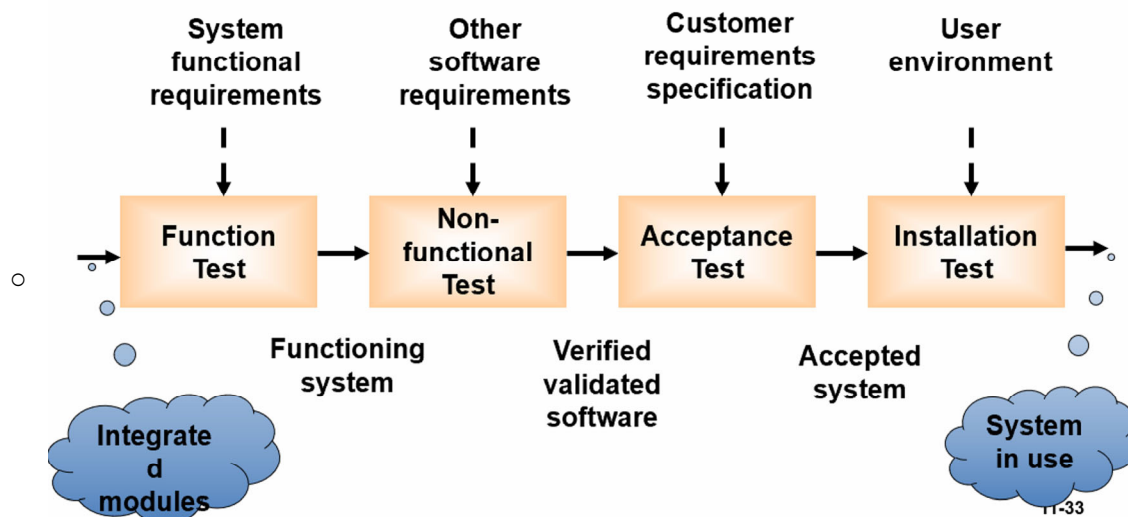


Ch8 系统测试

2018年12月21日 12:01

1. 系统测试概念

- **系统测试**(系统测试)的软件或硬件测试进行一个完整的、集成的系统来评估系统的符合指定的要求。
- 系统测试**属于黑盒测试**的范围，因此，不需要知道代码或逻辑的内部设计。
- 系统测试是一种比较有限的测试类型;它试图检测“内部装配”和整个系统中的缺陷。
- 系统测试是在功能需求规范(FRS)和/或系统需求规范(SRS)的上下文中对整个系统执行的。
- 系统测试是在单元和集成测试之后，在用户的使用环境下测试
- 为什么进行系统测试：
 - 有些属性只能在系统级验证：
 - 可安装性、可用性、兼容性、可维护性
 - 我们可能会涉及到这个级别的用户
 - 用例可能不映射到任何特定的集成单元
 - 考虑系统环境
- 系统测试过程：



2. 系统测试方法

安全性测试：安全测试检查系统对非法侵入的防范能力。安全测试期间，测试人员假扮非法入侵者，采用各种办法试图突破防线。

可扩展性测试：验证代码设计是否良好允许更多的功能在必要时可以被插入到适当的位置中

理智测试：软件主要功能成分的简单测试以保证它是否能进行基本的测试。

冒烟测试：对软件的主要功能进行快餐式测试，最早来自于硬件测试实践，以确定新的硬件在第一次使用时不会着火。用于确认代码中的更改会按预期运行，且不会破坏整个版本的稳定性。

探索测试：一边测试一边探索，是一个交互式的过程，在某种意义上讲是一个自由形式的测试过程

随机测试：测试人员通过随机的尝试系统的功能，试图是系统中断。

回归测试：发生修改后重新测试先前的测试以保证修改的正确性

可靠性测试：对系统在规定条件下在指定时间内执行期需要功能的能力进行的测试

可安装性测试：安装程序是否正确的测试

可维护性测试：测试系统是否满足可维护性指标，缺陷纠正、性能改进、特性增加引起的

可访问性测试：可用性（Usability）、可访问性（Accessibility）和可维护性（Maintainability）。探求人们怎样才能轻松利用网站以及反馈那些可以改进设计和实施的信息。

1.GUI software testing

2.Usability testing

3.Performance testing

4.Compatibility testing

5.Load testing

6.Volume testing

7.Stress testing

8.Security testing

9.Scalability testing

10.Sanity testing

• 11.Smoke testing

• 12.Exploratory testing

• 13.Ad hoc testing

• 14.Regression testing

• 15.Reliability testing

• 16.Recovery testing

• 17.Installation testing

• 18.Maintenance testing

• 19.Accessibility testing

- 图形用户界面测试：

- GUI测试是对使用图形用户界面的产品进行测试的过程，以确保其符合书面规范。
- GUI测试包括两部分：一是验证validation anastomosis(吻合)界面设计和接口的实现,另一种是确认接口处理的正确性。
- 难点：
 - 可能的接口空间非常庞大
 - 事件驱动特性
 - GUI测试的覆盖率不同于传统的结构化覆盖率，没有合适的自动化工具
 - 糟糕的设计使界面和功能混杂在一起，界面的修改会导致更多错误，同时增加了测试的难度和工作量
 - 界面的美学具有很大的主观性
- 为了更好的进行GUI测试，提倡界面与功能的设计分离,把GUI系统分为3层，界面层、界面与功能的接口层、功能层
- 设计测试用例时分三步
 - 1、划分界面元素并根据界面的复杂性进行分层
 - 2、在不同的界面层次确定不同的测试策略
 - 3、进行测试数据分析，提取测试用例

- 可用性测试：

- 是一种通过对用户进行测试来评估产品的技术。
- 专注于测量一个人造产品满足其预期目的的能力。
- 可操作性测试（operate testing）有很大的相似性，都是为了检验用户在理解和使用系统方面的能力
- 性能测试：
 - 性能测试用于确定计算机、网络、软件程序或设备的速度或有效性。
 - 可以在实验室做定量测试
 - 还可以评估可靠性、可伸缩性和互操作性等定性属性。
 - 性能测试通常与压力测试一起进行。
 - 性能测试的目标是消除瓶颈，提高系统的可靠性。
- 压力测试：
 - 指在高负载下更加强调健壮性、可用性和错误处理的测试，而不是在正常情况下认为正确的行为。
 - 是为使系统受压力而设计的。
 - 压力测试的目的是提高可恢复性。
 - 这种测试方式在执行系统下需要异常的数量、频率或资源。
 - 压力测试**属于边界测试**。
 - 其中一个被称为灵敏度测试技术。
 - 研究系统在短时间内响应高峰活动
- Volume容量测试
 - 批量测试是指使用一定数量的数据对软件应用程序进行测试。
 - 是**面向数据的**，其目的是**显示系统可以处理目标内确定的数据容量**。
 - 这个数量可以是数据库大小，也可以是接口文件的大小
 - 方法：
 - 1、分析系统的外部数据源并分类，
 - 2、对每类数据源分析可能的容量限制，
 - 3、对每类数据源构造大容量数据进行系统测试，
 - 4、分析测试结果，并与期望值比较，确定目前系统的容量瓶颈，
 - 5、对系统进行优化，重复上述步骤，直到系统达到期望的容量处理能力
- Load负载测试
 - 对系统或设备提出要求并测量其响应的过程。
 - 最适合多用户系统，通常是使用客户机/服务器模型构建的系统，例如web服务器。
 - 当施加在系统上的负载超出正常使用模式时，为了测试系统在异常高或峰值负载下的响应，称为应力测试
- 兼容性Compatibility测试
 - 兼容性测试是软件非功能测试的一部分，用于评估应用程序与计算环境的兼容性。
- 恢复性Recovery测试
 - 恢复测试是测试应用程序从崩溃、硬件故障和其他类似问题中恢复的能力的活动。
 - 在某种情况下，系统必须是容错的，运行过程中的错误不能使整个系统的功能都停止。
 - 在其他情况下，一个系统中的错误必须在一定时间段内改正，否则就会产生严重的

经济损失。

- 恢复测试采用各种人工干预的手段使软件出错，验证系统的恢复能力。
- 如果系统本身能够自动回复，检验恢复和重新启动是否正确。
- 如果这一恢复需要人工干预，则应该考虑平均修复时间是否在限定范围以内。

3. 系统测试过程

- 系统测试评价整个应用的功能和性能
- 类似于单元测试和集成测试，系统测试也需要遵循一定的过程。
- Deming 循环管理原则是日本制造工业改变的基础。包括政策管理、人力资源管理和安全管理。同样还包括组件管理、物料管理、设备管理和日程表管理等。
- 人们定义目标，并且确定为得到这些目标而需要的条件和方法。
- 让工作按计划进行
- 通过检查来确定工作是否按计划进行
- 如果没有按计划进行，或者不符合预期，选择适合的行动来进行更正

Plan:

- People define goals, and sure to get these goals and need conditions and methods.

Do:

- Do work according to plan

Check:

- Through the examination to determine whether it is scheduled

Act:

- If no scheduled, or does not conform to the expected, choose suitable action for correction

