互联网酒店预订系统

测试回顾

南京大学

软件工程与计算二

**MutualBeats团队测试回顾总结会议记录**

## 一、会议时间：2016/12/18

## 二、会议地点：MusicBar

## 三、与会人员：MutualBeats团队全体成员

## 四、会议内容：对互联网酒店预订系统的测试回顾

### 1、测试情况

#### 1.1系统测试，功能测试

详情见《互联网酒店预订系统测试总结报告》

有效性分析：主要使用手工测试，黑盒测试，辅助测试和白盒测试，比较有效。

## 五、总结：对互联网酒店预订系统的测试回顾

### 1.教训

#### 1.1任务分工不科学

描述：初期忽略了展示层，逻辑层，数据层三层之间的工作量差异，导致分工不均匀。

弥补手段：合理分工，最好按照功能分工，每一位成员尽量工作量相差不大。

#### 1.2mock，stub存储位置错误

描述：没有正确理解mock和stub的意义，导致在实际写bl和controller的时候删除了mock和stub，在之后的集成测试中，出现了大量的错误。

弥补手段：将mock和stub单独放置，便于统一修改。

#### 1.3controller接口写的过大

描述：没有详细分开各个接口的职责，导致测试过少，实际方法太过庞大，代码显得非常臃肿，重复代码偏多。

弥补手段：在需求和设计阶段详细定义各个接口，接口的定义要经过小组全体的讨论，之后的工作可以顺利进行。

#### 1.4vo没有事先定好

描述：vo开始依据po定义，导致部分方法获得数据大大超出所需数据，造成资源浪费；部分方法传出数据小于调用者所需数据，出现错误。

弥补手段：vo和po的定义要是想商量好，对于修改，尽量不要涉及po。vo要有无参数的构造方法。构建vo时秉持要什么数据传什么数据的原则。

### 2.经验

#### 2.1使用了Github优秀的版本控制工具

成员可以方便提交和拉取代码，为合作测试提供了便利。优秀的冲突检测让成员间合作高效。

#### 2.2 增加controller控制ui层调用bl的方法

ui和bl之间设立严格的controller方法，控制ui可以调用的方法，测试也能够方便获取需要测试的方法。

#### 2.3 增加讨论次数

成员之间交流频繁，极少出现同步冲突。