# 概述

互联网酒店预定系统的开发历时三个多月的时间，期间有的地方做的不错，譬如高效的沟通和很好的模块设计，但也有需要改进的方面，这些都为以后的团队合作提供了参考经验。

# 项目评审

1. **项目所使用的过程是什么？**

按照课程要求，使用瀑布模型，分层的方式。采用自底向上方法，用驱动的方式开发。

1. **实际的过程和原先确定的过程有什么不同？**

实际上的过程基本是按照教材的过程，没有什么大的变化。

1. **进度表是如何随着时间变化和改变的？**

按照老师给的deadline一个一个完成。

1. **有多少个同步点和里程碑按时达到或错失？**

小组活动基本上是按照deadline的要求执行的，肯定都是在deadline要求的时间之前完成，与老师的课程要求保持了同步。由于在体系结构设计阶段和详细设计阶段耗费了大量的时间并且返工多次，不过还是基本上跟上deadline。可是项目后期证明在设计环节花费的时间是值得的，写代码过程以及构造阶段都进行的较为顺利，因此里程碑基本都按时达到了，项目基本能够按期完成。项目最后打包方面由于对maven的使用及插件以来等诸多问题不够熟悉，一直无法实现打包，该同步点最初并未按时实现，不过最后还是赶在deadline之前完成了。

1. **过程的哪些部分运行得好？**

在整个项目过程中，整个小组都在摸着石头过河。从需求阶段开始，小组就以高昂的热情来面对，保证了小组每人对项目的激情和投入时间，整体气氛氛围很好。

在项目构造到中后期的时候，小组成员一边进行继续的构造工作，一般进行原有代码的重构工作。包括order、hotel、user等重要部分都进行了重构，让项目具有更好的可复用性和可修改行，项目的逻辑也更加清晰明了。

小组成员因为都来自一个班级，比较容易找到共同的空余时间开会，所以小组全程交流比较多，每个人都对项目的进度有着比较好的把握。

1. **过程的哪些部分应该运行得更好？**

在需求阶段对客户需求有不太明白的问题，因此体系结构设计的时候又花费了较多时间讨论具体需求的实现；在体系结构阶段写的桩和驱动，以及在详细设计阶段写的mock object考虑的不够成熟，无法保证后期能够有效的使用，导致后期测试阶段使用率不高；在人机交互阶段没有直接做好界面，而直是画了一个草图，导致后期构造阶段还在改界面，应该当时就直接画好确定下来；界面重复代码较多，在代码复用上应该可以做的更好；数据层较为复杂，导致逻辑层的代码较少，系统风格和既定的分层风格有些偏差；集成测试部分由于分工方式是按照模块分工，小组成员之间并没有太多的交叉部分，导致每个人在前期构造阶段只要保证自己的模块可以从界面到数据层的正常运行就可以了，整个系统的集成是到所有人的功能都基本实现的时候才进行的。

1. **工具支持这个过程吗？**

主要使用了Github，没有使用重量级的过程工具，持续集成的一系列工具可以正常使用，度量的一些轻量级工具也可以工作。

1. **从整体上讲，这个过程运行得有效吗？**

有效，一定程度上管理了系统的开发，提高了团队的运行开发效率，增加了团队的凝聚力。

1. **在今后，尤其要对哪些方面进行改进？**

需要提高各个方面的能力，如可以将服务器和客户端通过远程网络分到两个电脑上操作，用web模式开发界面，都可以提高软件的开发质量，还有测试部分做的不是很好，应该详细的写出各个部分的单元测试，提高测试质量，应该多个人部署一些工具，如有个人电脑崩了之后集成测试就停止了，必须重新开始。

1. **在每个阶段的每项任务上花费的时间是多少？**

需求开发：2016.9.18-2015.10.1

软件设计：2016.10.2-2016.11.13（体系结构设计和人机交互设计）

软件实现：2016.11.14- 2016.12.20

软件单元测试：2015.11.14-2017.1.1

软件系统测试：2015.12.16-2017.1.1

1. **在项目的上面周期中，产品是如何变化的？**

先进行需求的分析，这方面统一意见做的比较好。产品开发由计划进行，我们分层开发，先写出所有逻辑层，这里承担着整个项目的主要内容，通过stub和driver进行测试，这方面花费了很多的时间。在体系规格设计和详细设计之后，我们基于logic层，同时进行presentation和data层的设计，然后在各个层之间的控制出现了一些问题，但在邮件轰炸之下，我们逐一解决，几乎同时开发出data层和presentation层，但是在这样在集成测试的时候就有很多问题出现，不停的改接口，返回详细设计。同时进行着单元测试。但是界面方面比较难受，fxml的不能传入数值的方面有问题，最后形成了一个特别大的controller类，至今没有解决这个问题。总之产品进行的比较顺利。

1. **有没有出现重要的产品返工的情况？如果有，在什么时候？**

有。在构造阶段，需要不停的改变原来的接口，发现原来的接口并不能满足程序的实现，还是在设计的时候没有考虑清楚。

1. **工具支持产品的制造、维护和测量么？**

支持测试、度量和集成构建。

1. **产品最后的规模有多大？**

LOC：9451（client）+3790（server）+2948（common）=16189

1. **产品的质量如何？**

产品的质量总体来说达到了预期的标准，各个方面都做了相应的度量和测试。

1. **团队（个人）工作中哪些风险发生了，其影响又是怎样的？**

团队风险：团队里有两个人脱单，导致经常赶ddl的只有两个人，然后造成了一些质量方面的缺陷。经常是在最后一分钟交作业。分工方面有风险，各自负责各自的层，导致最后集成测试的时候出现很多矛盾和问题。

1. **在何时做出了哪项重要决定？**
2. 构造时候将common的部分全部弄出来形成一个新的层
3. 构造的同时将更改前面的接口和文档。
4. **这个决定又是如何影响这个项目的？**
5. 方便其他两个层的调用共同的东西，如PO
6. 方便组内的即使沟通。
7. **所遇到的主要问题是什么？**

团队里有两个人脱单，导致经常赶ddl的只有两个人，然后造成了一些质量方面的缺陷。经常是在最后一分钟交作业，主要是平时的效率问题。分工方面，各自负责各自的层，导致最后继承测试的时候出现很多矛盾和问题。

1. **对这些问题的解决方法产生了什么效果？**

通过返工进行小组内的交流，然后加强小组沟通。

对于赶ddl至今没有解决，每次都是最后一分钟。

1. **开发团队成员是如何看待自己的职责的？**

小组成员虽然出现了各种各样的情况，但是总体来说，都对项目倾注了自己的热情，体现了每个人的责任感，对大作业没有懈怠，整个过程有点坎坷，但是大家还是坚持到了看见胜利的果实。