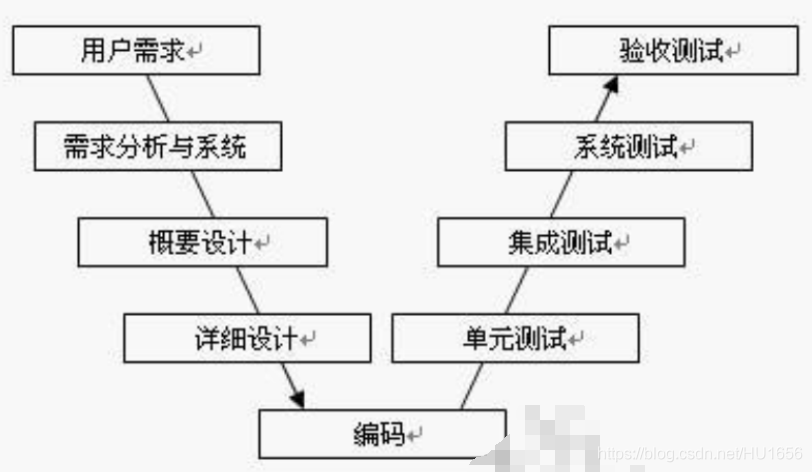
**Software Testing Model：**

**V Model is an enhanced version of the classic waterfall model whereby each level of the development life-cycle is verified before moving on to the next level.**

**With this model, software testing explicitly starts at the very beginning, i.e. as soon as the requirements are written.**

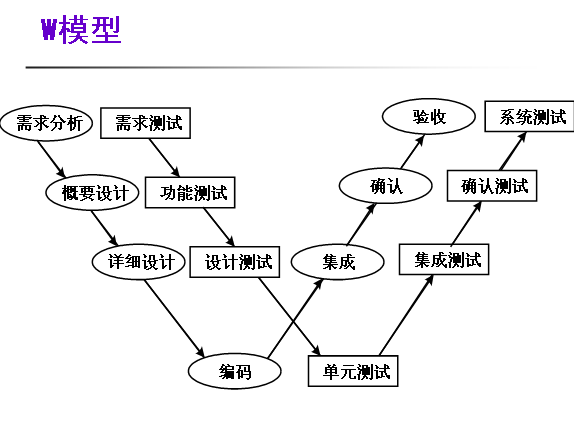


**In a W-Model, each stage must be finished completely before the following stage can start.**

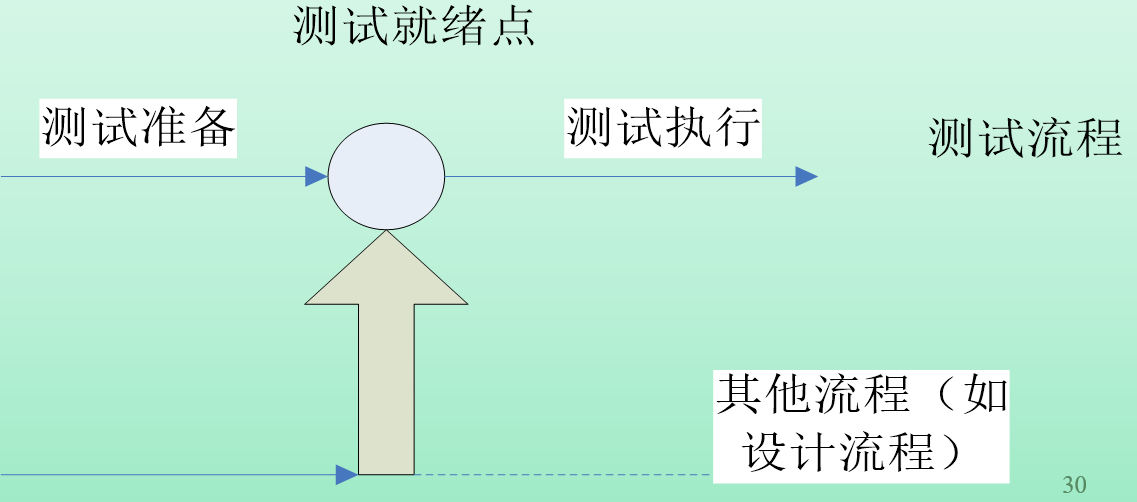
**测试与开发并行，各阶段同步进行验证和确认**

**伴随整个开发周期；不仅程序，需求、设计和功能都测试；**

**一旦有文档提供，就要及时确定测试的条件、编写测试用例**



**H模型：软件测试是一个独立的流程，贯穿产品整个生命周期，与其他流程并发的进行；尽早准备尽早执行；分层进行，可能顺序也可能反复。**

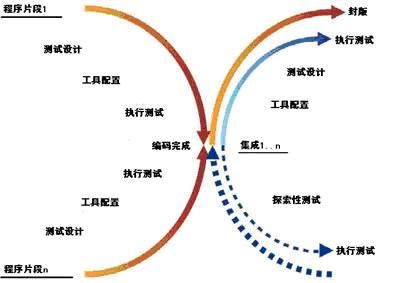


**X模型提出针对单独的程序片段进行相互分离的编码和测试,此后通过频繁的交接,通过集成最终合成为可执行的程序。**

**己通过集成测试的成品可以进行封装并提交给用户，也可以作为更大规模和范围内集成的一部分。**

**多根并行的曲线表示变更可以在各个部分发生。**

关键就是在于有交叉测试以及对代码片段的测试



Software testing（what does a software tester do?）

**to find bugs, find them as early as possible, and make sure they get fixed.**

Software bug：

1. **Doesn’t do something it should do.**
2. **Does something it shouldn’t do.**
3. **Does something it doesn’t mention.**
4. **Doesn‘t do something it doesn’t mention but should.**
5. **It’s difficult to understand, hard to use, slow, or——in the software tester's eyes——will be viewed by the end user as just plain not right.**

黑白灰盒测试概念：

**黑盒测试又称功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。**

**主要方法：**

* 等价类划分方法
* 边界值分析方法
* 错误推测方法
* 因果图方法
* [判定表方法]

**白盒测试也称结构测试或逻辑驱动测试，是一种测试用例设计方法，它从程序的控制结构导出测试用例。**

**白盒测试使用被测单元内部如何工作的信息，允许测试人员对程序内部逻辑结构及有关信息来设计和选择测试用例，对程序进行测试。**

**主要方法：**

**逻辑驱动测试：语句覆盖、判定覆盖（分支覆盖）、判定/条件覆盖、条件组合覆盖**

**路径测试：基本路径测试**

静态测试：通过检查和评审软件而不是运行软件对软件进行测试的方法。静态测试可以手工进行，也可以借助软件工具自动进行。方法：审查、走查、桌面检查 内容：需求定义、文档、源代码的静态测试。

Test-Driven Development（TDD，测试驱动开发）

TDD测试驱动开发的基本思想就是在开发功能代码之前，先编写测试代码。也就是说在明确要开发某个功能后，首先思考如何对这个功能进行测试，并完成测试代码的编写，然后编写相关的代码满足这些测试用例。然后循环进行添加其他功能，直到完全部功能的开发

集成测试

集成测试又称组装测试，是在单元测试的基础上，将所有模块按照设计要求组装成子系统或系统进行的测试活动。

软件质量的基本属性、需求：10个

可理解、可靠、可测试、可修改、可移植、效率、可使用

driver

用以模拟被测模块的上级模块。驱动模块在集成测试中接受测试数据，把相关的数据传送给被测模块，启动被测模块，并打印出相应的结果。

驱动模块（Driver）：所测模块的主程序：它接收测试数据，把这些数据传递给所测试模块，最后再输出实测结果。当被测试模块能完成一定功能时，也可以不要驱动模块。

桩模块（Stub）：用来代替所测模块调用的子模块。

α、β版本

-- α 是第一阶段，一般只供内部测试使用；

-- β是第二个阶段，已经消除了软件中大部分的不完善之处，但仍有可能还存在缺陷和漏洞，一般只提供给特定的用户群来测试使用；

-- λ是第三个阶段，此时产品已经相当成熟，只需在个别地方再做进一步的优化处理即可上市发行。

Acceptance Testing

在软件产品完成了功能测试和系统测试之后、产品发布之前所进行的软件测试活动它是技术测试的最后一个阶段,也称为交付测试。

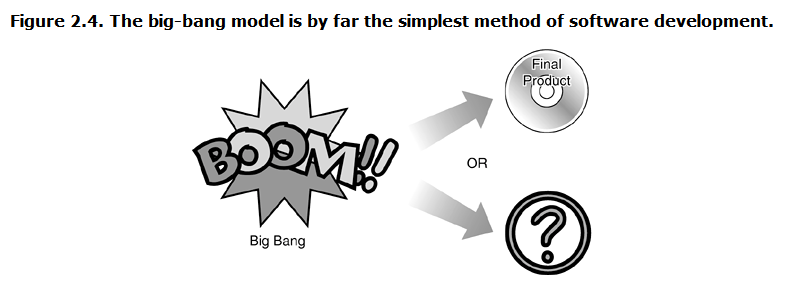
Version Control

版本控制就是管理在整个软件生存周期中建立起来的某一配置项的不同版本。

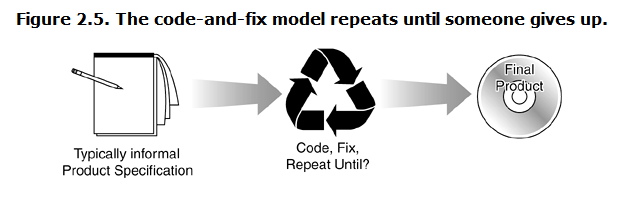
在软件工程过程中所涉及的软件对象都要加以标识。在对象成为基线以前可能要做多次变更，在成为基线之后也可能需要频繁地变更。这样对于每一配置对象可以建立一个演变图，以方便记叙这个对象的变更历史

开发方式：

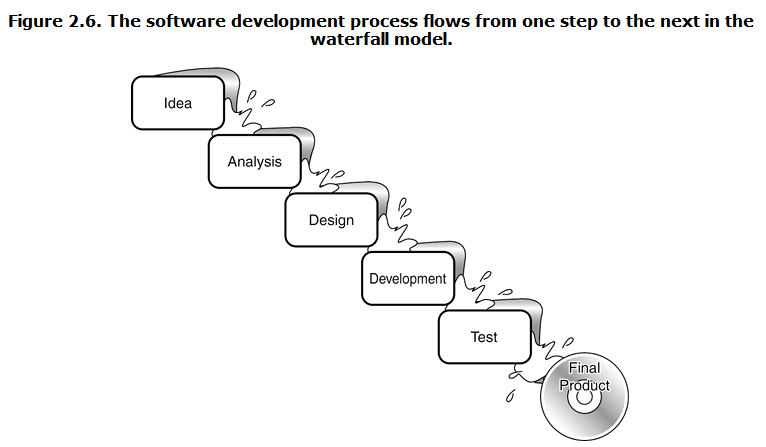
Big-Bang



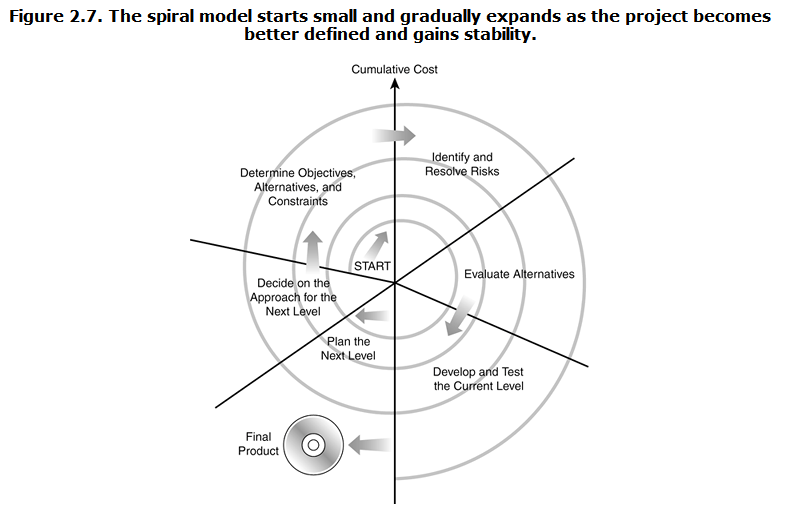
Code-and-Fix



Waterfall



Spiral



简答题：

QC和QA？

软件测试员（QC）：检测软件产品

　　找出缺陷，

　　评价软件质量

质量保证人员（QA）：检测软件过程

　　创建和加强促进软件开发并防止软件缺陷的标准、方法和过程。

**SQA与软件测试的关系**

* **SQA 是管理工作、审查对象是流程、强调以预防为主**
* **测试是技术工作、测试对象是产品、主要是事后检查**
* **SQA指导测试、监控测试**
* **测试为SQA提供依据，是它的一个环节和手段**

How to understand the relationship between specification and bugs?

The bugs are:

The software does not do something that the specification says it should do.

The software does something that the specification says it should not do.

The software does something that the specification does not mention.

The software does not do something that the product specification does not mention but should.

The software is difficult to understand, hard to use, slow …

Please describe the categories of software maintenance

改正性维护、适应性维护、完善性维护、预防性维护

1、诊断和修正遗留错误、缺陷。

2、修改软件适应外部环境和数据环境。

3、满足新的需求，扩充功能、增强性能、改进效率、提高可维护性。

4、提高可维护性、可靠性等，重新设计、编制和测试。

--Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.

--Adaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.

--Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.

--Preventive maintenance: Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

Functional test is? What are the methods used to complete functional testing?

   功能性测试也被称为E2E测试，或者浏览器测试。功能测试被解释为测试应用程序的全部的完整的功能。对于互联网应用程序来说，这意味着用一些工具在浏览器上自动化测试，在页面上自动点击来测试应用程序。

人工手动测试或使用自动化工具，如micro Focus的UFT自动测试工具，创建测试方案、添加动作、验证点等进行测试。进行一系列操作并达到预期效果即为测试通过。

测试退出的标准？是不是从头到尾，是不是bug越多越好

？

测试的范围：（方法：提问单，）

* + 首先测试最高优先级的需求。
  + 测试新的功能和代码或者改进的旧功能。
  + 使用等价类划分来减小测试范围
  + 重点测试经常出问题的地方

测试团队配置？

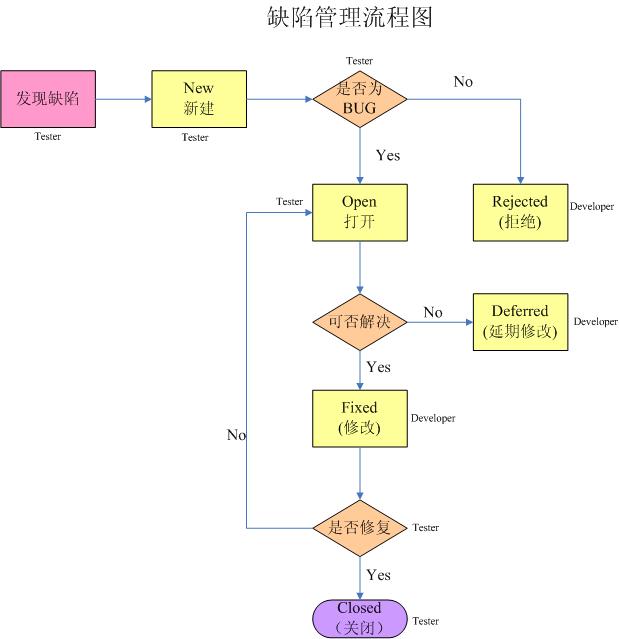
测试经理、环境管理人员、 测试组长, 测试设计人员，初级测试工程师，发布工程师、配置管理员

自动化测试和人工测试的比对

* 并不是所有的测试工作都可以由测试工具来完成
* 并不是一个自动化工具就可以完成所有的测试
* 使用自动化工具本身也是需要时间的，这个时间有可能超过手工测试的时间
* 如果测试人员不熟悉测试工具的使用，有可能不能更多发现软件错误，从而影响测试工作质量
* 自动化测试工具并不能对一个软件进行完全的测试
* 购买自动化测试工具，有可能使本项目的测试费用超出预算

软件测试的现实

缺陷管理流程Briefly describe the bug management process？



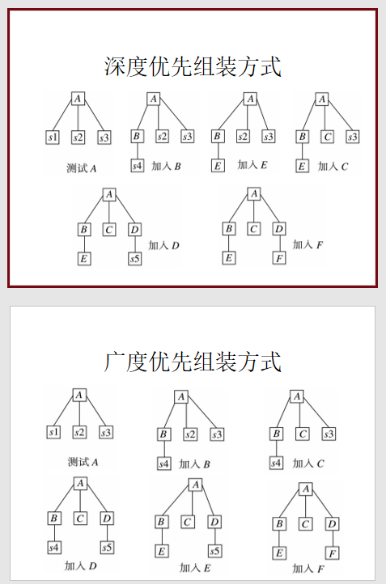
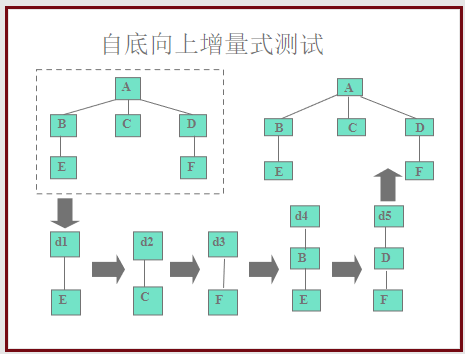
集成策略：

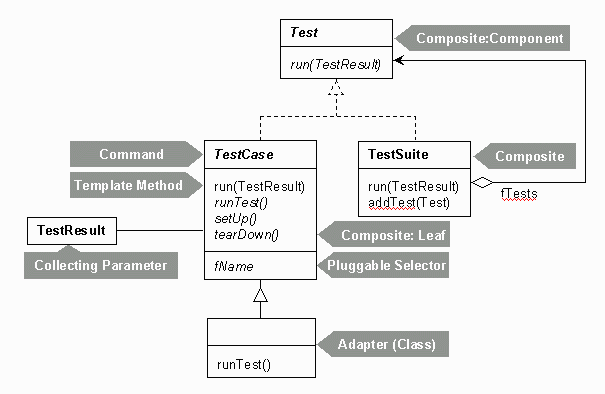
* + 非增量式集成策略——一步到位（又叫大爆炸式集成）
  + 增量式集成策略——逐步实现

增量式集成方法和它们的比对：

Please descricbe the difference between Top-down Integration and Bottom-up Integration through drawing their model graph？

步骤？优缺点分析；图示如下：

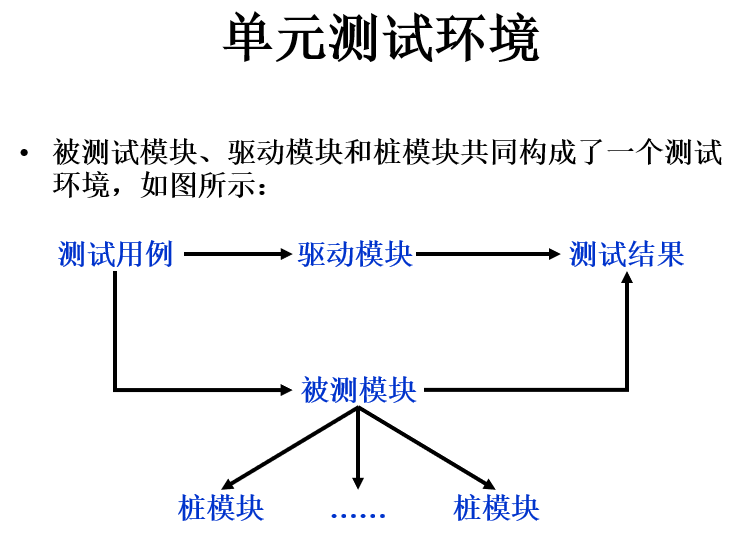
 

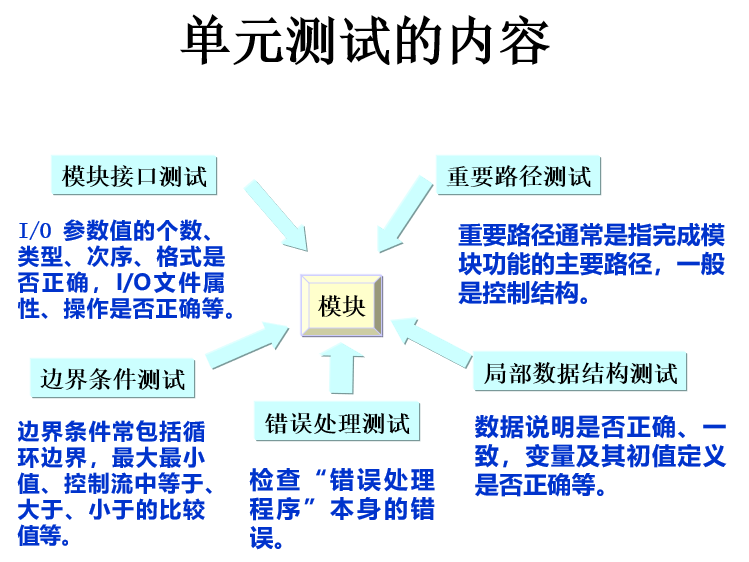
1. Briefly describe JUnit framework through drawing its structure graph? （6 points）

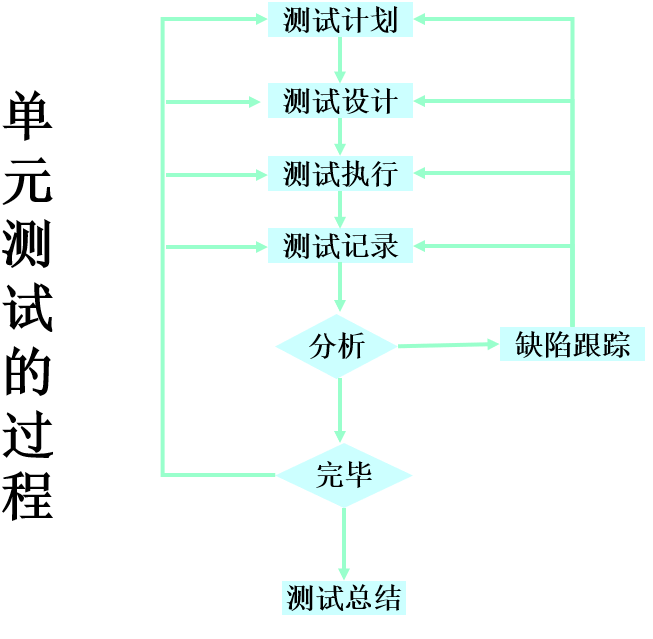
为什么避免测试自己的程序？

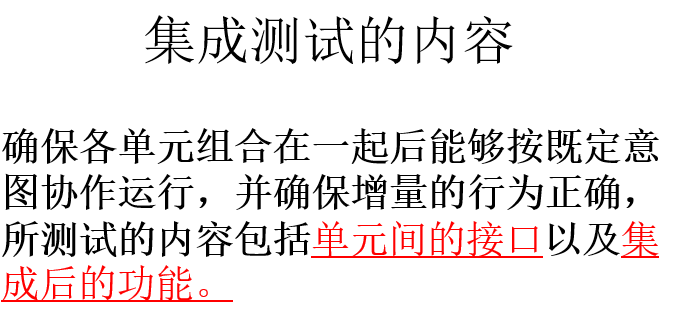
源代码的静态测试标准：

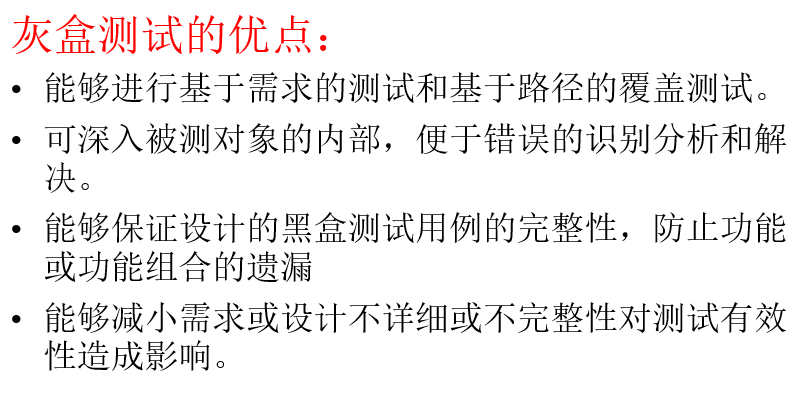
完备性、一致性、正确性、易修改性、可预测性、健壮性、结构化、易追溯性、易理解性、可验证性。







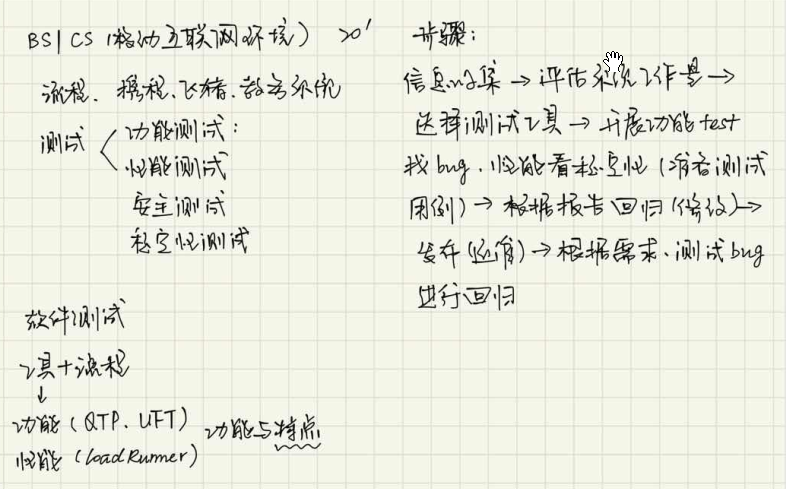




工具：（功能与特点）

功能测试（QTP、UFT）：功能与特点

性能（load runner）



分析题，测试用例（黑盒、白盒各1）

白盒：

考点 program flow chart?

Condition combinations of program flow graph

Design testcase (覆盖等级)

决策表法

黑盒：

等价类划分

使用等价类设计测试用例

综合题（如何做软件测试、维护，从技术和管理角度）

（重）广东电网、上海通用、大型系统测试，结合案例（理论研究，没有没有大型工具是进行软件测试）

According the graph, please give us the Testing Strategy to finish the Testing?（10 points）

组建测试队伍、时间安排、压力测试,功能测试

压力测试的测试策略（strategy）

怎么准备测试环境

14章ppt：

测试环境包括：设计环境、实施环境、管理环境；是为完成测试工作所必须的软件、硬件、设备、数据的总称。影响到测试结果的真实性和正确性。要求：稳定可控少变动（便于重现缺陷）

要满足的五个基本要素：硬件（pc、配置）、软件（操作系统、数据库、中间件、web服务器）、网络环境、数据准备（数据量、真实性）、测试工具。

测试的计划阶段，理解客户需求、掌握被测软件的基本特性、获取资源、合理调度。

搭建软件测试实验室：

1 机房环境建设

2 硬件环境的建立

3 网络环境的建立

4 软件环境的建立

5 测试环境说明及备案

测试环境的管理与维护：

1 设置专门的测试环境管理员的角色

2 明确测试环境管理所需的各种文档

3 测试环境访问权限的管理

4 测试环境的变更管理

5 测试环境的备份和恢复

15章ppt：软件测试计划

