

### 对原有体系结构/详细设计协作 中 集成测试的回顾：

经过了一个月的代码构造，首先我们发现在真正码代码解决问题的时候，跟之前没有动工仅仅写文档是有十分巨大的相差的。特别是详细设计，码代码时候发现之前所想的协作关系过于简单，例如：以订单执行为中心，需要调用credit、room、hotel和user，而这些详细设计并没有考虑到。对于集成测试，文档中所写的stub、mockobject和po也有十分大的修改，难免有很多接口改动，或者调用关系改动，或者po数据添加减少。但是从一开始思路我觉得是可以的，在码代码也有效的避免了一些bug的出现：当不同的人开发不同的层时，用stub隔开，并且用stub进行测试；同层开发时，对于没有完成的包，使用mockobject替换。最后测试时，一个个mock替换，再一个个替换stub。方便测试。但对于集成测试，不得不说我们没有很好的利用好jenkins，（事实上我们也不是特别理解jenkins及使用它的好处）环境的配置就花了我们很久时间去解决。

### 对单元测试/测试驱动 中 单元测试的回顾：

在需求分析时，我们写了测试用例套件。但事实证明，它在后面的构造阶段作用十分十分渺小。（当然很有可能我们测试用例套件写的不好），测试用例基本都是在写某个方法之前才写。但不得不说，测试用例十分有用，帮助发现了许多bug。以logic层为例，logic一共写了100+个测试用例，帮助发现了很多手写代码的“笔误”，以及一些错误的逻辑关系。目前对单元测试最大感想是，测试用例一定要在方法写之前写，最好能够让别人写测试用例，这样效果才最好。

### 回顾包括对原有测试用例有效性的分析和采取的弥补手段

原有的测试用例适用性很差，究其原因，是构造阶段对方法的确定与需求分析阶段确定的方法相差很大，所以设计的测试用例套件也没什么用处。所以我们很疑惑，在实际项目中是如何实现在需求分析阶段就可以设计出测试用例套件的.....