1软件验收测试的合格通过准则是ABCD

A软件需求分析说明书中定义的所有功能已全部实现性能指标全

部达到要求。

B所有测试项没有残余一级、二级和三级错误。

C立项审批表、需求分析文档、设计文档和编码实现一致。

D验收测试工件齐全。

2描述软件测试活动的生命周期

测试周期分为计划、设计、实现、执行、总结。其中计划对整个测试周期中所有活动进行规划估计工作量、风险安排人力物力资源安排进度等，设计完成测试方案从技术层面上对测试进行规划，实现。进行测试用例和测试规程设计，执行根据前期完成的计划、方案、用例、规程等文档执行测试用例。

总结记录测试结果进行测试分析完成测试报告。

3、添空

1.软件验收测试包括正式验收测试alpha测试beta测试。

2.系统测试的策略有功能测试性能测试可靠性测试负载测试易用性测试强度测试安全测试配置测试安装测试卸载测试文挡测试故障恢复测试界面测试容量测试兼容性测试分布测试可用性测试

3.设计系统测试计划需要参考的项目文挡有软件测试计划软件需求工件和迭代计划。

4.对面向过程的系统采用的集成策略有自顶向下自底向上两种。

4.软件测试的目的是尽可能多的找出软件的缺陷。Y

5.Beta测试是验收测试的一种。Y



6.区别阶段评审的与同行评审

同行评审目的:发现小规模工作产品的错误,只要是找错误;

阶段评审目的:评审模块阶段作品的正确性可行性及完整性

同行评审人数:3-7人人员必须经过同行评审会议的培训,由SQA指导

阶段评审人数:5人左右评审人必须是专家具有系统评审资格

同行评审内容:内容小一般文档< 40页,代码< 500行

阶段评审内容:内容多,主要看重点

同行评审时间:一小部分工作产品完成

阶段评审时间:通常是设置在关键路径的时间点上!

7软件测试计划评审会需要哪些人员参加ABCD

A项目经理

BSQA负责人

C配置负责人

D测试组

8简述集成测试的过程

系统集成测试主要包括以下过程

1.构建的确认过程。

2.补丁的确认过程。

3.系统集成测试测试组提交过程。

4.测试用例设计过程。

5.测试代码编写过程。

6. Bug的报告过程。

7.每周/每两周的构建过程。

8.点对点的测试过程。

9.组内培训过程。

9.验收测试是由最终用户来实施的。N

10.项目立项前测试人员不需要提交任何工件。Y

11.软件测试的八个基本原则：

所有的软件测试都应追溯到用户需求。

尽早和不断地进行软件测试。（3分）

在设计测试用例时，应该包括合理的输入与不合理的输入以及相应的预期的输出结果。

充分注意测试中的群集现象。

程序员应避免检查自己的程序。

尽量避免测试的随意性。

应当对每个测试结果做全面的检查。

保留测试文档，包括测试计划、用例、出错统计和最终分析报告。

12.自底向上集成需要测试员编写驱动程序。Y

13.负载测试是验证要检验的系统的能力最高能达到什么程度。N

14怎么做好文档测试

检查文档的编写是否满足文档编写的目的

内容是否齐全正确

内容是否完善

标记是否正确

15白盒测试有几种方法

总体上分为静态方法和动态方法两大类。

静态关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义

动态语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。

5Alpha测试与beta的区别

Alpha测试在系统开发接近完成时对应用系统的测试测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成不能由程序或测试员完成。

Beta测试当开发和测试根本完成时所做的测试最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成不能由程序员或测试员完成。

16比较负载测试容量测试和强度测试的区别

负载测试：在一定的工作负荷下系统的负荷及响应时间。

强度测试：在一定的负荷条件下在较长时间跨度内的系统连续运行给系统性能所造成的影响。

容量测试容量测试目的是通过测试预先分 析出反映软件 系统应用特征的某项指标的极限值如最大并发用户数、数据库记录数等

系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试 还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据 的并且它的目的是显示系统可以处理目标内确定的数据容量。

17软件的缺陷等级应如何划分

A类—严重错误。

1、 由于程序所引起的死机、非法退出 2、死循环 3、 数据库发生死锁 4、 因错误操作导致的程序中断 5、功能错误 6、 与数据库连接错误 7、 数据通讯错误

B类—较严重错误。

1、 程序错误 2、 程序接口错误 3、数据库的表、业务规则、缺省值未加完整性等约束条件

C类—一般性错误。

1、 操作界面错误包括数据窗口内列名定义、含义是否一致 2、 打印内容、格式错误 3

、 简单的输入限制未放在前台进行控制 4、 删除操作未给出提示 5、 数据库表中有过多的空字段

D类—较小错误。

1、 界面不规范 2、 辅助说明描述不清楚 3、 输入输出不规范 4、 长操作未给用户提示 5、 提示窗口文字未采用行业术语 6、 可输入区域和只读区域没有明显的区分标志

E类—测试建议

18您认为做好测试用例设计工作的关键是什么?

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒测试用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试以最少的用例在合理的时间内发现最多的问题

19.单元测试能发现约80%的软件缺陷。Y

20.代码评审是检查源代码是否达到模块设计的要求。N

21下列关于alpha测试的描述中正确的是AD

Aalpha测试需要用户代表参加

Balpha测试不需要用户代表参加

Calpha测试是系统测试的一种

Dalpha测试是验收测试的一种

22.黑盒测试和白盒测试是软件测试的两种基本方法，请分别说明各自的优点和缺点！

答：黑盒测试的优点有：

　　1）比较简单，不需要了解程序内部的代码及实现；

2）与软件的内部实现无关；

　　3）从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题；

　　4）基于软件开发文档，所以也能知道软件实现了文档中的哪些功能；

　　5）在做软件自动化测试时较为方便。

黑盒测试的缺点有：

　　1）不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的30%；

　　2）自动化测试的复用性较低。

　　白盒测试的优点有：帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐藏的问题。

　　白盒测试的缺点有：

　　1）程序运行会有很多不同的路径，不可能测试所有的运行路径；

　　2）测试基于代码，只能测试开发人员做的对不对，而不能知道设计的正确与否，可能会漏掉一些功能需求；

3）系统庞大时，测试开销会非常大。

23针对以下问题:某一种8位计算机，其十六进制常数的定义是以0x或0X开头的十六进制整数，其取值范围为,7f,7f(不区分大小写字母)，如0x13、0x6A、,0x3c。请采用等价类划分的方法设计测试用例。

用例1:0x7F， 覆盖等价类(1)(4)(6)(8)

用例2:-0Xb， 覆盖等价类(1)(4)(6)(8)

用例3:0X0， 覆盖等价类(1)(4)(6)(8)

用例4:0x， 覆盖等价类(1)(7)

用例5:A7， 覆盖等价类(2)

用例6:-1A， 覆盖等价类(3)

用例7:0X8h， 覆盖等价类(1)(5)

用例8:0x80， 覆盖等价类(1)(4)(10) 用例9:-0XaB， 覆盖等价类(1)(4)(9)

24. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

黑盒测试已知产品的功能设计规格可以进行测试证明每个实现了的功能是否符合要求。

白盒测试已知产品的内部工作过程可以通过测试证明每种内部操作是否符合设计规格要求所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子测试人员完全不考虑程序内部的逻辑结构和内部特性只依据程序的需求规格说明书检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。

黑盒测试主要是为了发现以下几类错误:

1、是否有不正确或遗漏的功能

2、在接口上输入是否能正确的接受能否输出正确的结果

3、是否有数据结构错误或外部信息例如数据文件访问错误

4、性能上是否能够满足要求

5、是否有初始化或终止性错误

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子它允许测试人员利用程序内部的逻辑结构及有关信息设计或选择测试用例对程序所有逻辑路径进行测试。通过在不同点检查程序状态确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查

1、对程序模块的所有独立的执行路径至少测试一遍。

2、对所有的逻辑判定取“真”与取“假”的两种情况都能至少测一遍。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性等等。

单元测试模块测试是开发者编写的一小段代码用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言一个单元测试是用于判断某个特定条件或者场景下某个特定函数的行为。

单元测试是由程序员自己来完成最终受益的也是程序员自己。可以这么说程序员有责任编写功能代码同时也就有责任为自己的代码编写单元测试。执行单元测试就是为了证明这段代码的行为和我们期望的一致。

集成测试也叫组装测试联合测试是单元测试的逻辑扩展。它的最简单的形式是两个已经测试过的单元组合成一个组件并且测试它们之间的接口。从这一层意义上讲组件是指多个单元的集成聚合。在现实方案中许多单元组合成组件而这些组件又聚合成程序的更大部分。方法是测试片段的组合并最终扩展进程将您的模块与其他组的模块一起测试。最后将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。

系统测试的目的是对最终软件系统进行全面的测试确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪并且可以让最终用户将其用于执行软件的既定功能和任务。验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后已经按照设计把所有的模块组装成一个完整的软件系统接口错误也已经基本排除了接着就应该进一步验证软件的有效性这就是验收测试的任务即软件的功能和性能如同用户所合理期待的那样。

25. 测试计划工作的目的是什么测试计划工作的内容都包括什么其中哪些是最重要的

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。

测试计划和测试详细规格、测试用例之间是战略和战术的关系测试计划主要从宏观上规划测试活动的范围、方法和资源配置而测试详细规格、测试用例是完成测试任务的具体战术。所以其中最重要的是测试测试策略和测试方法最好是能先评审

26测试设计员的职责有BC

A制定测试计划

B设计测试用例

C设计测试过程、脚本

D评估测试活动

27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。

4. 分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系测试计划主要从宏观上规划测试活动的范围、方法和资源配置而测试详细规格、测试用例是完成测试任务的具体战术。

28. 您所熟悉的测试用例设计方法都有哪些请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

1等价类划分

划分等价类: 等价类是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.并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况:有效等价类和无效等价类.

2边界值分析法

边界值分析方法是对等价类划分方法的补充。测试工作经验告诉我,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以查出更多的错误.

使用边界值分析方法设计测试用例,首先应确定边界情况.通常输入和输出等价类的边界,就是应着重测试的边界情况.应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据.

3错误推测法

基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法.

错误推测方法的基本思想: 列举出程序中所有可能有的错误和容易发生错误的特殊情况,根据他们选择测试用例. 例如, 在单元测试时曾列出的许多在模块中常见的错误. 以前产品测试中曾经发现的错误等, 这些就是经验的总结. 还有, 输入数据和输出数据为0的情况. 输入表格为空格或输入表格只有一行. 这些都是容易发生错误的情况. 可选择这些情况下的例子作为测试用例.

4因果图方法

前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系, 相互组合等. 考虑输入条件之间的相互组合,可能会产生一些新的情况. 但要检查输入条件的组合不是一件容易的事情, 即使把所有输入条件划分成等价类,他们之间的组合情况也相当多. 因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例. 这就需要利用因果图逻辑模型. 因果图方法最终生成的就是判定表. 它适合于检查程序输入条件的各种组合情况.