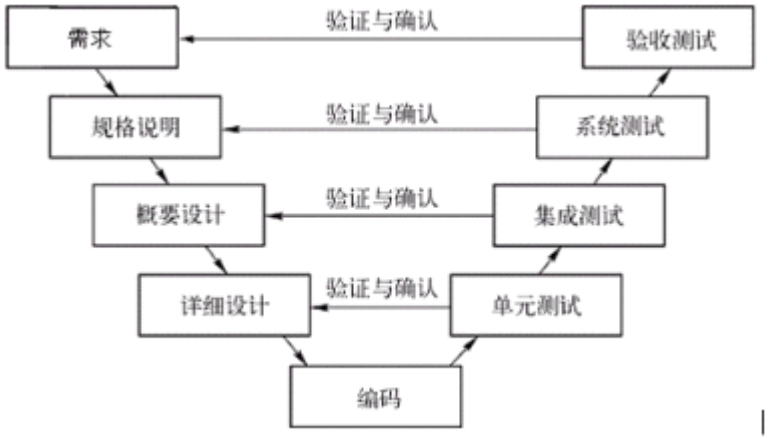
测试开发

2020年9月24日 14:40

V模型



测试过程-开发过程的后半部分

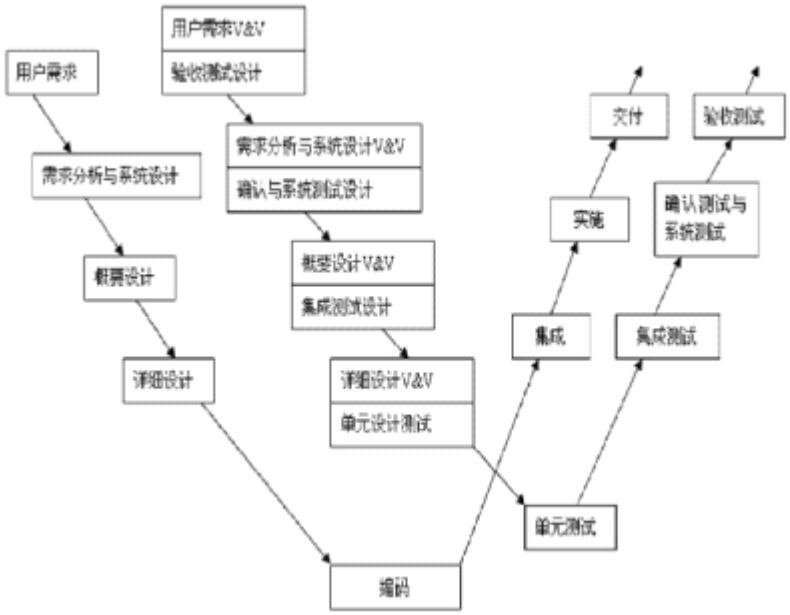
单元测试-代码开发-详细设计要求

集成测试-此前测试过的各组成部分是否能完好地结合到一起

系统测试-已集成在一起的产品-系统规格说明书要求

验收测试-检测产品是否符合最终用户的需求

W模型



测试与开发是同步进行的

1. 单元测试
   完成最小的软件设计单元（模块）的验证工作
   目标：确保模块被正确的编码
   白盒

对代码风格和规则、程序设计和结构、业务逻辑等进行静态测试，及早的发现和解决不易显现的错误

2. 集成测试

   通过测试发现与模块接口有关的问题

   目标：把通过了单元测试的模块拿来，构造一个在设计中所描述的程序结构

   避免一次性的集成（除非软件规模很小），而采用增量集成

   自顶向下集成：模块集成的顺序是首先集成主模块，然后按照控制层次结构向下进行集成，隶属于主模块的模块按照深度优先或广度优先的方式集成到整个结构中去

   自底向上集成：从原子模块开始来进行构造和测试，因为模块是自底向上集成的，进行时要求所有隶属于某个给顶层次的模块总是存在的，也不再有使用稳定测试桩的必要

3. 系统测试

   是基于系统整体需求说明书的黑盒类测试，应覆盖系统所有联合的部件

   针对整个产品系统

   目的：验证系统是否满足了需求规格的定义，找出与需求规格不相符合或与之矛盾的地方

   系统测试的对象：需要测试的产品系统的软件，还要包含软件所依赖的硬件、外设甚至包括某些数据、某些支持软件及其接口等

   将系统中的软件与各种依赖的资源结合起来，在系统实际运行环境下来进行测试

4. 回归测试

   在发生修改之后重新测试先前的测试用例以保证修改的正确性

   理论上，软件产生新版本，都需要进行回归测试，验证以前发现和修复的错误是否在新软件版本上再次出现。根据修复好了的缺陷再重新进行测试

   目的：验证以前出现过但已经修复好的缺陷不再重新出现

   一般指对某已知修正的缺陷再次围绕它原来出现时的步骤重新测试

5. 验收测试

   指系统开发生命周期方法论的一个阶段，这时相关的用户或独立测试人员根据测试计划和结果对系统进行测试和接收。它让系统用户决定是否接收系统

   它是一项确定产品是否能够满足合同或用户所规定需求的测试

   验收测试包括Alpha测试和Beta测试

   Alpha测试：是由用户在开发者的场所来进行的，在一个受控的环境中进行。

   Beta测试：由软件的最终用户在一个或多个用户场所来进行的，开发者通常不在现场，用户记录测试中遇到的问题并报告给开发者，开发者对系统进行最后的修改，并开始准备发布最终的软件

黑盒测试

功能测试或数据驱动测试

它是在已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用

在测试时，把程序看作一个不能打开的黑盆子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。

程序外部结构、不考虑内部逻辑结构、针对软件界面和软件功能进行测试。

穷举输入测试，只有把所有可能的输入都作为测试情况使用，才能以这种方法查出程序中所有的错误

测试所有合法的输入，不合法但是可能的输入进行测试

常用的黑盒测试方法：

等价类划分法；边界值分析法；因果图法；场景法；正交实验设计法；判定表驱动分析法；错误推测法；功能图分析法。

1. 等价类划分
   等价类划分是将系统的输入域划分为若干部分，然后从每个部分选取少量代表性数据进行测试。等价类可以划分为有效等价类和无效等价类，设计测试用例的时候要考虑这两种等价类。

2. 边界值分析法
   边界值分析法是对等价类划分的一种补充，因为大多数错误都在输入输出的边界上。边界值分析就是假定大多数错误出现在输入条件的边界上，如果边界附件取值不会导致程序出错，那么其他取值出错的可能性也就很小。
   边界值分析法是通过优先选择不同等价类间的边界值覆盖有效等价类和无效等价类来更有效的进行测试，因此该方法要和等价类划分法结合使用。

3. 正交试验法
   正交是从大量的试验点中挑选出适量的、有代表性的点。正交试验设计是研究多因素多水平的一种设计方法，他是一种基于正交表的高效率、快速、经济的试验设计方法。

4. 状态迁移法
   状态迁移法是对一个状态在给定的条件内能够产生需要的状态变化，有没有出现不可达的状态和非法的状态，状态迁移法是设计足够的用例达到对系统状态的覆盖、状态、条件组合、状态迁移路径的覆盖。

5. 流程分析法

流程分析法主要针对测试场景类型属于流程测试场景的测试项下的测试子项进行设计，这是从白盒测试中路径覆盖分析法借鉴过来的一种很重要的方法。

6. 输入域测试法
   输入域测试法是针对输入会有各种各样的输入值的一个测试，他主要考虑 极端测试、中间范围测试，特殊值测试 。

7. 输出域分析法
   输出域分析法是对输出域进行等价类和边界值分析，确定是要覆盖的输出域样点，反推得到应该输入的输入值，从而构造出测试用例，他的目的是为了达到输出域的等价类和边界值覆盖。

8. 判定表分析法
   判定表是分析和表达多种输入条件下系统执行不同动作的工具，他可以把复杂的逻辑关系和多种条件组合的情况表达的即具体又明确；

9. 因果图法
   因果图是用于描述系统输入输出之间的因果关系、约束关系。因果图的绘制过程是对被测系统的外部特征的建模过程，根据输入输出间的因果图可以得到判定表，从而规划出测试用例。

10. 错误猜测法
    错误猜测法主要是针对系统对于错误操作时对于操作的处理法的猜测法，从而设计测试用例

11. 异常分析法
    异常分析法是针对系统有可能存在的异常操作，软硬件缺陷引起的故障进行分析，分析发生错误时系统对于错误的处理能力和恢复能力依此设计测试用例。

白盒测试

结构测试或逻辑驱动测试，是针对被测单元内部是如何进行工作的测试

它根据程序的控制结构设计测试用例，主要用于软件或程序验证

白盒测试法检查程序内部逻辑结构，对所有的逻辑路径进行测试，是一种穷举路径的测试方法，但即使每条路径都测试过了，但仍然有可能存在错误

穷举路径测试无法检查出程序本身是否违反了设计规范，即程序是否是一个错误的程序；穷举路径测试不可能检查出程序因为遗漏路径而出错；穷举路径测试发现不了一些与数据相关的错误

白盒测试需要遵循的原则：

1. 保证一个模块中的所有独立路径至少被测试一次；

2. 所有逻辑值均需要测试真（true）和假（false）；两种情况；

3. 检查程序的内部数据结构，保证其结构的有效性；

4. 在上下边界及可操作范围内运行所有循环。

常用白盒测试方法：

静态测试：不用运行程序的测试，包括代码检查、静态结构分析、代码质量度量、文档测试等等，它可以由人工进行，充分发挥人的逻辑思维优势，也可以借助软件工具（Fxcop）自动进行。

动态测试：需要执行代码，通过运行程序找到问题，包括功能确认与接口测试、覆盖率分析、性能分析、内存分析等。

白盒测试中的逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。六种覆盖标准发现错误的能力呈由弱到强的变化：

1.语句覆盖每条语句至少执行一次。

2.判定覆盖每个判定的每个分支至少执行一次。

3.条件覆盖每个判定的每个条件应取到各种可能的值。

4.判定/条件覆盖同时满足判定覆盖条件覆盖。

5.条件组合覆盖每个判定中各条件的每一种组合至少出现一次。

6.路径覆盖使程序中每一条可能的路径至少执行一次。

软件质量六个特征

1. 功能特征：与一组功能及其指定性质有关的一组属性，这里的功能是满足明确或隐含的需求的那些功能。
2. 可靠特征：在规定的一段时间和条件下，与软件维持其性能水平的能力有关的一组属性。
3. 易用特征：由一组规定或潜在的用户为使用软件所需作的努力和所作的评价有关的一组属性。
4. 效率特征：与在规定条件下软件的性能水平与所使用资源量之间关系有关的一组属性。
5. 可维护特征：与进行指定的修改所需的努力有关的一组属性。
6. 可移植特征：与软件从一个环境转移到另一个环境的能力有关的一组属性。

来自 <https://www.nowcoder.com/tutorial/97/e1cb34a000db4fc78554fc6f528687bc>