

# 软件质量保证与测试

Software Quality Assurance and Testing

## 第 5 章 集成测试

### 5.1 集成测试概述



金陵科技学院

# 集成测试简介

集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。

它是在单元测试的基础上，把多个经过单元测试的模块按照概要设计书组装起来进行测试，检查模块组合后，其功能、业务流程等是否实现或符合要求。

## 集成测试简介

也就是说，在集成测试之前，单元测试应该已经完成，集成测试中所使用的对象应该是已经经过单元测试的软件单元。这一点很重要，因为如果不经过单元测试，那么集成测试的效果将会受到很大影响，并且会大幅增加软件单元代码纠错的代价。

## 集成测试简介

一般这样定义集成测试：根据实际情况对程序模块采用适当的集成测试策略组装起来，对模块之间的接口以及集成后的功能等进行正确性检验的测试工作。集成测试能够发现单个模块测试时难以发现的问题，集成测试的主要依据是软件概要设计。

## 集成测试简介

所有的软件项目都不能摆脱系统集成这个阶段。不管采用什么开发模式，具体的开发工作总是要从一个一个的软件单元做起，软件单元只有经过集成才能形成一个有机的整体。具体的集成过程可能是显性的也可能是隐性的。只要有集成，总是会出现一些常见问题，工程实践中，几乎不存在软件单元组装过程中不出任何问题的情况。

## 集成测试简介

经验表明，作为软件测试的一个阶段，集成测试是不可或缺的，直接从单元测试过渡到系统测试是极不妥当的做法，而且集成测试需要花费的时间通常会要超过单元测试。

开始集成测试的时间，总体上说应该是在单元测试之后，但在实际中往往单元测试和集成可能有一部分工作同步进行，先做完单元测试的就可以先集成，以节约时间。集成测试主要是白盒测试和灰盒测试，所以一般由开发人员或白盒测试工程师来进行集成测试。

# 集成测试简介

集成测试的目的是确保各单元组合在一起后能够按既定要求协作运行，并确保新增加的模块、代码等行为正确。

## 集成测试策略和分析

集成测试的必要性在于，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常工作。程序在某些局部反映不出来的问题，有可能在全局上会暴露出来，影响功能的实现。此外，在某些开发模式，如迭代式开发中，设计和实现是迭代进行的。在这种情况下，集成测试的意义还在于它能间接地验证概要设计是否具有可行性。



# 1. 需要进行集成测试的原因

- ① 一个模块可能对另一个模块产生不利的影响
- ② 将子功能合成时不一定产生所期望的主功能
- ③ 独立可接受的误差，在多个模块组装后可能会被放大，超过可接受的误差限度。
- ④ 可能会发现单元测试中未发现的接口方面的错误
- ⑤ 在单元测试中无法发现时序问题（实时系统）

## 1. 需要进行集成测试的原因

- ⑥ 在单元测试中无法发现资源竞争问题
- ⑦ 共享数据或全局数据的问题
- ⑧ 数据单位、环境参数统一的问题

例如某收费软件系统中有两个模块单元，一个称重，一个计费，但称重模块中重量单位为克，而计费模块中把重量的数据单位默认为公斤，这样当称重结果数据传到计费模块后，费用的计算结果肯定是错误的。

## 2. 相依性分析

相依性是模块以各种方式相互联系和依赖的关系，常见的相依性关系如下：

- ① 合成和聚集
- ② 继承
- ③ 全局变量
- ④ 调用API
- ⑤ 服务器对象
- ⑥ 被用作消息参数的对象

# 集成测试策略和分析

集成测试的策略简介：

集成测试的实施策略从集成的次数上来划分，可以分为一次性集成和增量式集成。一次性集成也叫大爆炸式集成或非增量式集成。增量式集成测试又可以分为自底向上集成测试、自顶向下集成测试、三明治集成测试、核心系统优先集成测试、分层集成测试等。

应当从集成测试的具体需要出发，根据被测对象的特点，选择合适的集成测试策略。

# 集成测试简介

## 1. 一次性集成:

在对软件所有模块单元逐个进行单元测试后，采用一步到位的方法来构造测试，按程序结构图将各个模块单元全部联接起来，把联接后的程序当作一个整体来进行测试。

## 1.一次性集成:

例如某软件有A, B, C, D, E, F共6个单元模块, 结构如图5-1所示。

在6个单元模块都已经完成单元测试之后, 即可按照图5-1软件结构图, 把6个模块组装起来进行一次性集成测试。

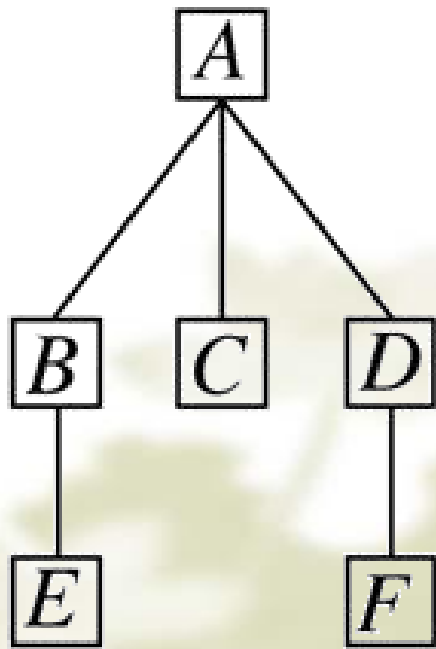


图5-1 软件结构图

# 1.一次性集成:

一次性集成的优点有:

- ① 集成次数少，能快速完成集成测试
- ② 所需测试用例较少，测试工作量较小
- ③ 不需要驱动模块和桩模块

一次性集成的缺点包括:

- ① 发现错误的时间较迟
- ② 错误较难定位
- ③ 测试不够不充分和彻底，即使通过测试，程序中可能还隐藏着一些错误
- ④ 测试的并行性较差

# 1.一次性集成:

一次性集成适用的情况主要是:

- ① 小的、良构的软件系统
- ② 一个已经存在的软件系统，只是做了少量的修改
- ③ 通过复用可信赖的构件构造的软件系统



## 2.增量式集成测试

按照某种关系，先把一部分模块组装起来进行测试，然后逐步扩大集成的范围，最后再把整个程序组装起来完成集成测试。

增量式集成测试又可以分为自底向上集成测试、自顶向下集成测试、三明治集成测试、核心系统优先集成测试、分层集成测试、基于使用的集成测试等。

## 2.增量式集成测试

与一次性集成相比，增量式集成的优缺点和适用情况分别如下。

优点：

- ① 集成测试可以较早的开始，测试的并行性较好
- ② 发现错误的时间较早
- ③ 错误定位较为容易
- ④ 测试比较不充分

## 2.增量式集成测试

缺点：

- ① 需要驱动模块和桩模块等辅助模块器
- ② 所需集成次数较多
- ③ 所需测试用例较多，测试工作量较大

适用情况：

- ① 增量式开发、框架式开发
- ② 并行软件开发
- ③ 较为复杂或者有一定规模的系统

### 3.两种集成策略的比较

非增量式集成和增量式集成各有其优缺点，它们的对比如表5-1。

表5-1

	非增式测试	增量式测试
集成次数	少，仅需1次	多
集成工作量	小	大
所需测试用例	少	多
驱动模块和桩模块	不需要	需要
发现错误的时间	较晚	早
错误定位	难	较容易
测试程度	不彻底	较为彻底
测试的并行性	差	较好
适用情况	结构良好的小型系统；原有系统做了少量的修改；通过复用可信赖的构件构造的软件系统。	增量式开发、框架式开发；并行软件开发；较为复杂或者有一定规模的系统。

# 集成测试分析

为做好集成测试工作，需要进行多种测试分析，包括软件结构分析、模块分析、接口分析等。

# 1. 软件结构分析

应先对软件结构进行分析。

1) 首先要明确系统的模块单元结构图，这是集成测试的基本依据。

2) 其次要对系统各个组件之间的依赖关系进行分析，然后据此确定集成测试的粒度，即集成模块的大小。

## 2. 模块分析

一般，可从以下几个角度进行模块分析：

- ① 确定本次要集成的模块；
- ② 明确这些模块的关系；
- ③ 明确本次集成所需要的所有桩模块和驱动模块，如果已有，则加进来，如果没有，则需要开发并加入到本次集成测试。



### 3. 接口分析

接口分析要以概要设计为基础，一般通过以下几个步骤来完成：

- ① 确定系统的边界、子系统的边界和模块的边界。
- ② 确定模块内部的接口。
- ③ 确定子系统内模块间接口。
- ④ 确定子系统间接口。
- ⑤ 确定系统与操作系统的接口。
- ⑥ 确定系统与硬件的接口。
- ⑦ 确定系统与第三方软件的接口。

必须尽可能早地分析接口的可测试性，提前为后续的测试工作做好准备。

本节内容就讲到这里，谢谢，再见！



金陵科技学院