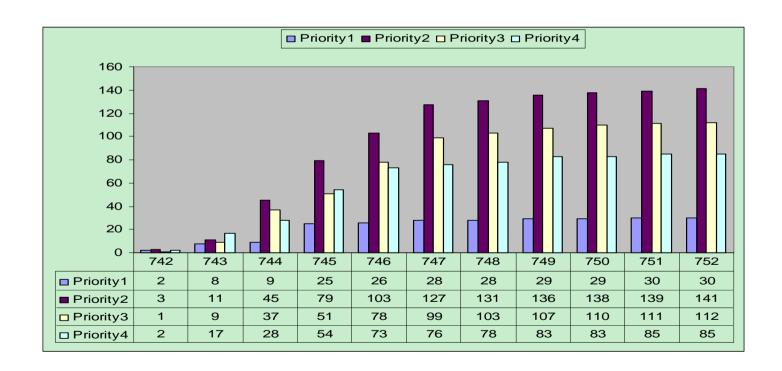
为了测量和监督当前测试的结果,我们需要合适的评判标准或者参考指标:测试度量的使用!

测试度量应该是简单的、规则的、可靠的!

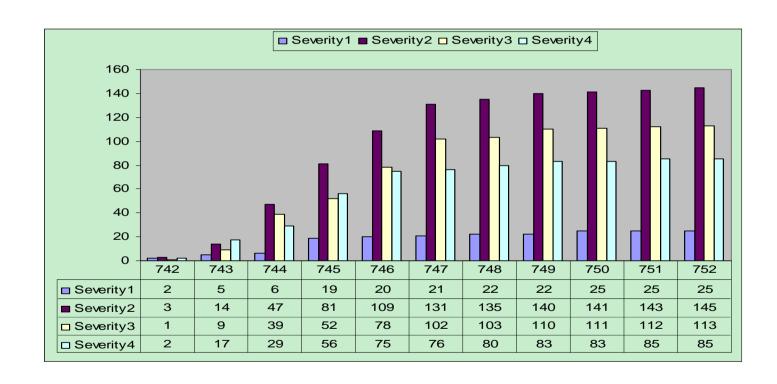
基于缺陷和失效的度量: 状态



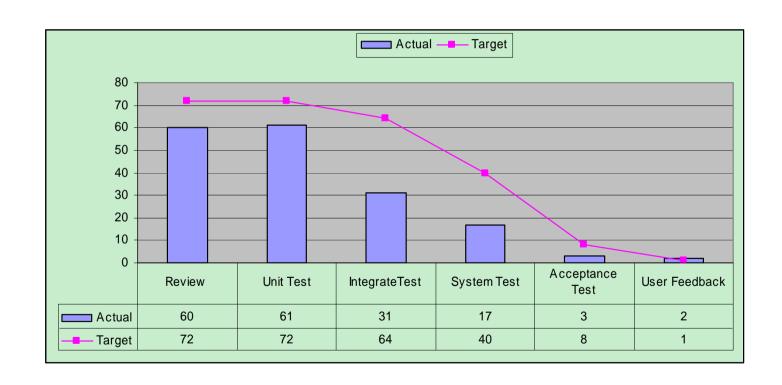
基于缺陷和失效的度量: 优先级



基于缺陷和失效的度量:严重程度



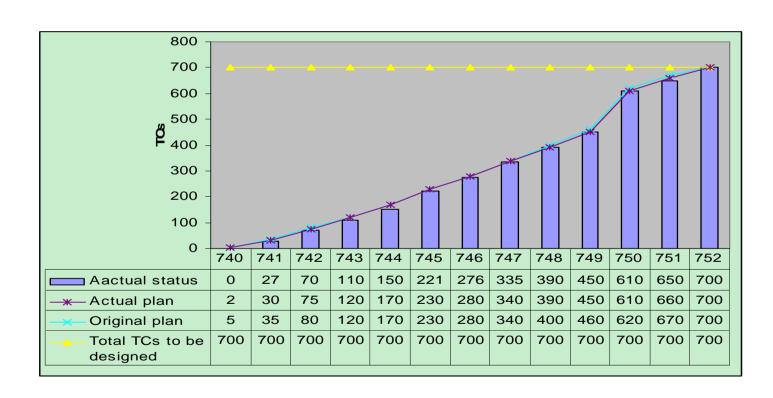
基于缺陷和失效的度量: 按阶段分布



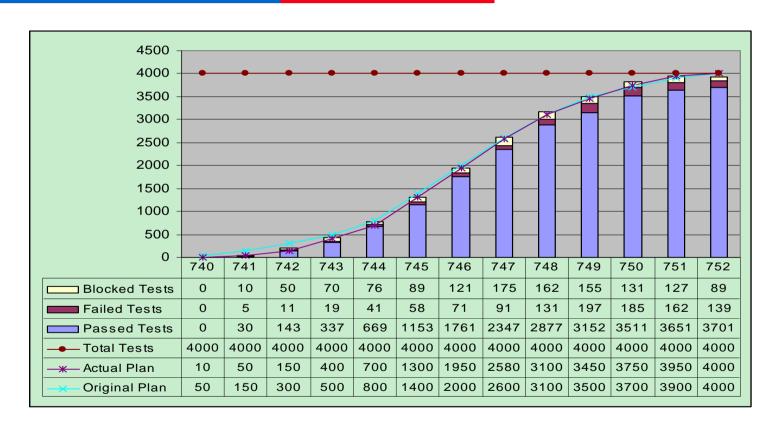
基于缺陷和失效的度量: 按模块分布



基于测试用例的度量:设计



基于测试用例的度量: 执行



基于测试对象的度量

★代码、对话框、可能的不同安装方式或平台等的 覆盖率;

基于成本的度量

★ 已经花费的测试成本,下一测试周期的成本与预期收益(已被预防的失效成本,或已被降低的项目风险和产品风险)的关系;

测试状态报告:每个测试周期结束之后

- ★测试对象、测试级别、测试周期起始终止时间;
- ★测试的计划、测试的运行、测试的阻塞状态;
- ★事件状态:新建、打开、已修正;
- ★风险:新的、变化的、已知的;
- ★展望:下个测试周期的计划;
- ★评估:对测试对象的成熟度、发布可能性或当前 信任度进行的(主观)评估;

测试管理

测试状态报告和出口准则

- ★根据测试的数据和度量,回答"测试的进度如何" 的问题;
- ★根据测试的数据和度量,回答"是不是可以结束测 试"的问题;
- ★根据合适的度量来定义合适的测试出口准则:质量要求、可用的测试资源,比如时间、人员、测试工具等;

出口准则例子

VSR项目中的测试用例被划分为以下3个优先级。

优 先 级	意 义
1	必须执行此测试用例
2	应该执行此测试用例
3	可以执行此测试用例

根据这个优先级划分,确定了如下用于测试VSR系统的基于测试 用例的完成准则。

- □ 所有优先级为1的测试用例成功执行。
- □ 至少60%的优先级为2的测试用例被执行。

测试报告的目的

测试总结报告或者测试报告指的是对软件系统或单元进行测试产生的行为及结果的描述文件;

测试总结报告以文档的形式,描述了被测软件的测试情况和测试结果,并对相关的结果和数据进行分析,向项目管理层提供信息和建议!

测试报告的模板

- ★测试总结报告标识:根据文档命名规则,赋予项目测试总结报告的一个唯一的标识符;
- ★测试总结报告概要:对测试对象进行总结性的评价。标识测试对象,以及它们的版本,测试使用的测试环境等。对于每个测试对象或者测试内容,提供相关的参考文档:测试计划、详细测试用例、测试日志和测试缺陷报告等;
- ★偏差:在测试总结报告中,需要描述测试过程中的各种偏差,包括测试对象和测试设计之间的偏差、测试计划和执行之间的偏差等。并且最好能够描述引起偏差的原因;

测试管理

测试报告的模板 (续)

- ★ 全面评估:根据测试计划中定义的测试准则,全面评估测试过程。识别没有进行足够测试的项目功能特性,并且给出没有完全测试的理由;
- ★ 测试结果总结: 识别所有已经解决的缺陷,并且总结缺陷解决的方法。同时识别所有没有解决的缺陷;
- ★ 评估:对每个测试对象进行整体的评估,包括它们使用的限制和使用软件可能存在的风险。评估应该基于测试对象的入口和出口准则;
- ★ 测试活动总结:对测试活动和主要的测试事件进行总结。 总结资源使用情况,包括人力资源使用情况,测试机器 使用情况以及每个测试活动花费的时间等等;
- ★ 批准: 描述需要对测试总结报告进行批准的人员和相应的职务。批准需要他们的签名和批准的时间:

测试周期的控制

- ★对背离测试计划作出反应:
 - → 请求和部署额外的测试资源;
 - → 调整测试计划本身, 比如低优先级的测试用例将被忽略;
 - → 根据所发现的故障和问题的严重性,测试的持续时间可能被延长;
 - → 根据测试环境的可用性, 更改测试进度计划;
- ★清楚的传达测试计划的变更

课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

ISTQB考试知识点

★总结配置管理如何支持测试(K2);

配置管理的目的

配置管理的目的是在项目和产品的生命周期内建立 和维护软件或系统产品的完整性,比如组件、数据 和文档的完整性!

软件开发中的混乱现象

- 发错了版本
- 安装后不工作
- 异地不能正常工作
- 已经解决的缺陷过后又出现错误
- 开发人员把产品拿出去出售赢利
- 找不到最新修改了的源程序
- 找不到编程序的人

混乱问题的表现

- ★因为可以同时访问共享文件,开发人员对源代码 和其他文档做的修改相互覆盖;
- ★集成活动受阻:
 - ◆ 因为不清楚开发团队中特定组件有哪些代码版本以及哪个是最新版本;
 - ◆ 因为不清楚哪些版本的组件属于一组并可被集成到更 大的子系统中;
 - → 因为使用不同版本的编译器和其他开发工具;

混乱问题的表现(续)

- ★问题分析、故障修正和回归测试变得复杂:
 - ◆ 因为不知道组件的代码与之前版本相比什么地方被修改了,以及为什么修改;
 - ◆ 因为不知道特定的集成子系统(对象代码)源于哪些代码文件;
- ★测试和测试评估受阻,因为以下信息不明确:
 - → 哪些测试用例属于哪个版本的测试对象;
 - → 哪个版本的测试对象的哪个测试周期得出哪个结果;

配置管理需求:测试角度

- ★版本管理: 指对不同版本的配置项例如某个系统的1.0版本和1.1版本,进行目录编制、存档和检索。这也包括对特定修改的原因做的评注;
- ★配置标识:这是对特定版本的所有文件(配置对象)的标识和管理,这些文件构成一个子系统(配置)。配置标识的前提条件是版本管理;

配置管理需求:测试角度(续)

- ★事件状态和变更状态记录:这是对事件报告和变更请求进行文档化,以及将其重新应用于配置对象的可能性;
- ★配置审计:有助于检查配置管理的有效性。这种审计(audit)可用于检查所有软件组件是否已通过配置管理进行文档化,配置是否能够被正确识别等等;

课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

ISTQB考试知识点

- ★ 将可能会威胁一个或多个利益相关者实现项目目标的可能问题描述为风险(**K2**);
- ★知道风险是由发生的可能性和影响力(发生后所造成的危害)来决定的(**K1**);
- ★区别项目风险和产品风险(K2);
- ★了解典型的产品风险和项目风险(K1);
- ★通过例子来描述在测试计划中如何进行风险分析 和风险管理(**K2**);

风险的定义

风险可以定义为事件、危险、威胁或情况等发生的可能性以及由此产生不可预料的后果,即一个潜在的问题!

风险级别由出现不确定事件的可能性和出现后所产生的影响(事件引发的不好的结果)两个方面来决定!

风险识别的活动

- ★风险的类别:
 - ◆ 资源方面:包括人力资源(人力资源的数量、技能水平等)和设备资源等;
 - ◆ 技术方面: 比如并清楚的需求规格说明、新采用的测试技术和开发技术等;
 - ★沟通方面: 比如和客户的沟通、组织内的沟通、测试团队内的沟通、项目开发在不同的地方需要进行的沟通、语言和文化差异等等;
 - → 时间方面: 比如测试设计、执行方面的时间安排等;
 - ◆相互依赖:测试与开发之间的依赖关系、和第三方的依赖关系等;
 - → 工具方面:测试工具的使用和应用;

风险识别的活动(续)

- ★风险识别的方法:
 - → 头脑风暴法:测试经理负责安排和召集相关人员进行 头脑风暴来识别项目测试中可能存在的风险;
 - → 风险分类问卷调查: 风险分类问卷调查来进行测试风险识别,测试经理选择合适的人员来参与问卷调查, 问卷调查得到的结果可以帮助识别和确认测试过程中可能存在的风险;

风险识别的活动 (续)

- ★风险识别的方法:
 - ★ 定期测试风险汇报:持续和不断进行测试风险识别的方法。在测试过程中识别或者碰到的风险,可以通过定期的汇报机制,和测试项目中原来的风险列表一起进行分析、跟踪和应对。从而不断的充实测试风险数据库,作为将来项目测试风险识别的重要输入;
 - ◆ 使用测试风险列表:测试风险列表在测试风险识别中 是一个非常有用的工具,可以帮助识别各种可能的测 试风险;

风险分析的活动: 影响程度

风险影响程度	测试质量	测试成本	测试进度
高(3)	无法达到测试质量要求	使得测试成本高出	使得测试进度延期
		50%	50%
中(2)	部分降低测试质量要求	使得测试成本高出	使得测试进度延期
		10%~50%	10%~50%
低(1)	基本对测试质量要求没	使得测试成本些许	使得测试进度些许
	有影响	增加(小于10	延期(小于10%)
		%)	

风险分析的活动:可能性

测试风险可能性等级	测试风险可能性
高 (3)	大于 67%
中 (2)	在 33%和 67%之间
低 (1)	小于 33%

风险分析的活动: 时间区间

测试项目周期	测试风险期限分类	测试风险期限值
	近期为1个星期	3
小于1个月	中期为2个星期	2
	远期为3个星期	1
小于4个月	近期为3个星期	3
	中期为8个星期	2
	远期为 13 个星期	1
小于1年	近期为8个星期	3
	中期为 26 个星期	2
	远期为 46 个星期	1
大于1年	根据实际情况来定义	N. A

风险应对的活动

- ★<u>测试风险减轻</u>:将测试风险产生的可能性或者测试风险产生的不利影响降低到一个可以接受的阈值;
- ★测试风险避免:测试风险避免是测试风险减轻的一种特殊情况,目的是使风险产生的可能性降低为0。比如通过更新测试计划来消除风险或者消除产生风险的原因(比如评审需求,使得需求满足可测试性条件;为新员工提供相关的项目技术培训和测试流程等);

风险应对的活动(续)

- ★<u>测试风险转移</u>:将测试风险产生的影响转移到第 三方:
- ★<u>测试风险接受</u>:经过分析和考虑,没有必要修改测试计划,或者没有办法采取一个合适的应对措施和活动来避免、转移或者减轻的时候,我们只能接受测试的风险;
- ★<u>测试意外计划</u>: 当风险变成现实的时候,我们需要相应的指定测试意外计划,即风险变成了实际的问题;

风险监控的活动

- ★收集测试风险的相关数据和信息,比如测试风险的状态、测试风险的影响程度、测试风险的可能性、测试风险的应对活动等;
- ★根据测试风险系数,判断测试风险的趋势,比如测试风险的趋势是增加了、减少了还是没有变化。 从而判断是否需要额外的测试风险管理活动;
- ★识别新的测试风险;

项目风险的定义

★关于项目按目标交付的能力方面的风险;

项目风险:公司组织因素

- ★技能和人才的不足;
- ★个人和培训问题;
- ★政策因素:
 - → 与测试员进行需求和测试结果沟通方面存在的问题;
 - → 测试和评审中发现的信息未能得到进一步跟踪(如未 改进开发和测试实践);
- ★对测试的态度或预期不合理;

项目风险: 技术方面

- ★不能定义正确的需求;
- ★给定现有限制的情况下,能够满足需求的程度;
- ★设计、编码和测试的质量;

项目风险: 供应商问题

- ★第三方存在的问题;
- ★合同方面的问题;

产品风险的定义

★ 在软件或系统中的潜在失效的区域(即将来可能 发生的不利事件或危险)称之为产品风险,因为 它们对产品质量而言是一个风险;

产品风险的例子

- ★易错的软件交付使用;
- ★ 软件/硬件对个人或公司造成伤害的可能性;
- ★劣质的软件特征,比如功能性、可靠性等;
- ★软件没有实现既定的功能;

基于风险的测试

- ★基于风险的测试能够在项目初期阶段的开始就提供一个降低产品风险的有效可能性;
- ★基于风险的测试包括对产品风险的识别以及在考虑了这些风险的情况下指导测试计划和测试控制、 以及规格说明、测试准备和执行;

识别的风险可以帮助

- ★决定采用何种测试技术;
- ★决定要进行测试的范围;
- ★确定测试的优先级,以尽早的发现严重的缺陷;
- ★决定是否可以通过一些非测试的活动来减少风险,比如对缺乏经验的设计者进行相应的培训;

课程内容

- 1. 测试的组织架构
- 2. 测试计划和估算
- 3. 测试进度监控
- 4. 配置管理
- 5. 风险和测试
- 6. 事件管理

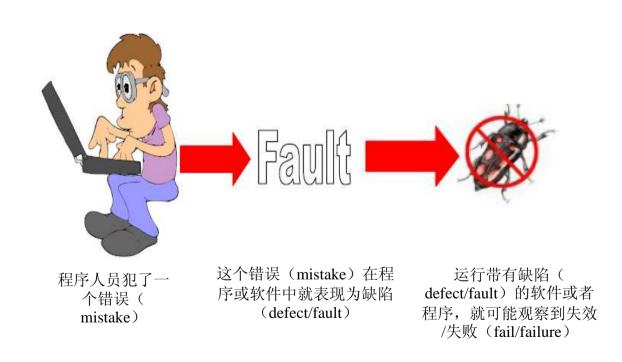
ISTOB考试知识点

- ★按照"软件测试文档标准(IEEE 829)"认识事件报告的内容(K1);
- ★针对测试过程中发现的失效来编写相关的事件报告(**K3**):

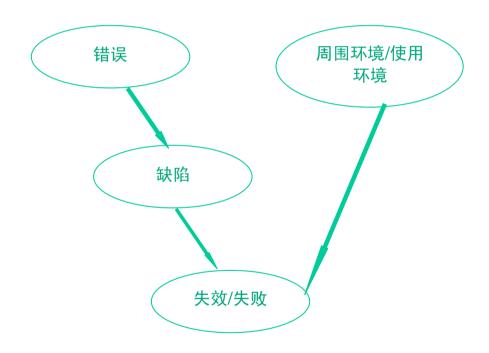
事件管理的定义

- ★事件管理主要指的是缺陷管理;
- ★事件管理的目的是为了保证可靠和快速地消除通过各种测试级别检测到的缺陷和失效;

缺陷基本定义



相互关系



事件报告的模板

	属 性	含 义	
标识	ID/编号 测试对象 版本 平台 报告人 责任开发人员 报告日期	每个报告的唯一标识符/编号 测试对象的标识或名称 测试对象的精确版本标识 软硬件平台标识或出现问题的测试环境的标识 报告问题的测试人员的标识(可能包括测试级别) 负责开发测试对象的人员或团队的名称 发现问题的日期,可能包括具体时间	
分类	状态 严重性 优先级 需求 问题源	报告处理的当前状态 (和完整历史) (见6.6.4 节) 问题的严重性分类 (见6.6.3 节) 修正的优先级分类 (见6.6.3 节) 指向由于出现问题而没有实现的 (客户) 需求 引入缺陷的项目阶段 (分析、设计、编程); 有助于计划过程改进方法	
问题描述	测试用例 问题描述 评注 缺陷修正 参考	测试用例描述 (名称、编号) 或重现问题的步骤 对发生的问题或失效的描述; 预期的和实际观察到 的结果或行为 开发人员和其他相关人员对报告的评论 描述了为修正缺陷而做的更改 其他相关报告参考	

事件报告的分类:严重度

类 别	描 述
1 —— 致命的	系统崩溃,可能丢失数据。这样的测试对象不能发布
2——非常严重	重大故障;不符合需求或实现错误;对许多相关人员造成实
	际影响。测试对象只能在严格的限制条件下使用(解决问题困
	难或代价昂贵)
3 ——严重	功能偏差或受限制 ("正常"故障) ; 错误地或部分地实现需
	求;对一些相关人员造成实际影响。测试对象可以在限制条件
	下被使用
4——一般	微小的偏差;对极少数相关人员造成一定影响。系统可以不
	受限制地被使用
5轻微	对极少数相关人员造成轻微影响。系统可以不受限制地被使
	用。例如,拼写错误或屏幕布局错误

事件报告的分类:优先级

优 先 级	描述
1立即	用户的业务或工作流程受阻,或运行中的测试无法继续。该
	问题需要立即修复,或必要的话采取临时措施(补丁
	(patch)) o
2——下次发布	在下次常规的产品发布或下次 (内部) 测试对象版本交付时
	实施修正。
3 —— 必要时	在受影响的系统部件应当进行修订时进行修正。
4——未决	尚无修正计划。

严重度和优先级的关系

- ★严重度和优先级是含义不同但密切相关的两个概念,从不同的侧面描述了软件缺陷对软件质量、最终用户、开发过程的影响程度和处理方式;
- ★一般来说,严重度高的缺陷具有较高的优先级。 严重度高说明缺陷对软件造成的质量危害性大, 需要优先处理,而严重性低的缺陷可能只是软件 不是尽善尽美,可以稍后处理;
- ★但是优先级和严重度并不总是一一对应的,也存在低优先级、高严重度的缺陷,或者高优先级、 低严重度的软件缺陷;

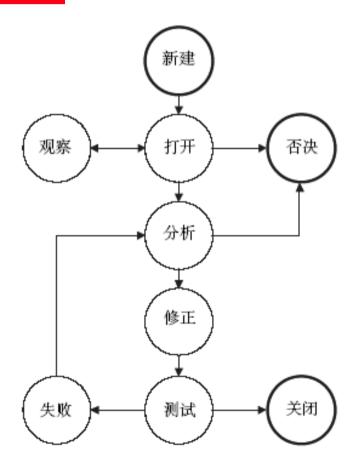
严重度和优先级的关系

- ★ 修改软件缺陷并不是纯技术的问题,有时候我们能需要同时考虑软件版本发布和质量风险等因素:
 - ★ 某个严重的缺陷只在极端的条件下产生,则可以将缺陷的优先级设置的比较低,可能没有必要马上修改;
 - → 如果修正一个软件缺陷,需要重新修改软件的整体架构,可能会产生更多的潜在缺陷,而且市场要求尽快发布软件版本。假如这个缺陷严重度很高,是否需要修改,也需要全盘考虑;
 - → 对于有些缺陷,可能它的严重度很低,比如界面单词拼写错误,但可能这是公司的名称或者LOG,则这个缺陷的优先级很高,必须尽快进行修复。因为这个关系到软件系统和公司在市场上的形象;

事件的状态

状 态 (设定人)	描述
新建	报告人编写了一份新报告、在其中包含了易于理解的描述和
(测试人员)	分类
打开	测试经理定期检查新报告的可理解性和所有必要属性的完整
(测试经理)	重复的或明显无用的报告将被调整或拒绝。该报告将被指派给
	责任开发人员,并将状态设置为"打开"
否决	报告被认为是不正确的并被否决(测试对象中没有故障,变
(测试经理)	更请求不被考虑)
分析	责任开发人员一开始处理报告时,就将状态设定为"分析"。
(开发人员)	分析的结果(根源、可能的修补方法、修正工作量估值等)将
	记录到评注中
观察	所描述的事件既不能重现,又不能简单删除。报告处于未决
(开发人员)	状态,直至得到更多信息或更深了解
修正	根据分析,项目经理决定应该开始修正,因此将状态设定为
(项目经理)	"修正"。责任开发人员进行修正并在评注中记录修正内容
测试	一旦责任开发人员认为问题已修正,报告就被设定为"测试"
(开发人员)	状态。包含此修正的新版本软件将被标识
关闭	设定为"测试"状态的报告将在下一测试周期进行验证。为
(测试人员)	此,至少发现此问题的测试用例将被重复执行。如果测试后确
	定修复已成功,测试人员将最终状态设定为"关闭",以此结
	束报告历史
失败	如果重复执行测试后表明修复尝试不成功或不充分,状态将
(测试人员)	被设定为"失败",并且有必要重新进行分析

事件的状态模型



缺陷相关的出口准则示例

VSR系统测试的测试出口准则不仅要反映测试过程,还要反映达到产品要求的质量。因此,测试经理使用如下度量来增强测试出口准则。

- □ 所有严重度为"1——致命"的故障都已关闭。
- □ 所有优先级为"2——立即"的故障也都已关闭。
- □ 每个测试星期(周)中状态为"新建"的事件报告数保持稳定 或下降。

答疑

