

第6章 软件测试（下）

- ❖ 软件测试策略
- ❖ 调试
- ❖ 软件测试报告

软件测试策略

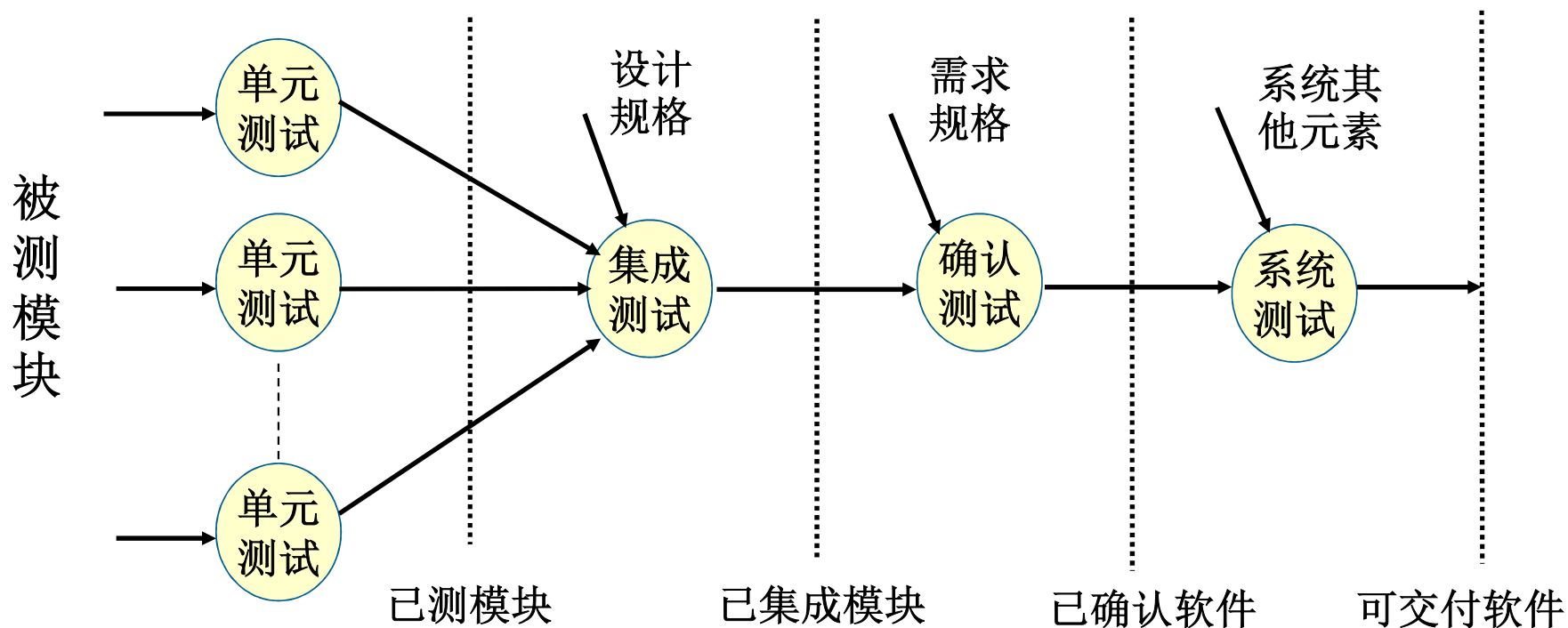
软件测试阶段完成的主要任务有两类：

- 一类是局部模块的测试，它是整个测试阶段的基石。
- 另一类是软件系统全局结构的测试，它构成整个测试系统的大厦。
- 从局部到全局，是经过一系列测试过程转换而成，它们包括单元测试、集成测试、确认测试和系统测试，每个测试过程都有各自不同的测试策略。

软件测试策略

测试步骤及策略

所有测试过程都应采用综合测试策略；即先作静态分析，制订测试计划，再进行动态测试。

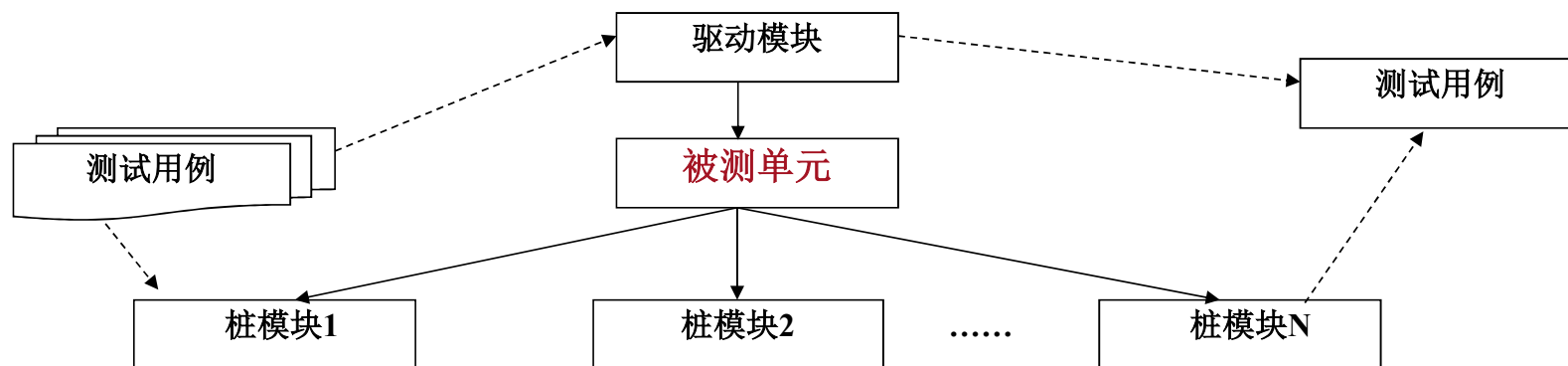


软件测试策略

1. 单元测试——测试策略

单元是软件测试过程中最小的测试单位，它有两个基本属性：

- 单元是可以独立编译的最小组件（函数、接口、类）；
- 单元是由个人开发的软件组件（菜单、界面）。



软件测试策略

1. 单元测试——MOCK

Mock测试是在测试过程中对于某些不容易构造，或者不易获取的对象，用一个虚拟对象（即**Mock**对象）来创建，以便进行测试的方法。

- ◆ 真实对象有难以确定的行为，会产生不可预测的结果；
- ◆ 真实对象很难被创建，如具体的**Web**容器；
- ◆ 真实对象的某些行为很难触发，如网络错误；
- ◆ 真实情况令程序的运行速度很慢；
- ◆ 真实对象有用户界面；

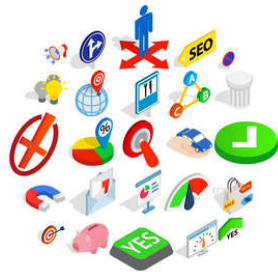
软件测试策略

2. 集成测试——测试策略

- ◆ 集成测试的重点，是将各模块按照软件系统结构的定义，将软件模块组装在一起。
- ◆ 如何实施组装过程，使得系统既能快速集成，又能准确发现在组装过程中出现的错误和问题呢？

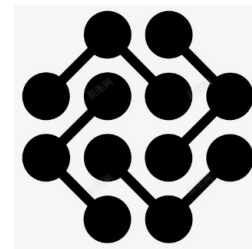
(1) 非渐增式集成

非渐增式集成是一次集成过程，即先按照各模块在系统结构中的位置，设计驱动模块或桩模块进行辅助测试。然后将测试合格的各模块按照系统结构一次性完成系统集成的过程。

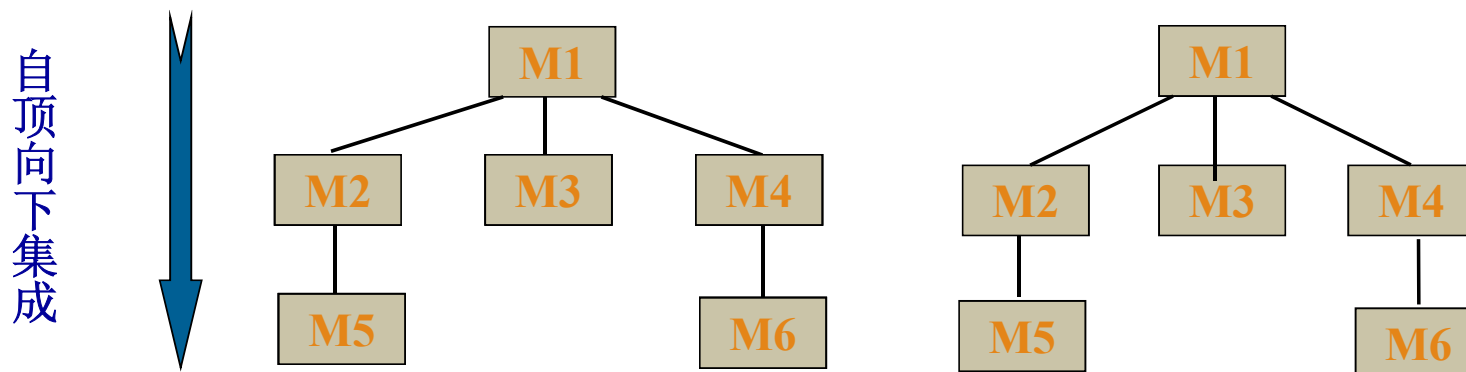


(2) 渐增式集成

渐增式集成不同于非渐增式集成，它不是将模块一次性组装，而是按照系统结构逐渐将模块依次集成到系统中。从集成的过程来看，渐增式集成过程分为**自底向上**的集成和**自顶向下**的集成。



软件测试策略



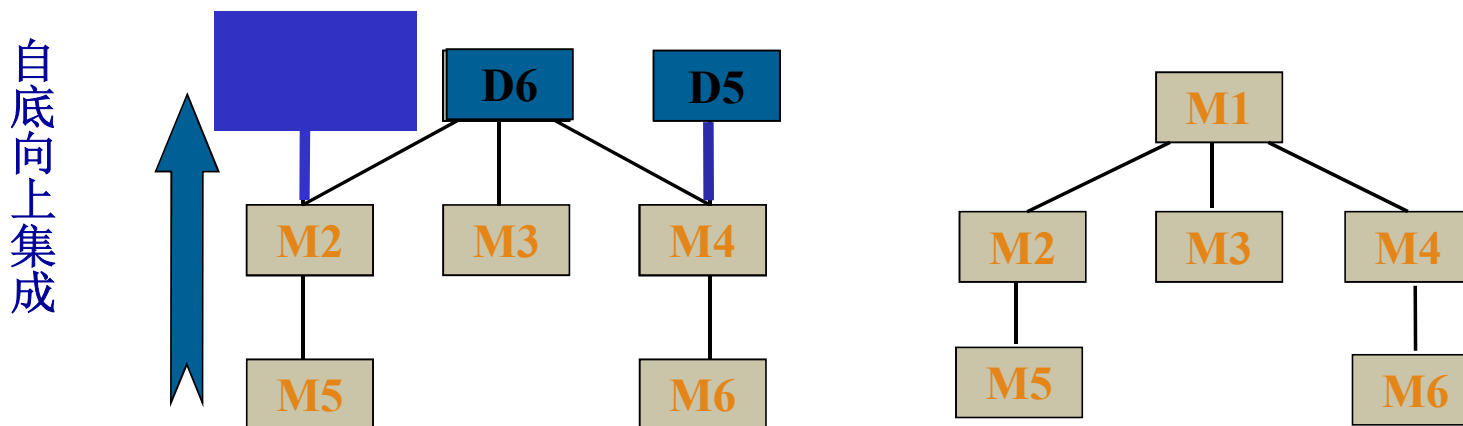
第一步，测试主控模块M1，设计桩模块S1、S2、S3，模拟被M1调用的M2、M3、M4。

第二步，依次用M2、M3、M4替代桩模块S1、S2、S3，每替代一次进行一次测试。

第三步，对由主控模块M1和模块M2、M3、M4构成的子系统进行测试，设计桩模块S4、S5。

第四步，依次用模块M5和M6替代桩模块S4、S5，并同时进行新的测试。组装测试完毕。

软件测试策略



第一步，对最底层的模块M3、M5、M6进行测试,设计驱动模块D1、D2、D3来模拟调用。

第二步，用实际模块M2、M1和M4替换驱动模块D1、D2、D3。

第三步，设计驱动模块D4、D5和D6模拟调用，分别对新子系统进行测试。

第四步，把已测试的子系统按程序结构连接起来完成程序整体的组装测试。

软件测试策略

● 自顶而下增值

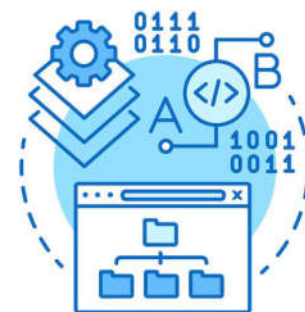
优点：能够尽早发现系统主控方面的问题。

缺点：无法验证桩模块是否完全模拟了下属模块的功能。

● 自底而上增值

优点：能够尽早查出底层涉及较复杂的算法和实际的I/O模块中的错误。

缺点：最后才能发现系统主控方面的问题。



Integration Testing

● 集成过程的原则

① 尽早测试关键模块。

② 尽早测试包含I/O的模块。

- 公共模块
- 关键路径上的模块
- 影响系统性能的模块

- 访问文件/数据库
- 与外部系统的数据交换
- 数据采集

软件测试策略

3. 确认测试

确认测试也称为验收测试，它是指在模拟用户实际操作的环境下（或开发环境下），运用黑盒测试法验证软件的有效性是否符合需求。

回想在需求获取时：

- 用户的操作场景；
- 用户的操作习惯；
- 用户的操作流程；
-

确认测试完成后，可能产生两种情形：

- (1) 软件系统功能、性能、领域等要素满足**用户需求规格说明**，软件是有效的。
- (2) 如果软件系统功能、性能、领域等要素存在不满足**用户需求规格说明**的某些方面，通过确认测试指出系统存在的错误，并对比预期结果给出测试分析报告。

软件测试策略

4. 系统测试

系统测试是指软件系统作为整个计算机系统的一部分，与计算机系统的硬件系统、数据、外部其它软件、文档等要素相结合，在用户实际运行环境中进行的确认测试。

- 系统测试的内容：
- 功能测试
 - 性能测试
 - 压力测试
 - 容量测试
 - 安全测试
 - 文档测试
 - 恢复性测试
 - 备份测试

软件测试策略

系统测试——压力测试

压力测试也成为强度测试，是指系统在各种资源超负荷得情况下对运行状况进行测试。

- 压力测试的基本思路：计算机系统资源匮乏或有限的条件下运行的测试，是测试系统在使用峰值时的性能。
- 压力测试的主要对象包括：内存、外部存储、网络负载、中断处理、缓冲区、事务队列、在线用户量等。

软件测试策略

系统测试——压力测试：QPS / TPS

- **QPS(Queries Per Second):** 每秒查询率，它是指一台服务器每秒能够响应的查询次数，是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准。
- **TPS(Transactions Per Second):** 每秒事务数。事务是指客户机向服务器发送请求到服务器做返回请求结果过程。



TPS即每秒处理事务数，每个事务包括以下三个过程：

- 用户请求服务器
- 服务器自己的内部处理
- 服务器返回给用户

QPS与TPS的区别：对于一个页面的一次访问，形成一个TPS；但一次页面请求，可能产生多次对服务器的请求，服务器对这些请求，就可计入“QPS”之中。

【例】访问淘宝商品的某个页面会请求服务器3次：基本信息、评价、直播。这样，一次商品页面的访问，产生一个TSP请求，产生3个QPS请求。

软件测试策略

系统测试——压力测试：PV模型（前端/后端）

- **PV（page view）**是指服务器页面的访问次数。当客户端打开一个网页或刷新一次网页，就计算一次**PV**。

每台服务器每秒处理请求的数量 =

$$((\text{总PV} * 80\%) / (24\text{小时} * 60\text{分} * 60\text{秒} * 40\%)) / \text{服务器数量}$$

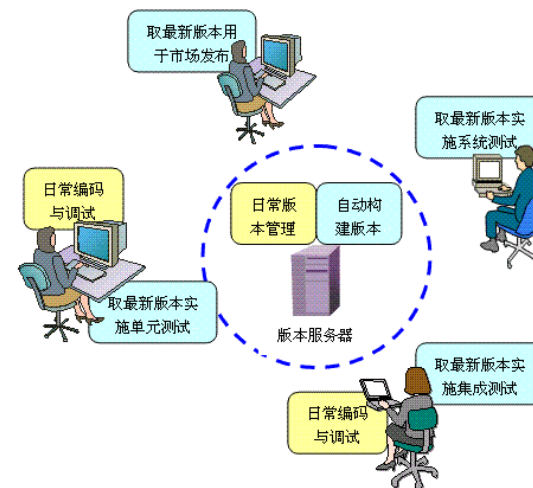
含义：表示在一天中，**80%**的请求发生**40%**的时间内。**24小时**的**40%**是**9.6小时**，有**80%**的请求发生一天的**9.6**个小时当中（可以理解为：白天请求多，晚上请求少）。

提问：如何建设一个网站，使之能分别支持**10万**、**100万**、**5000万**的**PV**访问量？

软件测试策略

测试策略——持续集成

持续集成是一种软件开发实践，团队成员频繁地集成各自工作，通常要求每人每天至少集成一次。经验表明，持续集成能显著地减少集成错误并快速整合软件。

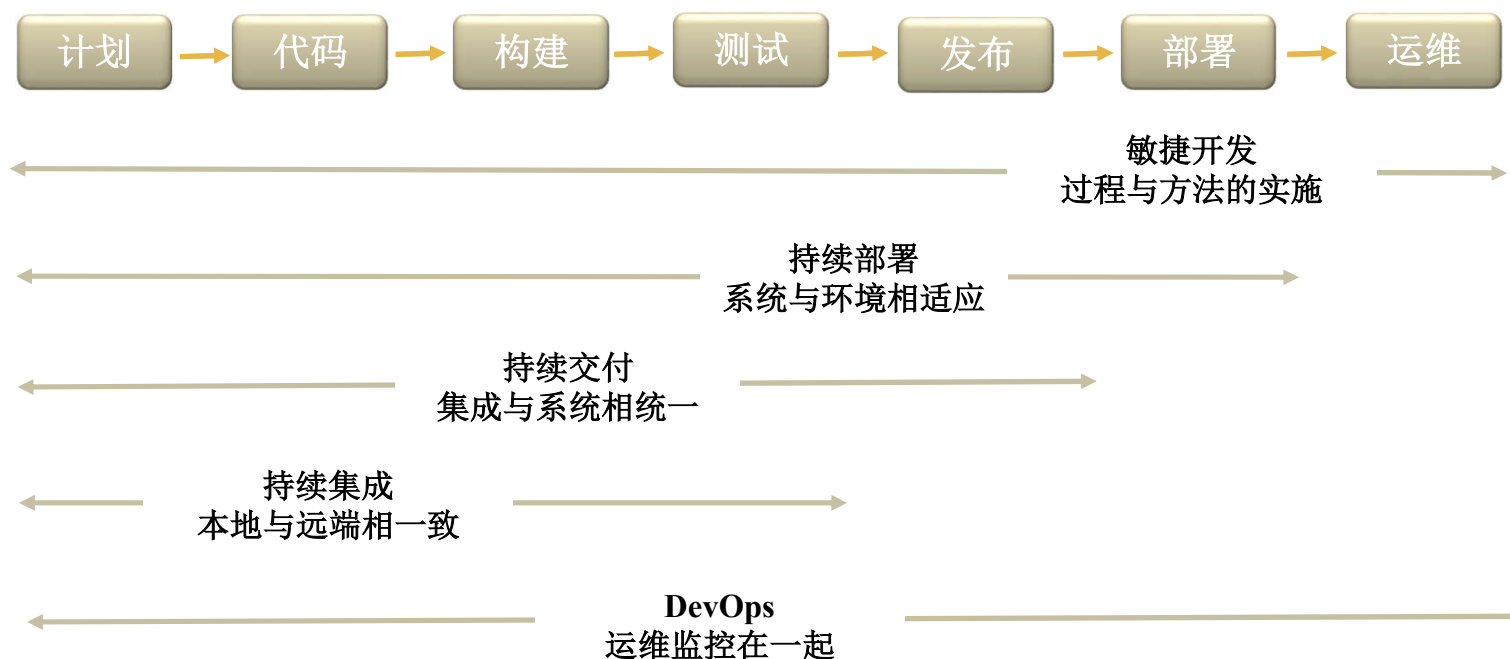


- 频繁发布：持续集成的目的就是通过高频度提交集成，满足软件系统频繁交付满足需求的软件。
- 自动化过程：无论软件系统发布定义为何种频度，都需要自动化过程来构建提交、集成、测试、发布、部署等进行处理。
- 可重复：对于软件系统来说，定义的自动化过程对于相同输入都应得到对应的输出结果。

软件测试策略

测试策略——持续集成的基本结构

- 在持续集成实践中，“集成”包括源代码提交及编译、测试、系统打包、发布等一系列过程。通过每次集成的自动化构建（包括编译，发布，自动化测试）来验证，力求尽快、尽早地发现集成错误。



软件测试策略

测试步骤及策略比较

测试类型	目的	测试依据	测试方法
单元测试	发现模块内部逻辑和功能上的错误或缺陷，以及接口存在的问题	详细设计规格说明	白盒测试
集成测试	发现模块间调用关系，以及模块间接口方面问题	概要设计规格说明	以黑盒测试为主，结合使用白盒测试
验收测试	测试系统软件整体功能和性能，进行一系列整体的、有效性测试	需求规格说明	黑盒测试
系统测试	结合网络、服务器等硬件，及支持软件等综合环境进行测试	需求规格说明	黑盒测试

调试

软件测试的目的是发现错误，测试完成之后的软件调试目的是为了定位和修改错误，以保证软件运行的正确性和可靠性。

软件调试活动主要分为以下三部分内容：

(1) **确定软件系统出现错误的准确位置**，这需要结合调试人员的经验和技巧。

(2) **对发现错误的修改**，如分析修改的程序是否涉及全局数据、是否涉及面向对象机制（如虚函数）、是否影响修改函数的主调函数结果的正确性等。

(3) **对修改后的内容重新进行测试**，特别是涉及到全局数据结构、文件结构、系统结构等内容的修改，还必须进行确认测试和系统测试。

最后，修改对应的文档内容，并通知与之对应的人员。



软件测试报告

测试报告文档

在软件测试各阶段完成之前，必须编写软件测试报告，并按照评审标准对软件测试报告进行评审。编写测试报告的目的是发现并消除其中存在的遗漏、错误和不足，使得测试用例、测试预期结果等内容符合标注及规范的要求。通过了评审的软件测试报告成为基线配置项，纳入项目管理的过程。



软件测试报告主要包括**软件测试说明**和**软件测试报告**两个部分。

软件测试常见问题

与软件测试有关的几个基本问题

- ❖ **问题1:** 由于单元测试要编写测试驱动程序，非常麻烦，能否等到整个系统全部开发完成后，再一次性的统一进行单元测试？
- ❖ **问题2:** 如果完成单元测试后，将所有单元集成在一起组装成系统有何不妥？集成测试是否多此一举？
- ❖ **问题3:** 在集成测试时，对于已经进行了功能测试、性能测试的子系统，在系统测试时是否能跳过相同内容的测试？
- ❖ **问题4:** 能否将系统测试和确认测试“合二为一”？