College of Computer Science, Chongqing University

Software Engineering

A Practitioner's Approach Seventh Edition

22 软件测试策略

zmqmail@cqu.edu.cn 13708390417

重庆大学 计算机学院 COLLEGE OF COMPUTER SCIENCE

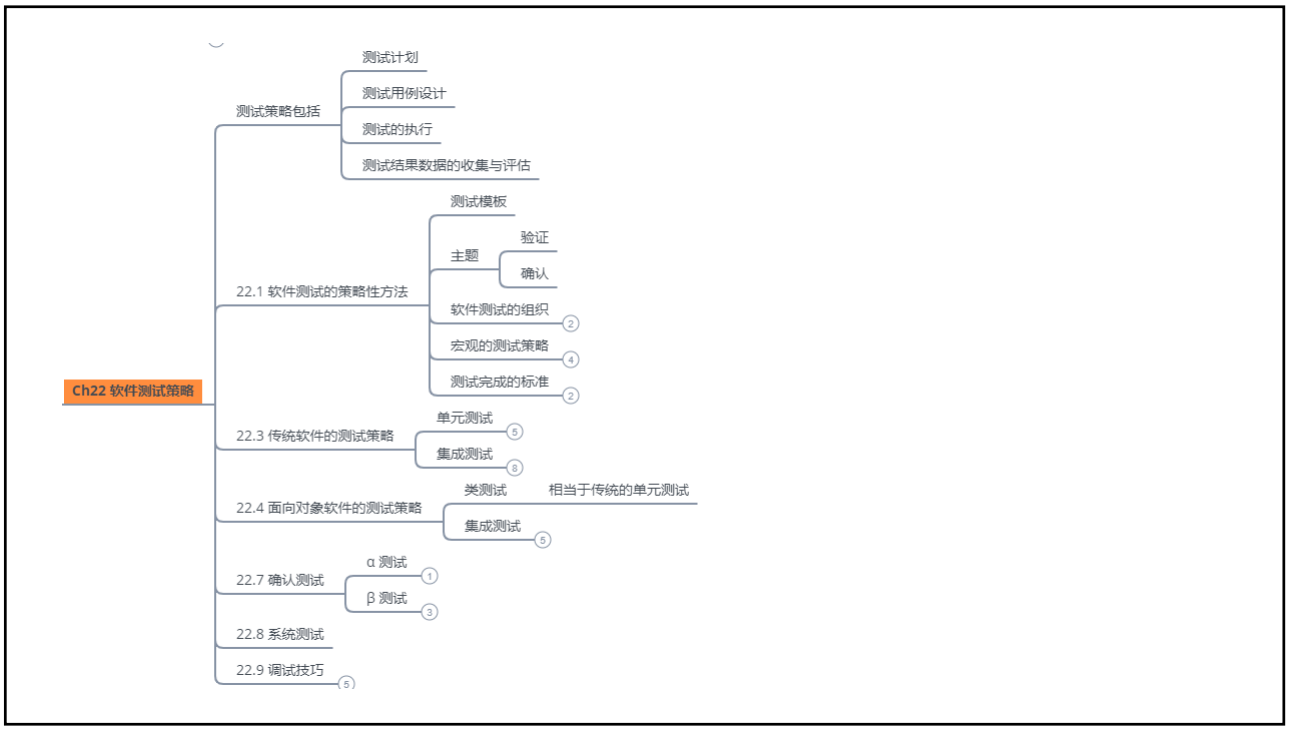
软件测试

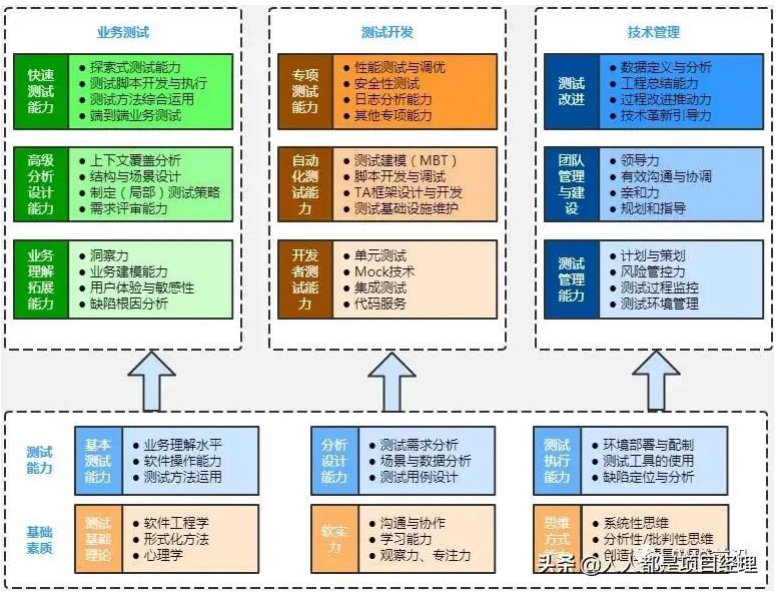
Software Testing

Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.

保证业务逻辑的正确性?

编写测试用例是最好的方法
为什么呢？因为测试用例可以反复使用。


Ch22 软件测试策略
测试策略包括
测试计划
测试用例设计
测试的执行
测试结果数据的收集与评估
22.1 软件测试的策略性方法
测试模板
主题
验证
确认
软件测试的组织
2
宏观的测试策略
4
测试完成的标准
2
22.3 传统软件的测试策略
单元测试
5
集成测试
8
22.4 面向对象软件的测试策略
类测试
相当于传统的单元测试
集成测试
5
22.7 确认测试
α 测试
1
β 测试
3
22.8 系统测试
22.9 调试技巧
5


业务测试
快速测试能力
探索式测试能力
测试脚本开发与执行
测试方法综合运用
端到端业务测试
高级分析设计能力
上下文覆盖分析
结构与场景设计
制定（局部）测试策略
需求评审能力
业务理解拓展能力
洞察力
业务建模能力
用户体验与敏感性
缺陷根因分析
测试开发
专项测试能力
性能测试与调优
安全性测试
日志分析能力
其他专项能力
自动化测试能力
测试建模（MBT）
脚本开发与调试
TA框架设计与开发
测试基础设施维护
开发者测试能力
单元测试
Mock技术
集成测试
代码服务
技术管理
测试改进
数据定义与分析
工程总结能力
过程改进推动力
技术革新引导力
团队管理与建设
领导力
有效沟通与协调
亲和力
规划和指导
测试管理能力
计划与策划
风险管控力
测试过程监控
测试环境管理
测试能力
基本测试能力
业务理解水平
软件操作能力
测试方法运用
分析设计能力
测试需求分析
场景与数据分析
测试用例设计
测试执行能力
环境部署与配制
测试工具的使用
缺陷定位与分析
基础素质
测试基础理论
软件工程学
形式化方法
心理学
软实力
沟通与协作
学习能力
观察力、专注力
思维方式
系统性思维
分析性/批判性思维
创造性思维
头条@大都是项目经理

软件测试策略提供了一张路线图

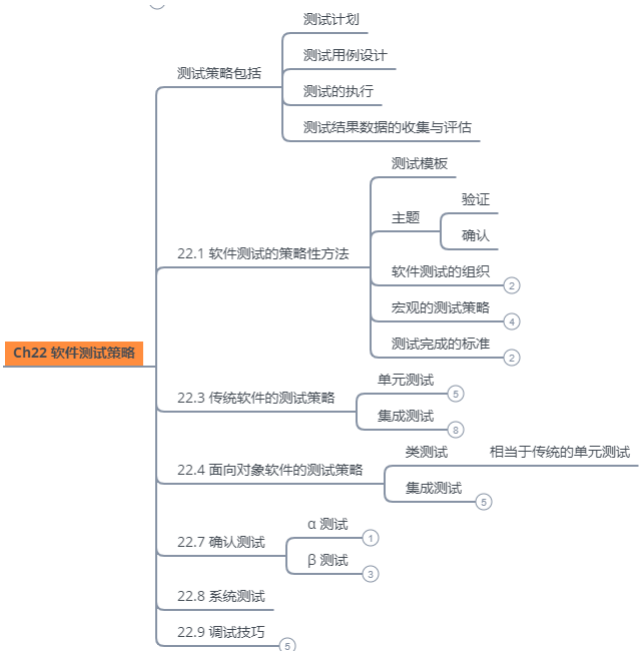
任何测试策略都必须包含测试计划、测试用例设计、测试执行以及测试结果数据的收集与评估。

或者说要做的工作!

软件测试策略

软件测试策略由项目经理、软件工程师以及测试专家共同制定

- 从“小范围”开始逐步过渡到“软件整体”
  - 测试规格说明
  - 将软件测试团队的具体测试方法文档化
  - 包括制定描述整体策略的计划，定义特定测试步骤的规程以及将进行测试的类型
- 包括测试计划、测试用例设计、测试执行以及测试结果数据收集与评估



Low-level Tests

Necessary to verify that a small source code segment has been correctly implemented

验证小段源代码是否正确实现

High-level Tests

Validate major system functions against customer Requirements

主要功能是否满足用户需求

谁来测试软件? (Who Tests the Software?)

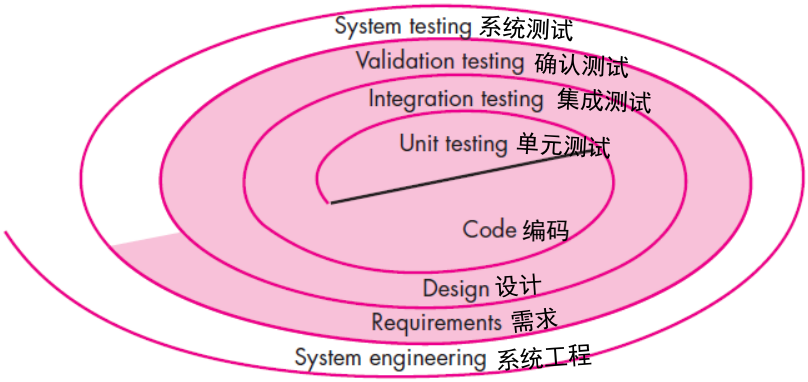
Developer

Understands the system but, will test "gently" and, is driven by "delivery"

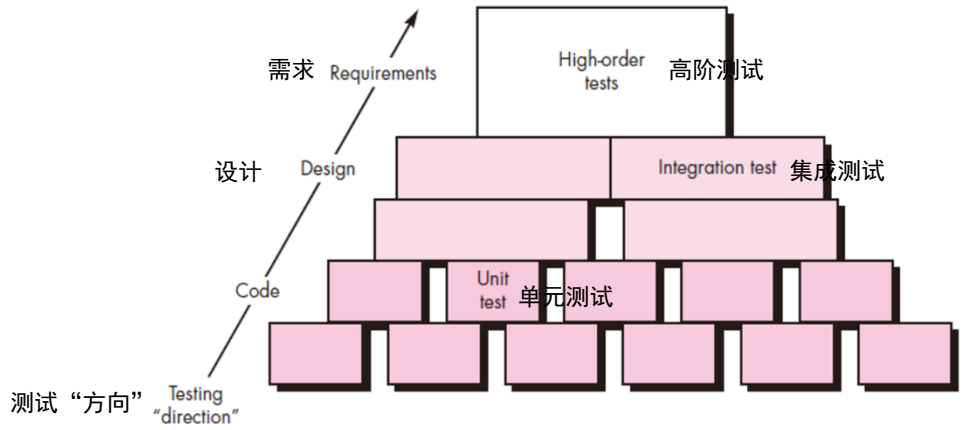
Independent Tester

Must learn about the system, but, will attempt to break it and, is driven by quality

Testing Strategy



Software Testing Steps



你永远也不能完成测试！

担子只会从你(软件工程师)身上转移到最终用户身上；
当你的时间或资金艳尽时，濒时就完成了。

22.2 策略问题 (Strategic Issues)

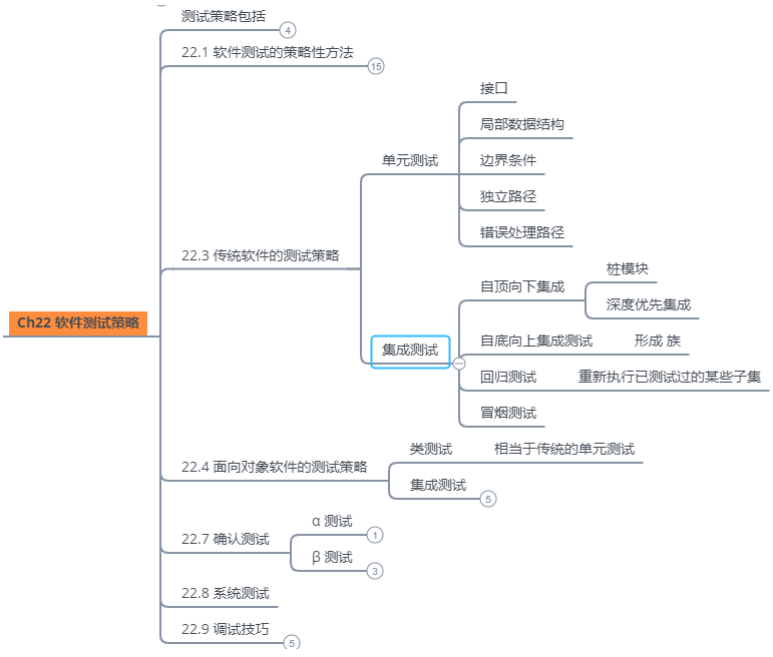
- **Specify** product **requirements** in a quantifiable manner long before testing commences.
  测试前即**量化规定产品需求**
- State testing objectives explicitly.
  明确陈述测试目标
- Understand the users of the software and develop a profile for each user category.
  了解软件用户并为每类用户建立用户描述
- Develop a testing **plan** that emphasizes "rapid cycle testing."
  制定强调"快速周期测试"的测试计划

22.2 策略问题 (Strategic Issues)

- Build "robust" software that is designed to test itself.
  建立能测试自身的"健壮"软件
- Use effective technical reviews as a filter prior to testing.
  测试前利用有效的**正式技术评审作为过滤器**
- Conduct technical reviews to assess the test strategy and test cases themselves.
  实施正式技术评审以评估测试策略和测试用例本身
- Develop a continuous improvement approach for the testing process.
  为测试过程建立一种持续的改进方法

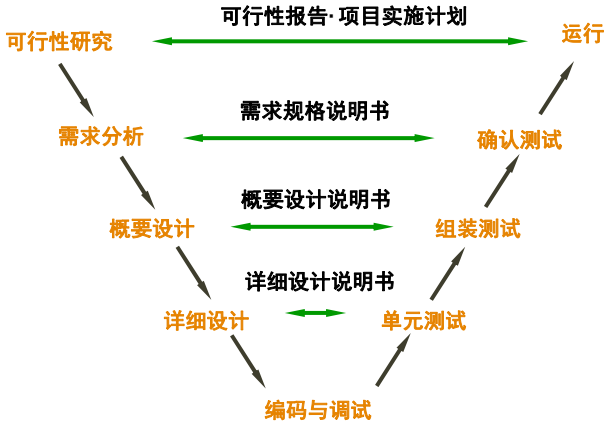
22.2 测试策略(Testing Strategy)

- We begin by ‘Testing-in-the-small’ and move toward ‘Testing-in-the-large’
- 传统软件的测试策略(**Conventional Software**)
  - The module (component) is our initial focus
  - Integration of modules follows
- 面向对象软件的测试策略(**OO Software**)
  - Our focus when “testing in the small” changes from an individual module (the conventional view) to an OO class that encompasses attributes and operations and implies communication and collaboration



V模型?

- 测试过程是依**相反顺序安排**的自底向上，逐步集成的过程

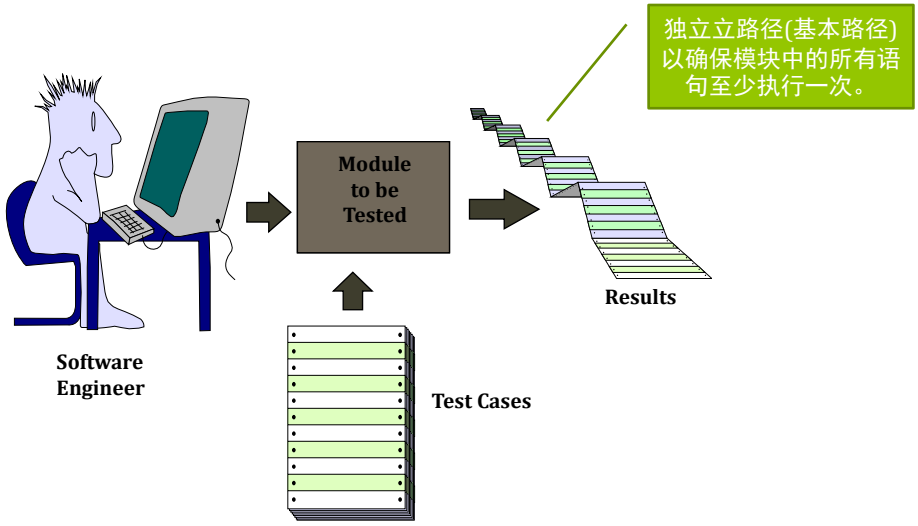


单元测试

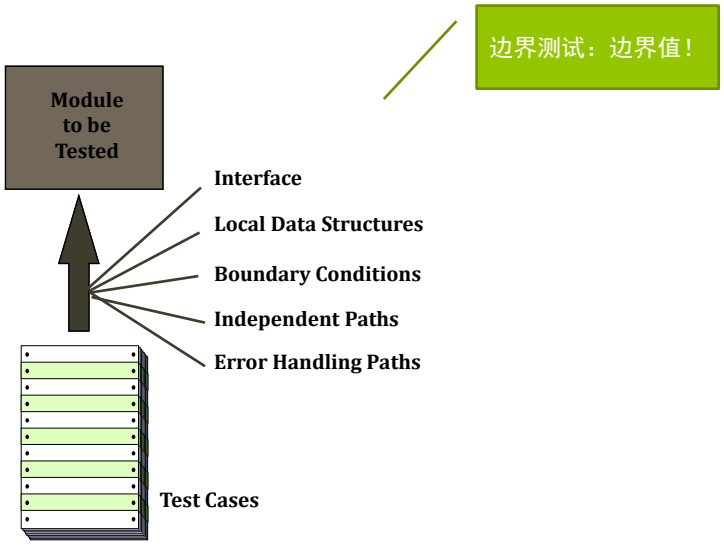
单元测试
Unit Testing

- 单元测试：侧重于软件设计的最小单元(软件构件或模块)的验证工作

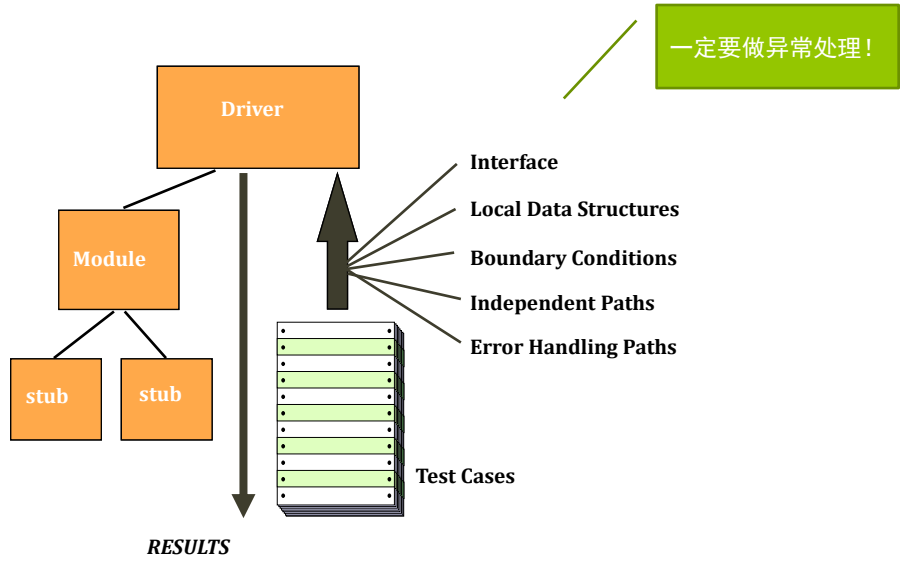
Unit Testing



Unit Testing



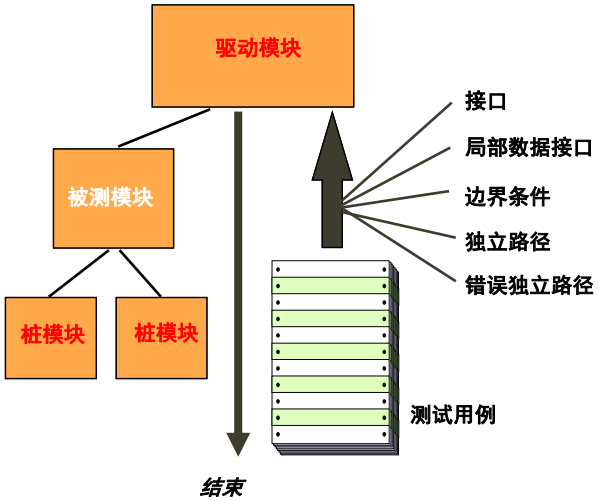
Unit Test Environment

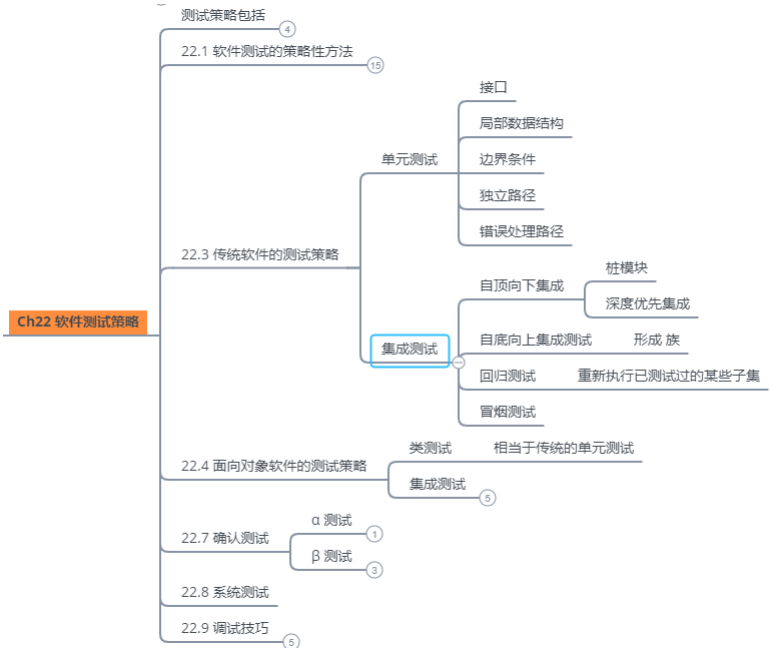


单元测试的环境

驱动模块——主程序

桩模块——从属代替部分





集成测试

Integration Testing

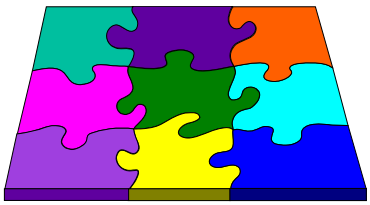
集成测试

集成测试
Integration Testing

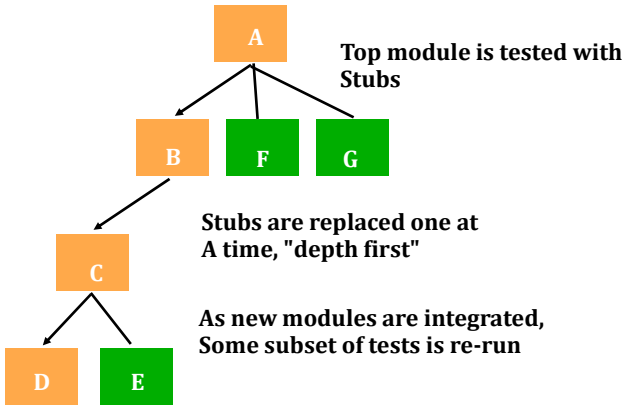
- Top Down Integration
- Bottom-Up Integration

集成测试 (Integrated Testing)

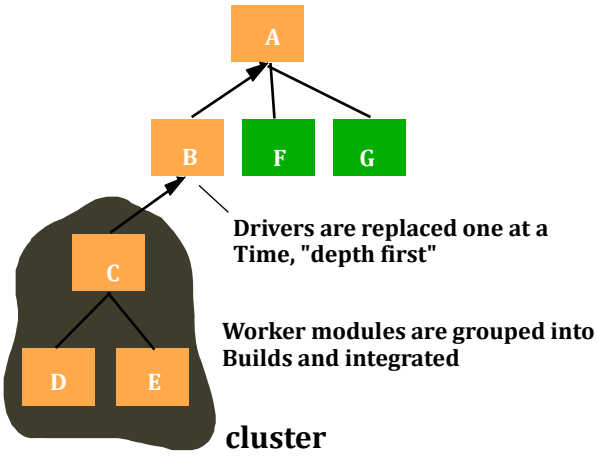
- 集成测试又称组装测试、综合测试。
  - 将所有模块按照设计要求组装成为系统。
- **需要考虑的问题：**
  - 在把各个模块连接起来的时侯，穿越模块接口的数据是否会丢失；
  - 一个模块的功能是否会对另一个模块的功能产生不利的影响；
  - 全局数据结构是否有问题；
  - 各个子功能组合起来，能否达到预期要求的父功能；
  - 单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。



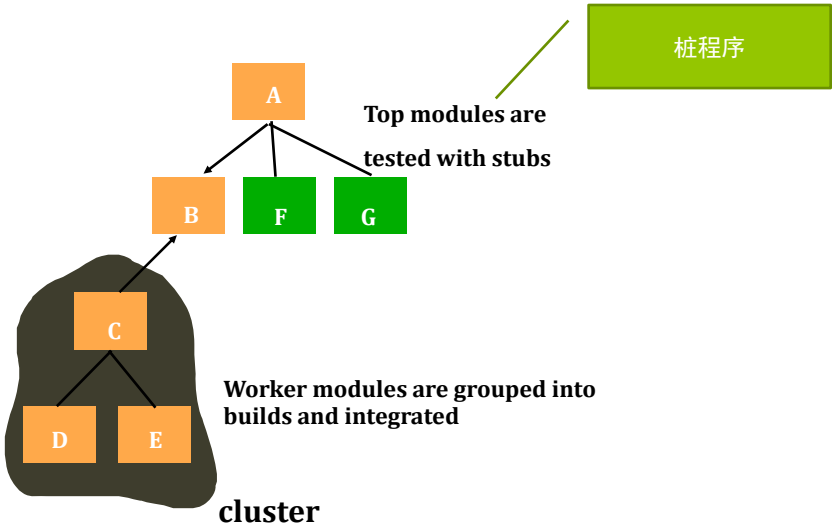
自顶向下集成测试 (Top Down Integration)



自底向上集成测试(Bottom-Up Integration)



Sandwich Testing



回归测试

回归测试
Regression Testing

- 回归测试是重新执行已测试过的某些子集，以确保变更没有传播不期望的副作用。
  - 重新(重复)执行测试

冒烟测试

冒烟测试
Smoke Testing

- 以成败论英雄！
  - 每天将该构建与其他构建及整个软件产品(以其当前的形式)集成起来进行冒烟测试。

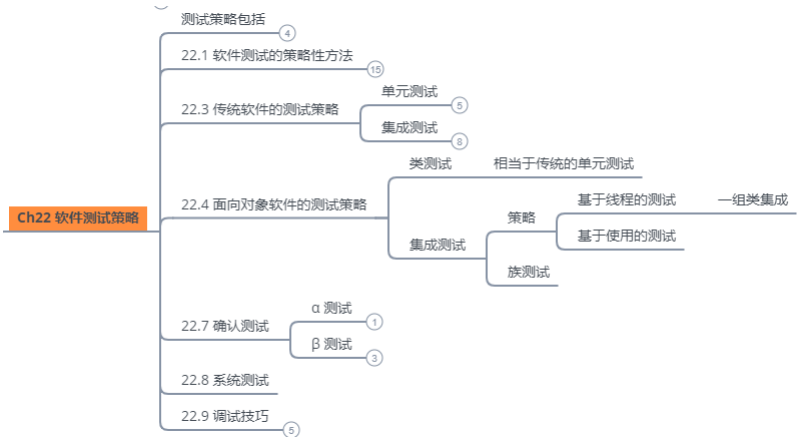
冒烟测试和回归测试的区别

- **冒烟测试**：自由测试的一种。在测试中发现问题，找到了一个Bug，然后开发人员会来修复这个Bug。
  - 优点：节省测试时间，防止build失败。
  - 缺点：是覆盖率还是比较低。

冒烟测试——自由测试

- **回归测试**：指修改了旧代码后，**重新**进行测试以确认修改没有引入新的错误或导致其他代码产生错误。
  - 工作量重，开发各阶段都进行多次回归测试。
    - 在渐进和快速迭代开发中，新版本的连续发布使回归测试进行的更加频繁，而在极端编程方法中，更是要求每天都进行若干次回归测试。
  - 自动回归测试将大幅降低系统测试、维护升级等阶段的成本。

自动回归测试—脚本编程



Unit

单元的概念发生了变化

- **Conventional Software**
  - The module (component) is our initial focus
  - Integration of modules follows
- **OO Software**
  - an OO class that encompasses attributes and operations and implies communication and collaboration

类测试等同于传统软件的单元测试

面向对象软件的类测试等同于传统软件的单元测试

面向对象测试策略

- 对象软件的本质特征:
  - 封装导出了类和对象的定义:
    - 实例包装有属性(数据)和处理这些数据的操作。
    - 超类定义某操作X，并且一些子类继承了操作X。
- 测试策略
- 测试战术

22.4.2 面向对象环境中的集成测试

- 基于线程的测试：**Thread-based Testing**
  - 对响应系统的一个输入或事件所需的一组类进行集成，每个线程单独地集成和测试。
- 基于使用的测试：**Use-based Testing**
  - 通过测试很少使用服务类(如果有的话)的那些类(称之为独立类)开始构造系统。
- 簇测试：**Cluster Testing**
  - 是面向对象软件集成测试中的一个步骤。

22.5 Validation Testing(确认测试)

- 确认测试又称可用性测试。

- 验证软件的功能和性能及其他特性是否与用户的要求一致。
  - 对软件的功能和性能要求在软件需求规格说明书中已经明确规定。
  - 它包含的信息就是软件确认测试的基础。

α测试 VS β测试

开发者在场　　开发者不在场

α测试和β测试

- 在软件交付使用后，用户将如何实际使用程序，对于开发者来说是**无法预测**的。
  - α测试是由一个用户**在开发环境下进行的测试**，也可以是公司内部的用户在模拟实际操作环境下进行的测试。
  - α测试的目的是评价软件产品的FURPS(即功能、可使用性、可靠性、性能和支持)。
    - 尤其注重产品的界面和特色。
  - α测试可以从软件产品编码结束之时开始，或在模块(子系统)测试完成之后开始，也可以在确认测试过程中产品达到一定的稳定和可靠程度之后再开始。

受控环境下进行！
开发者**在场**

α测试和β测试

- β测试是由多个用户在实际使用环境下进行的测试，用户返回有关错误信息给开发者。
  - 测试时开发者通常**不在测试现场**。
    - 因而，β测试是在开发者无法控制的环境下进行的软件现场应用。
  - 由用户记下遇到的所有问题，包括真实的以及主观认定的，定期向开发者报告。
  - β测试主要衡量产品的FURPS。
    - 着重于产品的支持性，包括文档、客户培训和支持产品生产能力。
  - 只有当α测试达到一定的可靠程度时，才能开始β测试。
  - 处在整个测试的最后阶段。
    - 产品的所有手册文本也应该在此阶段完全定稿。

受控环境下进行！
开发者**不在场**

β测试的一种变体称为客户验收测试。

有时是按照合同交付给客户时进行。

其他软件测试方法

收藏 421 276

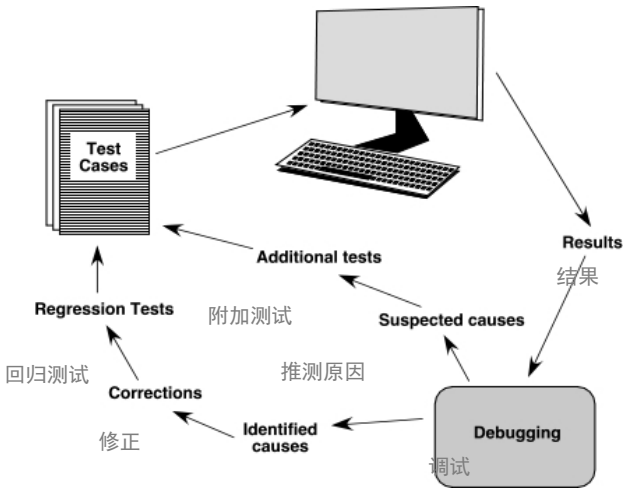
软件测试方法

软件测试方法是指测试软件性能的方法。随着软件测试技术的不断发展，测试方法也越来越多样化，针对性更强；选择合适的软件测试方法可以让我们事半功倍。

中文名	软件测试方法	所属行业	计算机
目 的	测试软件性能	作 用	选择合适的软件

目录

过程 (The Debugging Process)



有证据表明，调试本领属于一种个人天赋。

调试本领属于一种个人天赋。

证据表明，调试本领属于一种个人天赋。

Question?

Q&A?

College of Computer Science, Chongqing University

Software Engineering

A Practitioner's Approach Seventh Edition

22 软件测试策略

zmqmail@cqu.edu.cn 13708390417

重庆大学 计算机学院 COLLEGE OF COMPUTER SCIENCE