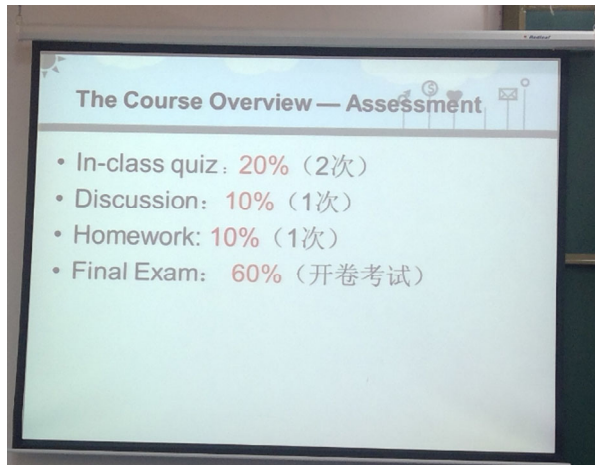


Ch1 软件质量与测试概念

2018年11月6日 10:41



随堂测试：黑盒测试+白盒测试

英文试卷

本章关键

➤ 软件质量的定义

○ 百度百科：

软件质量是“反映实体满足明确的和隐含的需求的能力的特性的总和”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的和隐含特征相一致的程度。

➤ 质量保证和质量控制的概念

Software Quality Assurance

为了确保软件开发过程和结果符合预期的要求，而建立的一系列规程，以及依照规程和计划采取的一系列活动，以及结果评价。

○

Quality Control

为了达到产品的品质要求所采取的作业技术和活动。

➤ 软件缺陷的定义

○ 只要是软件中存在的问题，最终表现为用户所需要的功能没有完全实现，没有满足用户的最终要求

- 从产品内部看，软件缺陷是产品开发或维护过程中所存在的错误、毛病等各种问题。
- 从产品外部看，软件缺陷是系统所需要实现的某种功能的失效或者违背。

➤ 软件测试的定义

○ 软件测试是为了发现错误而执行程序的过程。

➤ 测试的类型

- C1: 按照测试生成的来源
- C2: 按照生命周期的阶段

- C3: 按照测试活动的目的
- C4: 按被测对象的特征
- C5: 按测试过程的模型
- 软件测试在软件开发中的角色
 - 贯穿整个软件开发周期
- 测试规则
 - 尽早测试，经常测试。
 - 集成应用程序开发和测试生命周期。
 - 使测试方法形式化。
 - 制定全面的测试计划。
 - 同时使用静态和动态测试。
 - 定义你期望的结果。
 - 了解应用程序背后的业务原因。
 - 使用多个级别和类型的测试。
 - 评审和检查工作（静态测试的范围），会降低成本。
 - 不要让程序员检查他们自己的工作。

1.1 了解软件质量

- a. 什么是质量
 - i. 用户视角
 - ii. 开发者视角
- b. 质量的评判标准——度量criteria
 - i. validity有效性
 - ii. generality普遍性
 - iii. interoperability互操作性
 - iv.
- c. 软件质量涉及的活动包括
 - i. 过程process, 质量保证
 - ii. 产品product, 质量控制
- d. 质量预防和检测
- e. It's all about the End-user

Software Quality

1.满足用户需求； 2.建立合理的进度、成本与功能的关系； 3.具备扩展性和灵活性； 4.能有效的处理例外情况； 5.保持成本和性能的平衡

Software Quality Assurance

- f. 为了确保软件开发过程和结果符合预期的要求，而建立的一系列规程，以及依照规程和计划采取的一系列活动，以及结果评价。

Quality Control

为了达到产品的品质要求所采取的作业技术和活动。

1.2了解软件测试，是为了发现问题

- a. 软件测试的定义： is the essential step which is planned and systematic，是软件质量保证的关键步骤
- b. 为什么测试
 - i. 基于安全性和经济角度
- c. 什么是软件缺陷defect，是以下四点的统称
 - 从产品内部看，软件缺陷是产品开发或维护过程中所存在的错误、毛病等各种问题。
 - 从产品外部看，软件缺陷是系统所需要实现的某种功能的失效或者违背。
 - i. 定义：只要是软件中存在的问题，最终表现为用户所需要的功能没有完全实现，没有满足用户的最终要求
 - 1) 用户的期望没被满足
 - 2) 软件功能执行时发生错误
 - 3) Defect = bug
 - ii. 缺陷举例：
 - 1) 性能：运行太慢
 - 2) 不一致性：CTRL+S不是保存，是其他的
 - 3) 缺陷的表现形式有很多种，不仅仅表现为功能失效方面，还体现在其他方面。
 - a) 例如，数据结果不精确，精度不够，
 - b) 实际结果和预期结果不一致，
 - c) 用户界面出错、混乱、不美观，
 - d) 运行中断、系统崩溃。

Error（错误）：occurs in the process of writing a program.

Fault（故障）：is the manifestation（表现） of one or more errors.

iii. **Failure（失效）**：occurs when faulty piece of code is executed leading to an incorrect state that propagates（传播） to the program's output.

Incident（事故）：no message is displayed when failure occurs.

iv. 缺陷什么时候发生？

- 软件没有做规范中说它应该做的事情。
- 软件做了一些规范说它不应该做的事情。
- 软件做了规范中没有提到的事情。
- 软件没有做产品说明书中没有提到但应该做的事情。
- 软件难以理解，难以使用，速度慢.....

d. 软件测试的重要功能：

- i. **Verification 验证**，软件开发过程，比对需求规格说明
- ii. **validation 确认**，从用户角度考虑

e. 测试test和调试debug

Test & debug

- Automated test vs. manual operate
- Don't know details are OK vs. must know details
- Correctness proof and how to do with failure vs. correctness proof only
- Checking vs. reasoning
- Plan , under control vs. out of control

测试条件已知，规程可定义，结果可预知

测试可以计划，过程可控；调试不可控

测试是检验，调试是推理过程

测试除了表明程序正确还要验证程序如何处理失败，调试表明正确

测试可不了解设计细节,调试必须了解设计细节

测试可自动化,调试无法自动化

o

o 目的：为了发现软件的错误，降低软件的风险

软件测试是为了发现错误而执行程序的过程。

一个好的测试能过在第一时间发现程序中存在的错误。

一个好的测试是发现了至今尚未发现的错误。

o 测试的种类：

- C1: 按照测试生成的来源
- C2: 按照生命周期的阶段
- C3: 按照测试活动的目的

- C4: 按被测对象的特征
- C5: 按测试过程的模型

C1: 按照测试生成的来源

- 黑盒测试/功能性测试
- 白盒测试/结构性测试

Artifact	Technique
Requirements (informal)	Black-box testing
Code	White-box testing
Requirements and code	Black-box testing and white-box testing
Formal model: graphical or mathematical specification	Model-based specification
Component's interface	Interface testing

C2: 按照生命周期的阶段

- 单元测试
- 集成测试
- 系统测试

Phase	Technique
Coding	Unit testing
Integration	Integration testing
System integration	System testing
Maintenance	Regression testing
Post system, pre-release	Beta-testing

C3: 按照测试活动的目的

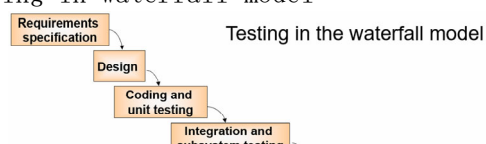
Goal	Technique
Advertised features	Functional
Security	Security
Invalid inputs	Robustness
Vulnerabilities	Vulnerability
Errors in GUI	GUI
Operational correctness	Operational
Reliability assessment	Reliability
Resistance to penetration	Penetration
System performance	Performance
Customer acceptability	Acceptance
Business compatibility	Compatibility
Peripherals compatibility	Configuration
Foreign language compatibility	Foreign language

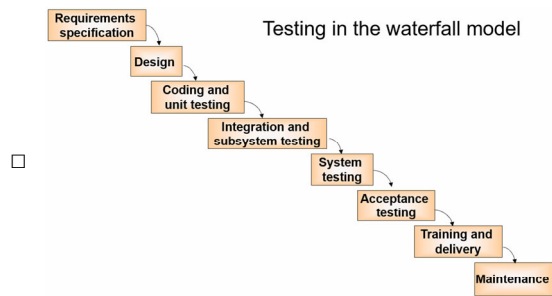
C4: 按被测对象的特征

Characteristic	Technique
Application component	Component testing
Batch processing	Equivalence partitioning, finite-state model-based testing, ...
Client and server	Client-server testing
Compiler	Compiler testing
Design	Design testing
Code	Code testing

C5: 按测试过程的模型

- Testing in waterfall model





□ 缺点：不能对上一步的模型进行调整

■ Testing in V-model

□ 在开发周期的每个阶段显式地指定测试活动

■ Spiral testing

□ 应用于软件增量时，每个增量都可能是最终将应用程序交付给客户的原型。

■ Agile testing

□ 要求团队与客户的良好合作

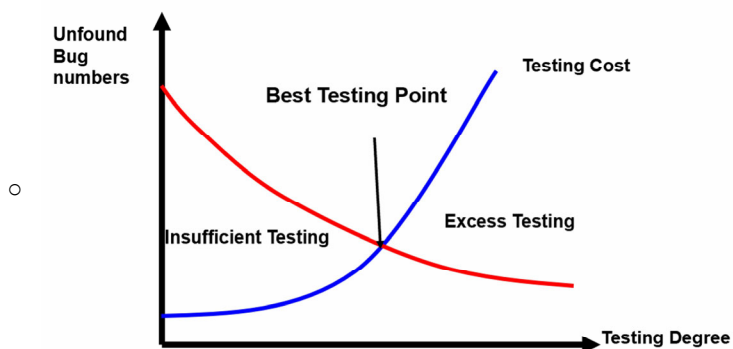
□ 用于极限编程

■ Test-driven development (TDD)

□ Requirements specified as tests

○ IBM研究结果表明：缺陷存在逐级放大的趋势

• 了解软件测试的基本原则



○ 20%的模块有时候有80%的缺陷

○ 尽早测试，经常测试。

集成应用程序开发和测试生命周期。

使测试方法形式化。

制定全面的测试计划。

同时使用静态和动态测试。

○ 定义你期望的结果。

了解应用程序背后的业务原因。

使用多个级别和类型的测试。

评审和检查工作（静态测试的范围），会降低成本。

不要让程序员检查他们自己的工作。

○ 测试通过的规则：

■ 是否执行了所有测试用例。

- 功能设计是否完成。
- 是否有足够的bug。
- 什么是测试用例：
 - 如果系统按照其规范操作，则用于测试系统的输入和这些输入的预测输出。
- 测试用例的设计原则：
 - 避免模糊vague的测试用例。
 - 相似函数应进行抽象和分类classified。
 - 避免复杂的测试用例。
 - 给定测试用例，必须要有预期结果
- 名词解释：
 1. 软件缺陷：（符合下列规则的叫软件缺陷）：
 - 1). 软件未达到产品说明书的功能
 - 2). 软件出现了产品说明书指明不会出现的错误
 - 3). 软件功能超出产品说明书指明范围
 - 4). 软件未达到产品说明书虽未指出但应达到的目标
 - 5). 软件测试员认为难以理解、不易使用、运行速度缓慢、或者最终用户认为不好
 2. 单元测试：单元测试是对软件设计的最小单元——模块进行正确性检验的测试工作，主要测试模块在语法、格式和逻辑上的错误。
 3. 白盒测试也称为结构测试或逻辑驱动测试，前提是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的机构测试程序，检测程序中的每条通路是否都能够按预定要求正确工作，而不管产品的功能，主要用于软件验证。
 4. 回归测试：指软件系统被修改或扩充（如系统功能增强或升级）后重新进行的测试，是为了保证对软件所做的修改没有引入新的错误而重复进行的测试。
 5. 等价类：指某个输入域的子集合，在该子集合中，各个输入数据对于揭露程序中的错误都是等效的。