

软件测试与质量保证期末考试复习资料

原创：林大夕可

 阅读原文 

2020.01.05

一、题型

二、重点

1. 选择与填空
2. 判断
3. 简答
 - 3.1 单元、集成、系统测试策略
 - 3.2 单元、集成、系统测试异同
 - 3.3 面向对象的单元/集成测试概念
 - 3.4 黑盒/白盒测试定义、执行过程
 - 3.5 黑盒/白盒测试方法
 - 3.6 自动化测试/手工测试的优缺点
4. 了解
 - 4.1 软件测试的原则
 - 4.2 Bug相关
 - 4.3 标准等价类与健壮等价类
 - 4.4 负载测试与压力测试
5. 大题

一、题型

1. 选择题 20分 20题
2. 判断题 10分 10~15个
3. 填空题 10分 10空
4. 简答题 15分 3个
5. 分析题 10分 2个
6. 综合题 35分 2个

注：逻辑覆盖、等价类划分、自动化测试不会有大题（了解概念即可）

二、重点

1. 选择与填空

- 1. 软件测试的对象：既包括源程序，又包括需求规格说明、概要设计说明、详细设计说明
- 2. 软件测试的基本职责：验证、确认
- 3. 软件测试技术分类：
 - 1. 按是否需要执行被测软件划分：静态测试、动态测试
 - 2. 按测试用例设计方法划分：黑盒测试、白盒测试（详见下方）
 - 3. 按测试策略和过程划分：单元测试、集成测试、确认测试、系统测试、验收测试
- 4. 静态测试：代码审查、走查，桌面检查；
动态测试：运行软件
- 5. 测试与调试的区别：
 - 1. 测试的**目的**是发现错误，调试的目的是修正错误
 - 2. 测试的**对象**是各阶段产生的所有产品，调试的对象主要是源代码
 - 3. 测试的**过程**是有计划的、可以重复的，调试的过程是随机的不可重复的
 - 4. 调试不能完全**替代**测试
- 6. 测试与调试的共同点：为了提高软件质量
- 7. 测试阶段与开发过程的对应关系

测试阶段	开发过程	测试重点
单元测试	详细设计	编码错误
集成测试	概要设计	接口错误
确认测试	需求分析	合乎用户预期
系统测试	需求定义	整体有效运行

- 8. V模型：明确地表明了测试过程中存在的不同级别，并且清楚地描述了这些测试阶段和开发过程期间的对应关系
- 9. W模型：V模型的改进，更好地体现了软件开发与软件测试工作的同步性，指出了测试伴随着整个软件开发周期，测试的对象不仅仅是程序，还包括需求和设计。每个软件开发活动结束后就可以执行相应的测试
- 10. H模型：从微观的角度来看待软件测试的过程
- 11. 回归测试：测试由于修正缺陷而更新的应用程序，以确保修正了旧缺陷而未引入新缺陷
- 12. α测试：由用户在**开发环境下**进行的测试，也可以是开发机构内部的用户在模拟实际操作环境下进行的测试。用户在开发者受控的环境下进行测试。

β测试：由多个用户在一个或多个用户的**实际使用环境下**进行的测试。开发者通常不在测试现场。

13. 单元测试中用来模拟被测模块调用者的模块是驱动模块

单元测试中用来代替所测模块调用的子模块是桩模块

14. 使用白盒测试方法时，确定测试用例应根据程序的**内部逻辑**和**指定的覆盖标准**。

15. 测试过程中，**测试计划**用于描述测试的整体方案，**缺陷报告**描述依据测试案例找出的问题。

16. 白盒测试又称为**结构测试**，黑盒测试又称为**功能测试**。

17. 黑盒测试用例设计方法包括**等价类划分法**、**边界值分析法**以及**因果图法**，错误推测法等。

18. 传统的等价类划分测试的实现分两步进行，一是**划分等价类表**，二是**边界值分析法**。

19. 某种程度上，可以把自动化测试工具叫做回归测试工具

20. 逻辑覆盖能力强弱：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

2. 判断

1. V模型要求开发和测试保持一种线性的前后关系。

错。出自书P.6

2. 测试是为了证明程序能够正确工作。

错。出自书P.14

3. 一个好的测试用例在于他能发现至今未发现的错误。

对。出自书P.15

4. 一个成功的测试是发现了至今未发现的错误的测试。

对。出自书P.15

5. 测试一般在软件开发结束后进行。

错。出自书P.29

6. 自动化测试可以提高测试的质量。

错。出自书P.30

7. 白盒测试的条件覆盖标准强于判定覆盖。（×）

8. 静态白盒测试可以找出遗漏之处和问题（√）

9. 黑盒测试方法中最有效的是因果图法（×）

10. 边界测试中所选择的输入测试数据一定是有效数据（×）

11. 白盒测试仅与程序的内部结构有关,完全可以不考虑程序的功能要求（√）

12. 软件测试员无法对产品说明书进行白盒测试（√）

13. 单元测试能发现约80%的软件缺陷（√）

14. 自顶向下集成需要测试员编写驱动程序 (x)
 15. 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 (x)
测试软件的数据吞吐量上限
 16. 错误推测法是根据输出对输入的依赖关系来设计测试用例的 (x)
错误推测法是基于经验和直觉推测程序中所有可能出现存在的各种错误，从而有针对性地设计测试用例
 17. 测试用例应由测试输入数据和对应的实际输出结果这两部分组成 (x)
 18. 每一个软件项目都有一个最优的测试量 (√)
 19. 黑盒测试往往会造成测试用例之间可能存在严重的冗余和未测试的功能漏洞 (√)
 20. 软件测试是一个过程,包含若干活动,运行软件进行测试只是活动之一 (√)
 21. 集成测试是为确定软件系统是否满足验收标准以及使客户决定是否接受而进行的正式测试 (x)
 22. 永远有缺陷类型会在测试的一个层次上被发现,并且能够在另一个层次上逃避检测 (√)
-

3. 简答

3.1 单元、集成、系统测试策略

1. 单元测试策略：自顶向下的单元测试、自底向上的单元测试、孤立测试
 2. 集成测试策略：基于分解的集成、三明治集成、修改过的三明治集成、基于调用图的集成、基于路径的集成、分层集成、基于功能的集成、高频集成、基于进度的集成、基于风险的集成、基于事件的集成、基于使用的集成、客户/服务器的集成
 3. 系统测试策略：功能测试、协议一致性测试、性能测试、容量测试、安全性测试、恢复性测试、备份测试、GUI测试、健壮性测试、兼容性测试、可用性测试、可安装性测试、文档测试、在线帮助测试、数据转换测试、验收测试
-

3.2 单元、集成、系统测试异同

1. 从时间上看，测试的时间先后顺序依次是单元测试、集成测试、系统测试；
 2. 从测试对象来看，单元测试的对象是实现具体功能的单元，集成测试的对象是接口以及模块组合，系统测试的对象是整个系统的功能；
 3. 从测试方法来看，单元测试的主要测试方法是基于代码的**白盒测试**，集成测试的主要测试方法是基于功能的**黑盒测试**；
 4. 从测试角度来看，单元测试是从开发者的角度考虑的，系统测试（验收测试）则是从用户的角度出发的
-

3.3 面向对象的单元/集成测试概念

1. 单元测试

针对程序模块进行正确性检验的测试工作。其目的在于发现每个程序模块内部可能存在的差错

2. 集成测试

将程序模块采用一次性或增值方式组装起来，对系统的接口进行正确性检验的测试工作

3. 系统测试

将需测试的软件与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素及环境结合在一起的测试

3.4 黑盒/白盒测试定义、执行过程

1. 黑盒测试：又称**功能**测试、数据驱动测试。测试时忽略程序内部特性，在只知道输入输出关系或程序功能的情况下，依靠需求规格说明书，来确定测试用例、推断测试结果。
 2. 白盒测试：又称**结构**测试、逻辑驱动测试。针对特定条件和循环设计测试用例，对软件的逻辑路径进行测试。
-

3.5 黑盒/白盒测试方法

1. 黑盒测试：等价类划分、边界值分析、因果图、决策表、错误推测
 2. 白盒测试：逻辑覆盖（语句覆盖、判定覆盖、条件覆盖、判定-条件覆盖、路径覆盖）、边界值分析、基本路径测试、循环语句测试、程序插装、数据流测试、变异测试
-

3.6 自动化测试/手工测试的优缺点

自动化测试**优点**：

1. 回归测试
2. 运行更多更频繁的测试
3. 进行一些手工测试难以完成的测试（如压力测试、并发测试）
4. 充分利用资源
5. 测试具有一致性和可重复性
6. 测试具有复用性
7. 缩短软件发布时间
8. 增强软件的可靠性

自动化测试**缺点**：

1. 不能取代手工测试，测试主要还是靠人工
2. 新缺陷越多，自动化测试失败的几率就越大
3. 工具本身不具有想象力

4. 技术问题、组织问题、脚本维护
 5. 测试工具与其他软件的互操作性
-

4. 了解

4.1 软件测试的原则

1. 尽早、不断地进行测试
 2. 不可能完全的测试
 3. 增量测试
 4. 避免测试自己的程序
 5. 设计周密的测试用例
 6. 注意错误集中的现象
 7. 确认Bug的有效性
 8. 合理安排测试计划
 9. 回归测试
 10. 测试结果的统计与分析
 11. 及时更新测试
-

4.2 Bug相关

1. 概念：从内部看，软件缺陷是软件产品开发或维护过程中所存在的错误、毛病等各种问题；从外部看，软件缺陷是系统所需要实现的某种功能的失效。
2. 常见类型
 - 功能、特性的缺失
 - 设计缺陷导致运行结果和预期不一致
 - 运行出错（运行中断、系统崩溃、界面混乱）
 - 数据结果不正确、精度不够
 - 存取时间过长、界面不美观等
3. 产生原因
 - 程序编写错误
 - 需求变更过于频繁
 - 软件的复杂度
 - 交流不充分/沟通出问题
 - 测试人员的经验与技巧不足
 - 时间过于紧迫
 - 缺乏文档
 - 管理上的缺陷
4. 分类

- 需求阶段的bug
- 分析设计阶段的bug
- 实现阶段的bug
- 配置阶段的bug
- 短视将来的bug
- 静态文档的bug

5. 执行过程：？

6. 管理流程（可能考简答题或是分析题）：详见书P. 331

4.3 标准等价类与健壮等价类

标准等价类：不考虑无效数据

健壮等价类：考虑无效等价类

4.4 负载测试与压力测试

性能测试（Performance Test）：通常收集所有和测试有关的所有性能，通常被不同人在不同场合下进行使用。测试软件在系统中的运行性能，度量系统与预定义目标的差距。

关注点：how much和how fast

负载测试（Load Test）：负载测试是一种性能测试，指数数据在超负荷环境中运行，程序是否能够承担。通过逐步增加系统负载，确定在满足性能指标的情况下，系统所能承受的最大负载量。

关注点：how much

压力测试（Stress Test）：压力测试是一种高负载下的负载测试，也就是说被系统处于一个负载的情况，再继续对他进行加压，形成双重负载，知道系统崩溃，并关注崩溃后系统的恢复能力，以前再加压的一个过程，看看系统到底是否已经被彻底破坏掉了。

5. 大题

1. 控制流图、独立路径
2. 因果图（重点）

步骤：因果图 > 判定表（决策表） > 测试用例

3. 自顶向下、自底向上、三明治🥪集成

复习结束

那么...

与君👉歌一曲👈，请君为我倾耳听~

我的其它相关文章

- [Python期末考试总复习资料](#)
- [Oracle期末考试总复习资料](#)
- [嵌入式期末复习重点](#)
- [离散数学期末复习笔记](#)