快

**物流管理系统（LMS） 项目评价文档**

**编写者： 崔浩 杜天蛟 高露 倪安松**

目录

[一、 项目管理评价 2](#_Toc439364410)

[二、 产品评价 2](#_Toc439364411)

[1.产品在项目生命周期中的变化 2](#_Toc439364412)

[2.产品返工情况 3](#_Toc439364413)

[3.开发工具 3](#_Toc439364414)

[4.产品规模 3](#_Toc439364415)

[5.产品质量 3](#_Toc439364416)

[6.产品度量数据及分析 4](#_Toc439364417)

[三、 团队评价 5](#_Toc439364418)

[1. 团队总体评价 5](#_Toc439364419)

[2.个人自我评价 5](#_Toc439364420)

[3.团队会议记录 7](#_Toc439364421)

## 修改记录

|  |  |  |
| --- | --- | --- |
| 修改人 | 修改时间 | 修改原因 |
| 倪安松 | 2015/12/30 | 初始版本 |
|  |  |  |

## 项目管理评价

在本次“快递物流系统（LMS）”的开发过程当中，项目管理整体较为良好。项目所使用的过程是迭代式开发，每一个阶段都能有部分产品产生，极大地鼓舞了团队士气。但是，实际的过程和当初的设想有所不同，在当初的设想中，从设计阶段开始，每一次的迭代都应该产生可运行的代码；然而在实际开发过程当中，从构造阶段开始才有了可执行的代码。

在进度方面，我们按照作业时间的提交期限，每一个阶段的产品都在提前1-2天完成，并且在最后一天做最后的检查和提交，应该说几乎所有的同步点和里程碑都按时到达。

在过程当中，由于经验不足的原因，需求阶段和体系结构设计做的不是很好，需要在下次的项目当中强化和改进；详细设计阶段设计的较为出色；构造阶段分工并不是很合理，导致每个人的工作量不是很均衡，好在后来将提前完成自己任务的队员帮助尚未完成任务的队员完成任务，才将整个项目的构造阶段顺利完成，但构造阶段总体时间花费较多，改动太大。

需求阶段花费时间大概为15天；设计阶段中，体系结构设计花费13天，详细设计花费20天；由于构造阶段和部分设计阶段的工作是并行的，所以构造整体时间花费大约为30天。

## 产品评价

### 1.产品在项目生命周期中的变化

在项目的需求、体系结构、详细设计阶段，产品主要以文档为主，包括需求阶段的用例文档、需求文档、需求度量文档，设计阶段的体系结构文档和详细设计文档、测试用例文档等，此外包括少量的代码，主要是桩和驱动，单元测试用例等。在项目的构造阶段，代码量发生了巨大的变化，产品基本成型，但还存在不少缺陷，界面也不够美观。在测试阶段，我们修复了找到的缺陷，并美化了界面，同时写了产品测试报告，产品基本达到交付标准，在此阶段我们准备了产品的部署工作，并编写了产品介绍PPT。

### 2.产品返工情况

因为项目经验不足，在这次项目开发过程中，我们出现了不少返工情况。问题主要集中在以下方面：

1. 在项目设计阶段发现对需求的理解不足，需求阶段一些关键需求的具体内容没有能细化，导致设计阶段有些功能无法设计，要回到需求文档落实细化部分需求。
2. 在构造阶段，我们发现设计阶段定义的接口不够稳定，出现过一些修改，由此导致层与层之间的冲突，出现集成失败的情况。需要修改设计阶段定义的接口。

### 3.开发工具

在项目开发过程中，我们使用Word编写文档，用UmlET画设计图，使用Github进行配置管理，用maven进行项目构建，使用Jenkins进行项目的持续集成，使用eclipse进行软件构造，Junit进行单元测试，使用Metrixs进行软件度量。我们使用大量的工具，成功提高了项目的开发效率和产品质量。

但是在项目开发过程中，我们在Github上遇到一些问题，但我们学习使用Git Shell解决问题，让问题保持在可控的范围内。

总体来说，我们使用的工具支持产品的制造、维护和测量。

### 4.产品规模

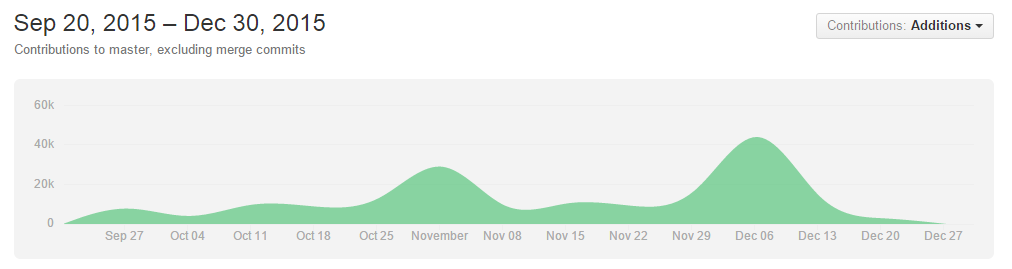
最终产品共有17181行代码。

### 5.产品质量

因为有前期大量的文档支持和采用了良好的设计模式和编码规范，项目的产品质量较之前的项目有很大的提高。但因为开发经验不足和开发能力有限，产品难免存在缺陷。我们在开发过程中已经尽力提高产品的质量，尽我们最大的努力使产品达到交付标准。在测试阶段，我们约20多个有记录的产品缺陷，并修复了这些问题。

### 6.产品度量数据及分析

#### 产品的增长情况



#### 产品在每个里程碑上的测量

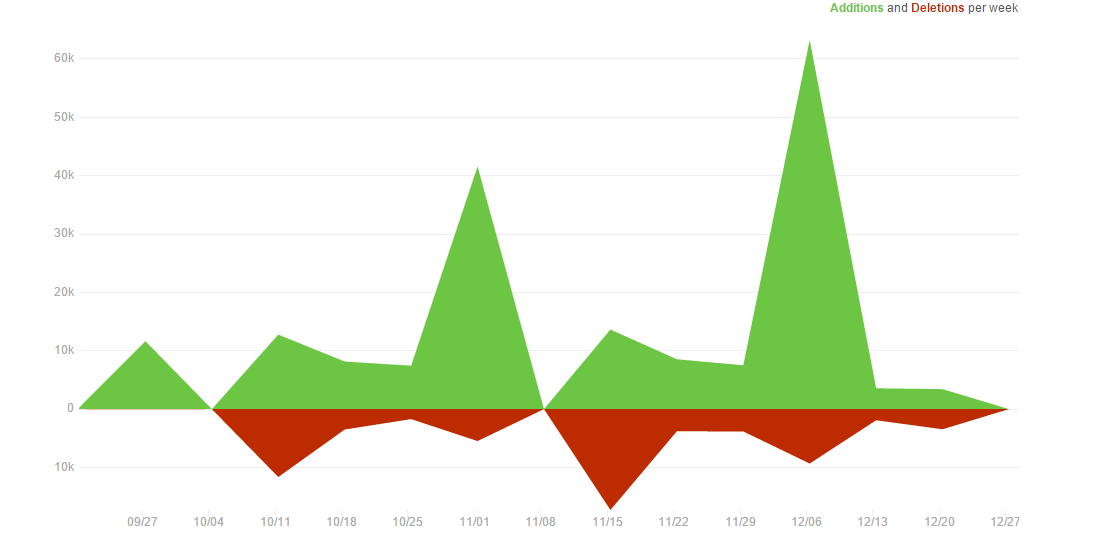
|  |  |  |
| --- | --- | --- |
| 里程碑 | 时间 | 成果 |
| 需求阶段 | 2015/9/30——2015/10/14 | 软件需求规格说明书、软件需求度量文档 |
| 体系结构设计阶段 | 2015/10/14——2015/10/26 | 体系结构设计模型、体系结构设计文档、体系结构原型代码（含桩和驱动） |
| 详细设计阶段 | 2015/10/26——2015/11/07 | 详细设计模型、详细设计文档、详细设计的集成测试代码和驱动代码 |
| 构造阶段 | 2015/11/07——2015/12/09 | 可执行程序（功能实现率86%） |
| 测试阶段 | 2015/12/09——2015/12/20 | 系统测试用例和测试报告文档（功能实现率95%） |

#### 产品复杂度和内容的测量

代码量为17181行，共有326个类。该产品实现了快递物流系统的基本功能需求，实现了快递公司操作的一体化。

#### 过程和工具对产品的影响

本项目采取瀑布模型为开发过程模型，开发周期为三个月。开发过程的频率如下图。



采用Jenkins、maven进行构建和集成，提高了集成的效率。利用junit进行测试驱动开发，减少了代码的缺陷。通过git进行项目的同步和备份，使小组成员可以随时获得项目的最新进度，也有助于项目开发过程中的回退。

## 团队评价

### 团队总体评价

团队总体评价良好。管理当中常见的问题有组内人员空闲时间不统一，导致难以确定开会和集体开发的时间；当出现争执时不能够冷静地通过讨论优缺点来提出解决方案，容易情绪化。主要优点体现在定好了会议时间以后，大家都能较为准时地参加；大家都能以团队项目为重，能够意识到“团队的失败就是每个人的失败”，当出现问题的时候都能够停下手中的工作来修复项目问题；在布置任务之后都能够按时完成，没有任何拖延的现象。

在团队过程当中，风险集中出现在构造阶段，由于对项目代码量的没有准确的把握和最后期限的限制，导致组内对于能否按时完成构造产生了一种怀疑的态度，好在随着过程的进行，组员们渐渐意识到项目能够按时完成，有了充足的信心。相信这样的问题能随着经验的上升慢慢改进。

在团队管理当中做出的最重要的决定应该是构造阶段的分工决定。由于构造阶段较为重要，分工方式又有很多，但是考虑到我们的体系结构是分层式的设计，拥有并行开发的特点；又考虑到界面层要学习很多并不掌握的UI知识，还要承担美工的任务；最后，考虑到我们做这个项目仍然处于一个学习的阶段，我们的团队已经在上次的EL比赛中就进行过初步的分工，我的决定是为了让组员们更多的接触和熟悉新的知识，应该分配给他们与上一次不同的任务。于是决定分配崔浩和杜天蛟写界面层；高露写逻辑层；倪安松写数据层。

### 2.个人自我评价

**倪安松——**在这次项目实践的过程当中，我第一次作为队长带领这个团队。虽然在以前带领过各种各样的很多不同性质的团队，但是由于这次组内成员水平都比较高，我还是感受到了当组长的压力。压力主要表现在每一次分工的过程当中，要充分考虑组员的技术优缺点，作业量的不同，最为合理，最为公平地分配任务。其次，在看法不一致的情况下，我尽量都希望通过沟通和分析达成一致，但是难免有谁也没有办法说服谁的情况，这时候只能听组长(我)的安排，有时候组员会很抵制，但是之后还是会完成任务。

在这次的项目实践中，我学会了利用java的反射实现PO的更新删除保存等，而且学会了MySQL的基本操作，学会了java RMI的实现，以及在内外网不同的情况下如何实现java RMI的调用。为了项目数据库同步，我学会了如何利用xshell远程控制linux服务器，并且将代码部署在远程服务器上。在工具使用方面，我学会了MAVEN的配置POM的编写以及Jenkins的集成，学会了包括JDBC和DOM4J的jar包的使用等

**崔浩——**在这次大作业过程中，我在需求设计等阶段和组员合作完成文档的编写，在构造阶段主要负责界面层架构的设计，并和组员一起完成界面层的编写。我能积极参与团队的每次会议，但在开会过程中，我经常和组长和组员有一些冲突。但最终我们都能做出让步并互相妥协。在这次大作业的过程中，我增进了对组长和组员的了解，意识到团队的每个人都是项目不可或缺的一部分，一个良好的项目一定是团队的所有人合作完成的。尽管冲突不可避免，但我们学习掌握处理冲突的方法才是最重要的。

**杜天蛟——**在完成这次项目的过程中，我学习了java的反射机制，通过反射构建类的对象，通过类名和方法名执行方法，减少了大量重复代码。我还学习了xml文件的配置和读取，利用第三方jar包实现，把界面的所有组件写在xml文件里，通过读取xml文件和反射的结合来创建组件，所有的位置、大小以及其他信息都写在了xml文件里，避免了硬编码现象。

这次项目也培养了自己面向对象的思维，也想了一些办法减少代码的圈复杂度，比如增加类，通过类名执行方法等。

但是在这个过程中也发现了自己的不足，比如考虑问题不周全，没有养成防御式编程的习惯。还有项目即将结束时，自己设计项目框架的能力还非常弱，对整个项目没有整体把握，只能在小范围内实现一些具体的功能。

在详细设计的过程中，自己写了项目所有的po和大部分接口、vo，但是在写代码的过程中却发现很多设计都非常不合理，也给自己带来了很大麻烦。相信吸取了这次的经验后，下次设计会有一些进步。

**高露——**在做大作业的过程中，我学到了不少知识。首先自然是对分层的体系结构风格、模块化等有了更深一步的了解，对各层之间的分工、协作以及其之间的关系也有了了解。在逻辑层的实现中，操作基本上实现vo，po的相互转换，在实现的过程中，有时也要去看看数据层或界面层的代码。在此过程中，我也学习了数据库的相关语句操作，包括建表、增删改查等；也了解了一些界面层与数据层用到的反射的机制，这对我的学习无疑是一大提高。当然在完成大作业的过程中，我也发现了自己的很多不足。比如逻辑层的异常处理机制做的不好，只是随意的抛出异常并做可能并不合适的处理；有一些代码重复的现象；有时候只是为了实现一些功能，而不考虑与其他模块的联系，忽略内聚耦合等等。而正是有了这些问题，我也学到了一些知识，比如异常处理的方法、抽象工厂模式等。经过一次大作业的锻炼，相信以后我也会不断提高自我。

### 3.团队会议记录

1. 会议记录

第八组评审会议记录1

会议时间：

2015.10.10

会议地点：

丙区512

出席人员：

倪安松 崔浩 杜天蛟 高露

会议内容：

1. 需求用例描述讨论

与会者发言：崔浩提出UC20（制定城市、距离用例）应该拆分成两个，因为其中包含了与该用例名称无关的用例价格的修改

与会者发言：倪安松认为为保持用例粒度合理，可以不必拆分，只要改变用例名字即可

投票表决结果：该用例不予拆分，但为了描述准确，应改变用例名称

与会者发言：崔浩提出UC18人员机构管理用例应该拆分成人员管理和机构管理

与会者发言：倪安松认为机构管理和人员管理虽然相对独立，但是系统中应该将人员机构看成一个整体进行管理

与会者发言：崔浩认为功能上相对独立就应该分开

投票表决结果：该用例不予拆分

与会者发言：倪安松提出财务人员的用例描述中的运费标准没有详细描述

与会者发言：高露提出运费标准的含义（即是指收取客户的费用，还是属于公司自己的成本）

与会者发言：倪安松认为财务人员处理公司成本账单，而总经理负责收益表

与会者发言：杜天蛟认为用户添加的方式应该是公司员工申请账户，然后管理员批准而不应该管理员自己添加

与会者发言：崔浩认为实施起来难度太大

与会者发言：倪安松认为登录是进入系统的入口，如果无法登陆，则无法验证更无法进行申请，如果要实现要新增加用例，同意崔浩的观点，实施起来难度太大

投票表决结果：添加方式保留为管理员添加账户

1. 详细需求规格说明文档修改

与会者使用需求评审检查列表[Wiegers2003]，对需求规格说明文档进行了修改，具体如下：

1. 统一了系统中使用的默认数据和数据格式
2. 删减了非功能性需求3.3.4中的第一二条，降低了实施难度
3. 统一规定了财务人员期初建账时账本的详细内容
4. 将部分业务规定的内容用info的形式写到系统级需求中，方便理解和查阅
5. 评审报告

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | 物流管理系统 | | |
| 评审名称 | 需求评审 | | |
| 是否复审 | 否 | 评审级别 | 小组 |
| 评审方式 | 会议方式 | 记录人 | 杜天蛟 |
| 所属开发阶段 | 需求开发阶段 | | |
| 评审目的 | 评审第一阶段的需求开发，更正一些错误，防止带入下一阶段 | | |
| 评审检查结果 | |  |  |  | | --- | --- | --- | | 检查点数量 | 通过数量 | 不通过数量 | | 22 | 14 | 8 | | | |
| 评审结论 | 通过评审 | | |
| 权限修正及验证 | | | |
| |  |  |  | | --- | --- | --- | | 序号 | 缺陷内容 | 修正措施 | | 1 | 非功能性需求不现实 | 进行适当删减 | | 2 | 部分系统级需求描述详细程度不当，遗漏了必要的信息 | 进行适当的改正 | | | | |

第八组评审会议记录2

会议时间

2015/10/14

会议地点

机房

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

主要讨论并设计了vo，po的类以及各个vo、po的内容，讨论了大作业的模块划分问题，最后确定了设计department，finance，list，log，personnel，transport，user，warehouse八个模块。

第八组设计阶段评审会议记录

会议时间

2015/10/24

会议地点

图书馆研讨间

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 讨论修改了上次设计的vo，po
2. 就上次会议设计的八个模块进行了接口的设计（包括需接口与供接口），并依此完成了体系结构原型代码的初步建立
3. 一起讨论了桩程序与驱动程序的写法与设计时的考虑点及注意事项，讨论完成后按模块分工每个人完成自己的桩程序与驱动程序的设计与编写。

第八组会议记录3

会议时间

2015/11/4

会议地点

机房和图书馆研讨间

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 讨论并确定了界面的主体风格，并设计了总体的原型界面的模板。确定了绘制原型界面的工具（axure RP pro），之后分模块每个人完成对应模块的原型界面的设计。
2. 讨论研究了每个模块的设计类图，各个模块操作的基本流程，然后据此设计了每个模块的顺序图。
3. 完成了对详细设计文档的初步设计与编写。

第八组会议记录4

会议时间

2015/11/15

会议地点

机房

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 修改并完成了对详细设计文档的设计与编写。
2. 完成了各模块的顺序图
3. 修改完善原先的初步设计模型，形成最终的详细设计模型

第八组会议记录5

会议时间

2015/11/16

会议地点

图书馆研讨间

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 根据上次会议的详细设计文档与设计模型，讨论了集成测试mock代码与测试驱动代码的设计，并分模块由每人完成自己的代码编写
2. 讨论寻找了度量数据的工具，研究了书上度量数据的方法，并进行了内聚与耦合的度量

第八组会议记录6

会议时间

2015/11/18

会议地点

图书馆研讨间

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 根据逻辑层、数据层和界面层分配代码的构造任务
2. 完善详细设计和接口
3. 确定数据的持久化保存方式
4. 完善包结构
5. 修改vo po

第八组会议记录7

会议时间

2015/11/24

会议地点

机房

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 修改接口和vo po
2. 编写界面层的桩和mock service

第八组会议记录8

会议时间

2015/12/04

会议地点

机房

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 交流遇到的bug并集中处理
2. 修改详细设计文档
3. 交流实现具体特色功能

第八组会议记录9

会议时间

2015/12/17

会议地点

图书馆研讨间

参与人员

崔浩 杜天蛟 高露 倪安松

会议主要内容

1. 把四个人的缺陷报告拿出来审查，然后集中处理缺陷。
2. 分配测试度量、测试报告文档等任务
3. 修改系统测试用例
4. 增加新的测试用例并进行测量