

# 单元测试—测试驱动开发实践

HH@BJUG, 2005-3

# 你看到这些问题了吗

## ◆ 开发就像和泥巴

- 有很多种方法堵住一个漏洞，可只有一种方法能釜底抽薪

## ◆ 遭遇规模瓶颈

- 3~5万行代码，10来条枪，出错的速度与改错的速度持平

# 观察与思考

## ◆ 建筑与建筑工人

- 你不会没见过这些人，衣衫褴褛，文化不高，却盖起高楼大厦

## ◆ 电子装配工厂

- 普通女工，做的都是高科技

## ◆ 软件业：为什么这么多菁英，却满足不了客户的基本要求？

# 重新认识测试

- ◆单元测试就是设计
- ◆管理上进行约束的一种手段
- ◆以终为始的体现
- ◆最好的、最精确的沟通语言

# 单元测试就是设计

- ◆ 什么是设计？UML图，概要、详细文档，编码，测试，这些都是设计
- ◆ 没有测试，就没有设计
- ◆ 难以测试，就是拙劣的设计

# Unit Testing Is Design

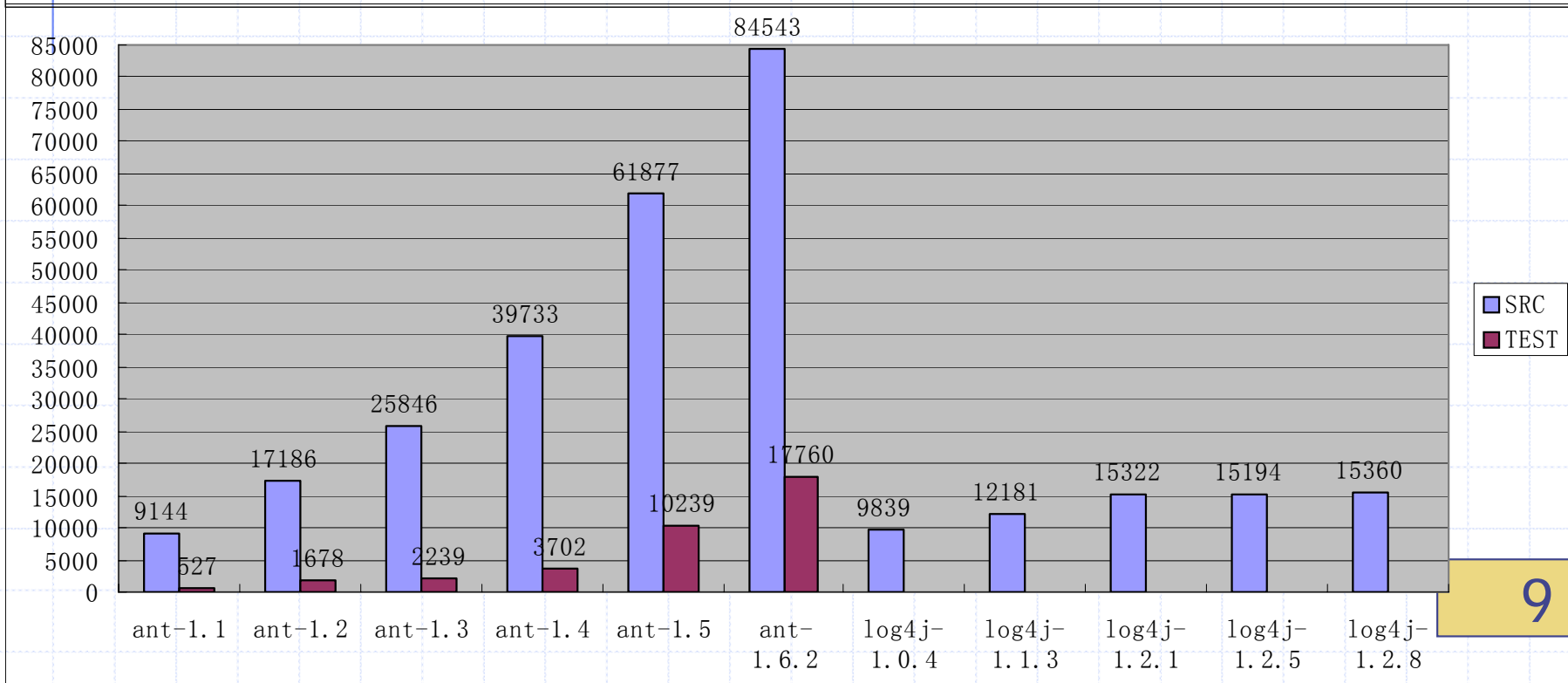
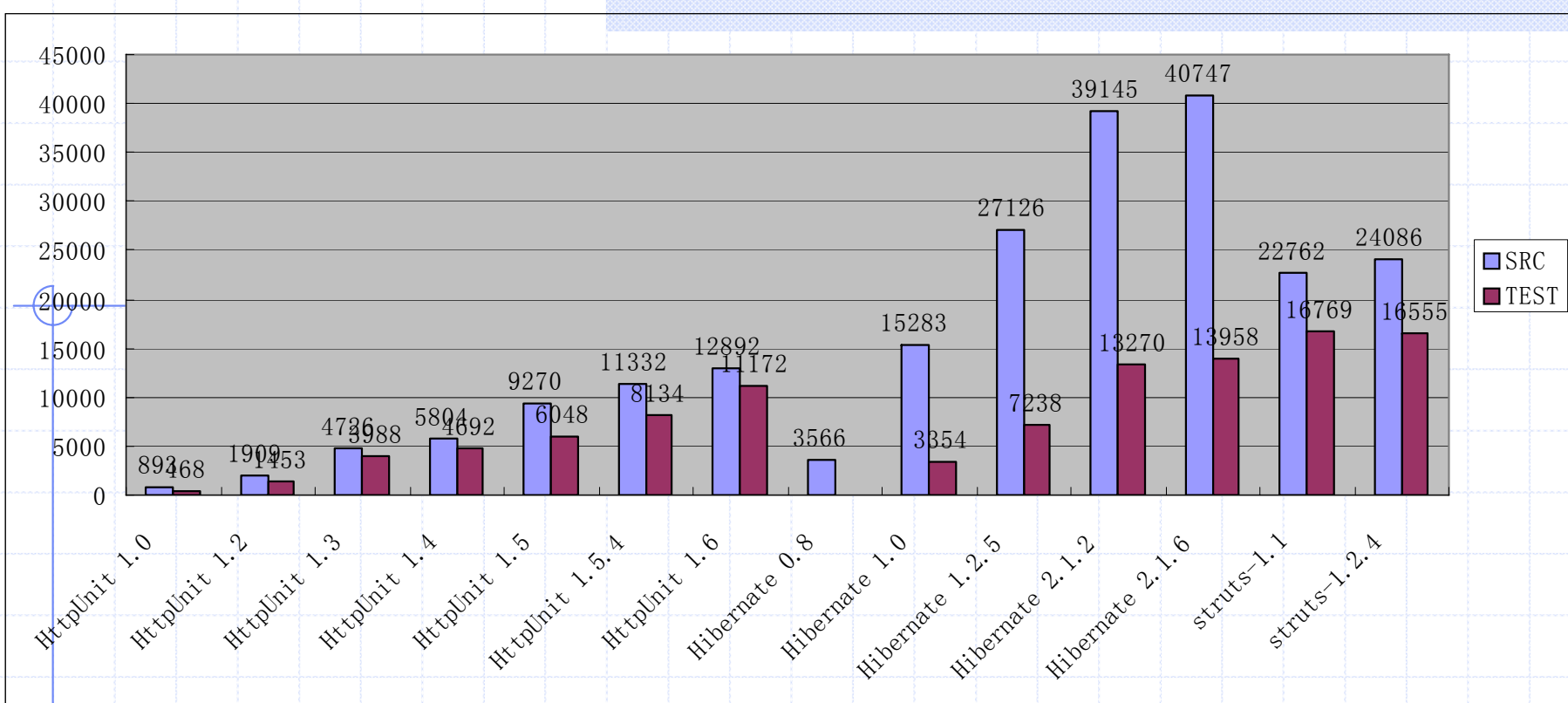
- ◆ People have noticed that XP doesn't talk about designing. Here's what I noticed the other day. When I write UnitTests, I am doing some of my most sophisticated design. But I don't do it in the pursuit of truth, beauty, or even money. I do it because I am lazy.  
(c2.com)

# 测试不仅仅是测试部的事

- ◆ 对管理者，测试可以成为约束手段
  - 测试和实现，是软件开发对立的两方面，通过分离测试和实现代码，可以形成对代码修改的制约
- ◆ 对程序员，以终为始（高效能人士的七个习惯之一）：测试通过就是编程的终点，多做些测试反而能缩短时间，提高效率
- ◆ 对程序员，单元测试是沟通的语言

# 一些数据

|                 | SRC   | TEST  | RATIO |
|-----------------|-------|-------|-------|
| HttpUnit 1.0    | 893   | 468   | 1.91  |
| HttpUnit 1.2    | 1909  | 1453  | 1.31  |
| HttpUnit 1.3    | 4726  | 3988  | 1.19  |
| HttpUnit 1.4    | 5804  | 4692  | 1.24  |
| HttpUnit 1.5    | 9270  | 6048  | 1.53  |
| HttpUnit 1.5.4  | 11332 | 8134  | 1.39  |
| HttpUnit 1.6    | 12892 | 11172 | 1.15  |
| Hibernate 0.8   | 3566  |       |       |
| Hibernate 1.0   | 15283 | 3354  | 4.56  |
| Hibernate 1.2.5 | 27126 | 7238  | 3.75  |
| Hibernate 2.1.2 | 39145 | 13270 | 2.95  |
| Hibernate 2.1.6 | 40747 | 13958 | 2.92  |
| struts-1.1      | 22762 | 16769 | 1.36  |
| struts-1.2.4    | 24086 | 16555 | 1.45  |
| ant-1.1         | 9144  | 527   | 17.35 |
| ant-1.2         | 17186 | 1678  | 10.24 |
| ant-1.3         | 25846 | 2239  | 11.54 |
| ant-1.4         | 39733 | 3702  | 10.73 |
| ant-1.5         | 61877 | 10239 | 6.04  |
| ant-1.6.2       | 84543 | 17760 | 4.76  |
| log4j-1.0.4     | 9839  |       |       |
| log4j-1.1.3     | 12181 |       |       |
| log4j-1.2.1     | 15322 |       |       |
| log4j-1.2.5     | 15194 |       |       |
| log4j-1.2.8     | 15360 |       |       |



# 单元测试要点

- ◆ 每个测试用例独立运行(new, setUp, test, tearDown), 测试用例之间尽可能不相互依赖
- ◆ 编写步骤: 结果(assert)——前提(setUp)——操作(testABC)
- ◆ 以条件组织用例, 而不是一个类一个测试

# 单元测试模式

- ◆ 失败，就败得惨烈些(assert, 抛出所有异常)
- ◆ 日志(log)
- ◆ 模拟对象，模拟器等(手编模拟对象或通用模拟器)

# 单元测试工具

- ◆ JUnit
- ◆ HttpUnit/ServletUnit
- ◆ Cactus
- ◆ JUnitPerf
- ◆ MockObjects/MockMaker/EasyMock
- ◆ StrutsUnit, Jester, NoUnit, Clover, AOP, ...

# 测试驱动开发要点

- ◆先写测试，再写具体实现
- ◆重构代码，也重构测试（并且先重构测试）
- ◆精益思想：从简单的测试开始，一次完成一个测试，不要过度设计
- ◆先维护测试，再维护代码
- ◆代码:测试=1:1

# 单元测试的位置

- ◆比测试更重要的事：设计、沟通、情感……(还有哪些?)
- ◆没测试重要的事：编码规范、工作日志……(还有哪些?)
- ◆哪些情况下单元测试不适用?
- ◆提高软件工程水平路线图：版本控制——单元测试——(下面是什么?)

# 骨骼与血肉

- ◆ 软件不应该是一个泥人，要有骨骼、血肉
- ◆ 通过接口、前置条件、后置条件等结构上的和语义上的契约塑造软件的骨骼
- ◆ 通过单元测试定义逻辑契约（骨骼）
- ◆ 骨骼使人强壮，血肉使人生动，骨骼是稳定的，容貌(GUI)却是千变万化、永无止境的

# 附录一：ServletUnit 简介

- 1) 首先，你需要创建一个Servlet运行器（它模拟了一个容器），并且注册你的 Servlet:
- 2) 然后，你需要创建一个单元测试客户对象，它相当于一个浏览器
- 3) 你告诉它需要浏览的网页地址，这要用到一个Get方式请求或Post方式请求对象:
- 4) 最后，你就可以通过访问上面指定的网页来调用你的 Servlet 了:
- 5) 要查看 Servlet 输出的内容，使用一个 WebResponse 对象:

```
WebResponse webResponse = invocation.getServletResponse();  
System.out.print(webResponse.getText());
```

# 对象清单及其使用次数

|                      |             |              |
|----------------------|-------------|--------------|
| ServletRunner        | Web容器       | 一组测试一个       |
| ServletUnitClient    | 浏览器         | 一个测试用例<br>一个 |
| WebRequest(Get/Post) | URL地址       | 一次请求一个       |
| InvocationContext    | Servlet运行环境 | 一次请求一个       |
| WebResponse          | 返回内容        | 一次请求一个       |

# 参考

- ◆ <http://c2.com/cgi/wiki?UnitTestingIsDesign>
- ◆ 活用 XP: (五)测试管理 林星
- ◆ 《测试驱动开发》，Kent Beck
- ◆ 《测试驱动开发实用指南》，David Astels