**软件测试总结**

1.软件生命周期(SDLC)的六个阶段

1、问题的定义及规划

此阶段是软件开发方与需求方共同讨论，主要确定软件的开发目标及其可行性。

2、需求分析

在确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析。需求分析阶段是一个很重要的阶段，这一阶段做得好，将为整个软件开发项目的成功打下良好的基础。"唯一不变的是变化本身。"，同样需求也是在整个软件开发过程中不断变化和深入的，因此我们必须制定需求变更计划来应付这种变化，以保护整个项目的顺利进行。

3、软件设计

此阶段主要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计，数据库设计等等。软件设计一般分为总体设计和详细设计。好的软件设计将为软件程序编写打下良好的基础。

4、程序编码

此阶段是将软件设计的结果转换成计算机可运行的程序代码。在程序编码中必须要制定统一，符合标准的编写规范。以保证程序的可读性，易维护性，提高程序的运行效率。

5、软件测试

在软件设计完成后要经过严密的测试，以发现软件在整个设计过程中存在的问题并加以纠正。整个测试过程分单元测试、组装测试以及系统测试三个阶段进行。测试的方法主要有白盒测试和黑盒测试两种。在测试过程中需要建立详细的测试计划并严格按照测试计划进行测试，以减少测试的随意性。

6、运行维护

软件维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投入使用后，由于多方面的原因，软件不能继续适应用户的要求。要延续软件的使用寿命，就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。

2.软件生命周期模型

从概念提出的那一刻开始，软件产品就进入了软件生命周期。在经历需求、分析、设计、实现、部署后，软件将被使用并进入维护阶段，直到最后由于缺少维护费用而逐渐消亡。这样的一个过程，称为"生命周期模型"（Life Cycle Model）。

典型的几种生命周期模型包括瀑布模型、快速原型模型、迭代模型。

瀑布模型的特点（文档是主体），很多的问题在最后才会暴露出来。迭代模型比瀑布模型问题暴露的要早；快速原型法比瀑布模型直观。

3.软件测试概念

广义概念：指软件生存周期中所有的检查、评审和确认工作，其中包括了对分析、设计阶段，以及完成开发后维护阶段的各类文档、代码的审查和确认

狭义概念：识别软件缺陷的过程，即实际结果与预期结果的不一致

4.软件测试目的

ü测试的目的就是发现软件中的各种缺陷

ü测试只能证明软件存在缺陷，不能证明软件不存在缺陷

ü测试可以使软件中缺陷降低到一定程度，而不是彻底消灭

ü以较少的用例、时间和人力找出软件中的各种错误和缺陷，以确保软件的质量

5．软件测试原则

üGood-enough: 一种权衡投入/产出比的原则

ü保证测试的覆盖程度，但穷举测试是不可能的

ü所有的测试都应追溯到用户需求

ü越早测试越好，测试过程与开发过程应是相结合的

ü测试的规模由小而大，从单元测试到系统测试

ü为了尽可能地发现错误，应该由独立的第三方来测试

ü不能为了便于测试擅自修改程序

ü既应该测试软件该做什么也应该测试软件不该做什么

6．软件测试的的重点

ü测试用例的设计

–测试用例的设计是整个软件测试工作的核心

–测试用例反映对被测对象的质量要求，决定对测试对象的质量评估

ü测试工作的管理

–尤其是对包含多个子系统的大型软件系统，其测试工作涉及大量人力和物力，有效的测试工作管理是保证有效测试工作的必要前提

ü测试环境的建立

–测试环境应该与实际测试环境一致

7．黑盒测试

ü什么是黑盒测试

–又称功能测试或数据驱动测试，是针对软件的功能需求/实现进行测试，通过测试来检测每个功能是否符合需求，不考虑程序内部的逻辑结构

ü黑盒测试方法

–功能划分

–等价类划分

–边界值分析

–因果图

–错误推测等

8．什么是白盒测试

–白盒测试也称结构测试或逻辑驱动测试，必须知道软件内部工作过程，通过测试来检测软件内部是否按照需求、设计正常运行

–白盒测试的主要方法

–对应于程序的一些主要结构：语句、分支、逻辑路径、变量；白盒测试的主要方法是：

–语句覆盖方法

–分支覆盖方法

–逻辑覆盖方法

9.什么是动态测试

动态测试需要在开发/测试环境或实际运行环境中运行软件，并使用测试用例去查找软件缺陷；动态测试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

10.什么是静态测试

静态测试不实际运行软件，主要是对软件的编程格式、结构等方面进行评估.静态测试包括代码检查、程序结构分析、代码质量度量等。它可以由人工进行，也可以借助软件工具自动进行

11.手工测试和自动测试

a.手工测试缺点在于测试工作量大，重复多，回归测试难以实现

b.自动测试利用软件测试工具自动实现全部或部分测试工作：管理、设计、执行和报告；节省大量的测试开销，并能够完成一些手工测试无法实现的测试

ü手工完成测试的全部过程无法保证测试的科学性与严密性:

–修改的缺陷越多，回归测试越困难

–没有人能向决策层提供精确的数据以度量当前的工作进度及工作效率

–反复测试带来的倦怠情绪及其他人为因素使得测试标准前后不一

–测试花费的时间越长，测试的严格性也就越低

ü自动测试将测试人员从反复、烦杂的测试执行中解放出来，用更多的时间进行测试设计和结果分析

ü软件测试不可能完全自动化

ü不能完成所有手工测试任务

ü无创造性且灵活性差，不能改进测试的有效性

ü过程中可能会遇到许多意想不到的问题，特别是当软件不稳定时

ü测试脚本的维护高

12. 测试流程

ü单元测试

ü集成测试

ü系统测试

ü用户验收测试

ü回归测试

13.单元测试

ü完成对最小的软件设计单元—模块的验证工作

ü目标是确保模块被正确地编码

ü使用过程设计描述作为指南，对重要的控制路径进行测试以发现模块内的错误

ü通常情况下是面向白盒的

ü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，及早地发现和解决不易显现的错误

ü单元测试的内容

–接口测试

–内部数据结构

–全局数据结构

–边界

–语句覆盖，错误路径

14.集成测试

ü通过测试发现与模块接口有关的问题

ü目标是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

ü应当避免一次性的集成（除非软件规模很小），而采用增量集成

集成测试主要内容

üAPI

üAPI/参数组合

15．系统测试

ü根据软件需求规范的要求进行系统测试，确认系统满足需求的要求

ü系统测试人员相当于用户代言人

ü在需求分析阶段要确定软件的可测性，保证有效完成系统测试工作

ü系统测试主要内容

ü所有功能需求得到满足

ü所有性能需求得到满足

ü其他需求（例如安全性、容错性、兼容性等）得到满足

16.用户验收/确认测试

üAlpha测试

–是由用户在开发者的场所来进行的，Alpha测试是在一个受控的环境中进行的

üBeta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

17．压力测试VS性能测试

性能测试的目的不是去找bugs,而是排除系统的瓶颈，以及为以后的回归测试建立一个基准。而性能测试的操作，实际上就是一个非常小心受控的测量分析过程。在理想的情况下，被测软件在这个时候已经是足够稳定了

性能测试是为了检查系统的反映，运行速度等性能指标，他的前提是要求在一定负载下，如检查一个网站在100人同时在线的情况下的性能指标，每个用户是否都还可以正常的完成操作等。

概括就是：在不同负载下（负载一定）时，通过一些系统参数（如反应时间等）检查系统的运行情况；

压力测试是为了发现系统能支持的最大负载，他的前提是要求系统性能处在可以接受的范围内，比如经常规定的叶面3秒钟内响应；概括就是：在性能可以接受的前提下，测试系统可以支持的最大负载。

举例说明：针对一个网站进行测试，模拟10到50个用户就是在进行常规性能测试，用户增加到1000乃至上万就变成了压力/负载测试。如果同时对系统进行大量的数据查询操作，就包含了强度测试。

18.主流测试工具的测试流程

========winrunner

1　启动时选择要加载的插件

2　进行一些设置（如录制模式等）

3　识别应用程序的GUI，即创建map(就是学习被测试软件的界面）

4　建立测试脚本（录制及编写）

5　对脚本除错及调试（保证能够运行完）

6　插入各种检查点（图片，文字，控件等）

7　在新版应用程序中执行测试脚本

8　分析结果，回报缺陷

=========quicktestpro========

1　准备录制

打开你要对其进行测试的应用程序，并检查QuickTest中的各项设置是否适合当前的要求。

2　进行录制

打开QuickTest的录制功能，按测试用例中的描述，操作被测试应用程序。

3　编辑测试脚本

通过加入检测点、参数化测试，以及添加分支、循环等控制语句，来增强测试脚本的功能，使将来的回归测试真正能够自动化。

4　调试脚本

调试脚本，检查脚本是否存在错误。

5　在回归测试中运行测试

在对应用程序的回归测试中，通过QuickTest回放对应用程序的操作，检验软件正确性，实现测试的自动化进行。

6　分析结果，报告问题

查看QuickTest记录的运行结果，记录问题，报告测试结果。

＝＝＝＝TestDirect============

安装好后，先进入站点管理

1　创建域及工程

2　添加用户

3　编辑licenses及本服务器

4　编辑数据库

－－TD

1　选择新建的工程进行定制(列表，用户，组，版本等)

2　在require中增加需求

3　把需求转化为plan

4 在testlab中由计划新建测试具体用例与执行

5　发现bug，在defect中提交bug

（每一部分都可以相对独立地使用）

**编写软件测试计划需要考虑的几个问题**

软件测试是有计划、有组织和有系统的软件质量保证活动，而不是随意地、松散地、杂乱地实施过程。为了规范软件测试内容、方法和过程，在对软件进行测试之前，必须创建测试计划。

　　《ANSI/IEEE软件测试文档标准829-1983》将测试计划定义为：“一个叙述了预定的测试活动的范围、途径、资源及进度安排的文档。它确认了测试项、被测特征、测试任务、人员安排，以及任何偶发事件的风险。”

　　软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

　　做好软件的测试计划不是一件容易的事情，需要综合考虑各种影响测试的因素。为了做好软件测试计划，需要注意以下几个方面。

　　1. 明确测试的目标，增强测试计划的实用性当今任何商业软件都包含了丰富的功能，因此，软件测试的内容千头万绪，如何在纷乱的测试内容之间提炼测试的目标，是制定软件测试计划时首先需要明确的问题。测试目标必须是明确的，可以量化和度量的，而不是模棱两可的宏观描述。另外，测试目标应该相对集中，避免罗列出一系列目标，从而轻重不分或平均用力。根据对用户需求文档和设计规格文档的分析，确定被测软件的质量要求和测试需要达到的目标。

　　编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确。

　　2. 坚持“5W”规则，明确内容与过程“5W”规则指的是“What（做什么）”、“Why（为什么做）”、“When（何时做）”、“Where（在哪里）”、“How（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的（Why），明确测试的范围和内容（What），确定测试的开始和结束日期（When），指出测试的方法和工具（How），给出测试文档和软件的存放位置（Where）。

　　为了使“5W”规则更具体化，需要准确理解被测软件的功能特征、应用行业的知识和软件测试技术，在需要测试的内容里面突出关键部分，可以列出关键及风险内容、属性、场景或者测试技术。对测试过程的阶段划分、文档管理、缺陷管理、进度管理给出切实可行的方法。

　　3. 采用评审和更新机制，保证测试计划满足实际需求测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

　　测试计划包含多方面的内容，编写人员可能受自身测试经验和对软件需求的理解所限，而且软件开发是一个渐进的过程，所以最初创建的测试计划可能是不完善的、需要更新的。需要采取相应的评审机制对测试计划的完整性、正确性、可行性进行评估。

例如，在创建完测试计划后，提交到由项目经理、开发经理、测试经理、市场经理等组成的评审委员会审阅，根据审阅意见和建议进行修正和更新。

　　4. 分别创建测试计划与测试详细规格、测试用例编写软件测试计划要避免一种不良倾向是测试计划的“大而全”，无所不包，篇幅冗长，长篇大论，重点不突出，既浪费写作时间，也浪费测试人员的阅读时间。“大而全”的一个常见表现就是测试计划文档包含详细的测试技术指标、测试步骤和测试用例。