

软件工程与计算III

迭代三、产品、以及期末检查

刘嘉

Outline

- Part 1 迭代二部署检查的情况
- Part 2 “产品化” 的诉求
- Part 3 期末检查的要求点（含迭代三中期检查和提交情况）

Part 1 迭代二部署检查的问题

1、部署部分比较好的示例（1）

- **161250179 GeekMark**: 部署文档简约有效，系统能够成功运行。给出了启动脚本和停止脚本。

GeekMark 部署指南（傻瓜式）

STEP 1:

将GMWeb/Deployment文件夹中的GMWeb.jar、startup.sh、shutdown.sh复制到需要部署的目录下，三个文件在同一个目录下即可

STEP 2:

赋予两个sh脚本执行权限

```
sudo chmod +x startup.sh
sudo chmod +x shutdown.sh
```

STEP 3:

运行程序

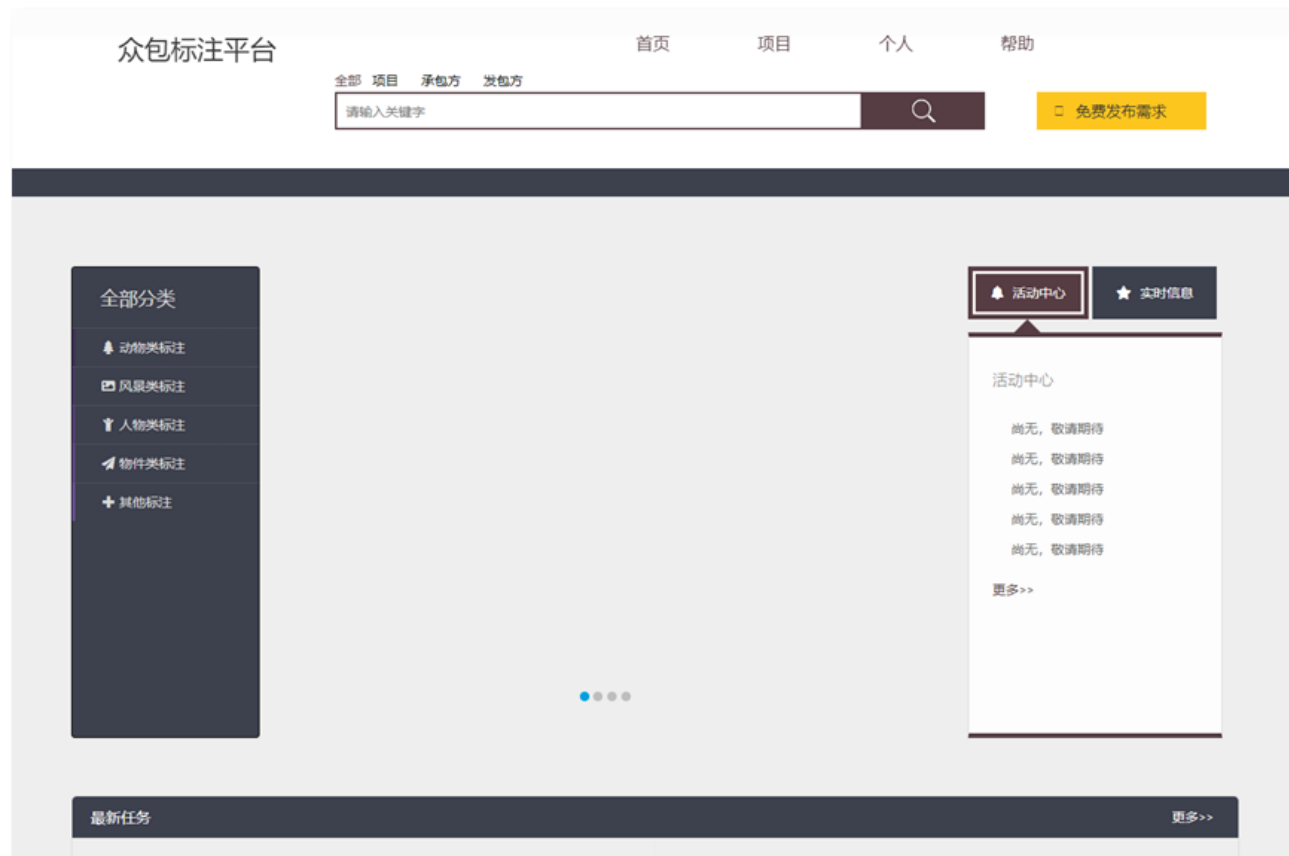
```
./startup.sh
```

停止程序

```
./shutdown.sh
```

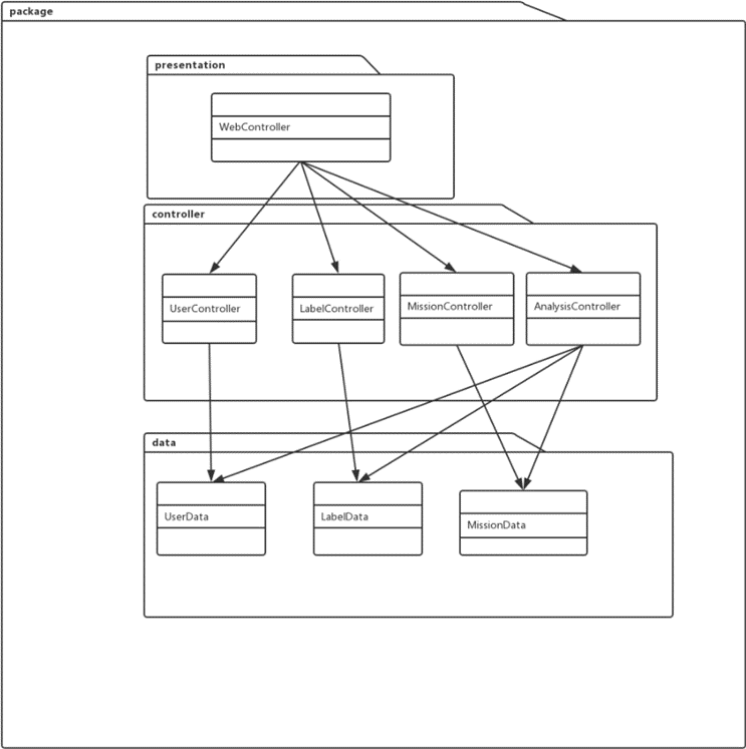
1、部署部分比较好的示例（2）

- **161250144_CWZZ**: 文档步骤清晰，部署方式正确，（部署在阿里云上）



2、设计部分比较好的示例（1）

- **161250036 COUNTS FX:** 设计部分，包含了组合视角、接口视角和信息视角，层次清晰，定义明确。代码文档实现部分使用了分层的风格，严格按照设计实现的



模块	职责
UserController	实现登录登出所需要的服务
LabelController	实现标记所需要的服务
MissionController	实现任务所需要的服务
ImageController	实现图片上传，下载的服务
AnalysisController	实现分析所需要的服务。

5.4 数据层的分解

数据层主要是给控制层提供数据访问服务，包括对于结构化数据的查询，以及将非结构化数据转换为结构化数据。数据层模块的职责如下表所示。

模块	职责
UserData	提供用户信息数据的访问服务，以及将非结构化数据转换为结构化数据。
LabelData	提供标签信息数据的访问服务，以及将非结构化数据转换为结构化数据。
MissionData	提供任务信息数据的访问服务，以及将非结构化数据转换为结构化数据。

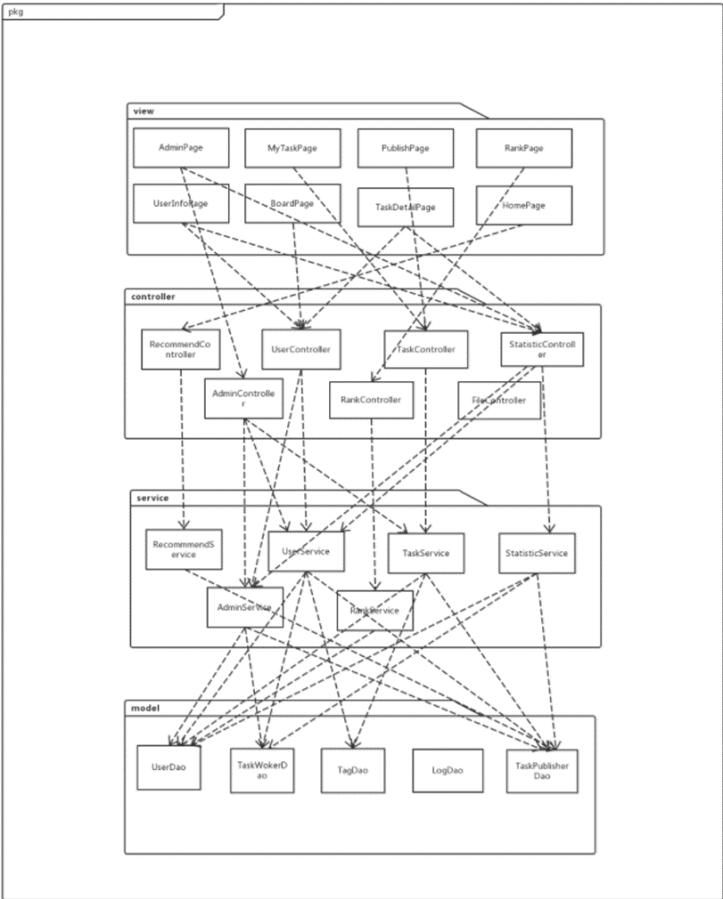
6. 信息视角

6.1. 数据对象

类名	包含的属性
User	id,用户名,密码,手机号码,邮箱,身份,用户简介,头像,任务列表,积分,评分,级别,正确性,省份,城市。
Mission	id,requestorID,missionName,description,begin,end,type,selects,maxNumber,currentNumber,points
AttributeLabel	fileName,attributes
CaptionLabel	fileName,capture
ClassificationLabel	fileName,select
DetectionLabel	fileName,rectangles
SegmentLabel	fileName,dots

2、设计部分比较好的示例（2）

- 161250053_Horizon:同上，且接口规范设计部分更为详细



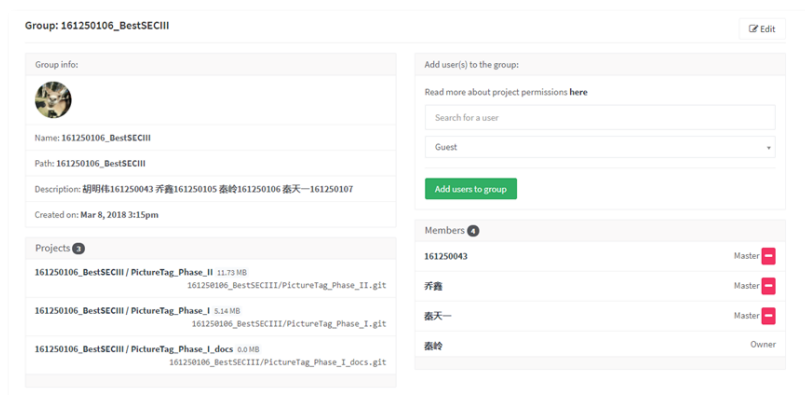
UserService模块的接口规范		
提供的服务	提供的服务	提供的服务
UserService.findUserById	语法	User findUserById(long id)
	前置条件	无
	后置条件	根据用户id拿到用户，如果不存在符合条件的用户返回null
	语法	ResultMessage register(User user)
UserService.register	前置条件	该用户未被注册过
	后置条件	添加持久化数据，若邮箱或电话号码存在返回用户已存在
UserService.login	语法	List<Object> login(String name, String password)
	前置条件	该用户已经注册
	后置条件	加载用户数据登录任务列表界面，登录成功
UserService.findTaskWorkerByState	语法	PageDTO<TaskWorker> findTaskWorkerByState(Long userid, TaskState taskState, Integer size, Integer currentPage)
	前置条件	该用户id在数据层存在
	后置条件	根据状态查找用户任务，返回分页信息
UserService.submitTask	语法	ResultMessage submitTask(TaskWorker taskWorker, List<Tag> tags)
	前置条件	该用户已经接受该任务
	后置条件	更新持久化数据
UserService.attend	语法	ResultMessage attend(Long userid)
	前置条件	无
	后置条件	更新持久化数据，签到成功或已经签到
UserService.deleteTask	语法	ResultMessage deleteTask(Long taskWorkerId)
	前置条件	该任务在数据层已存在
	后置条件	删除持久化数据
UserService.recharge	语法	ResultMessage recharge(Long userid, double amount)
	前置条件	无
	后置条件	更新持久化数据
UserService.findTaskPublisherById	语法	TaskPublisher findTaskPublisherById(long id)
	前置条件	该发起者任务在数据层已存在
	后置条件	查找持久化数据
UserService.getFitTags	语法	List<Tag> getFitTags(long taskWorkerId, String filename)
	前置条件	无
	后置条件	得到符合要求的Tag

相互学习也是软件以及软件工程重要的工作

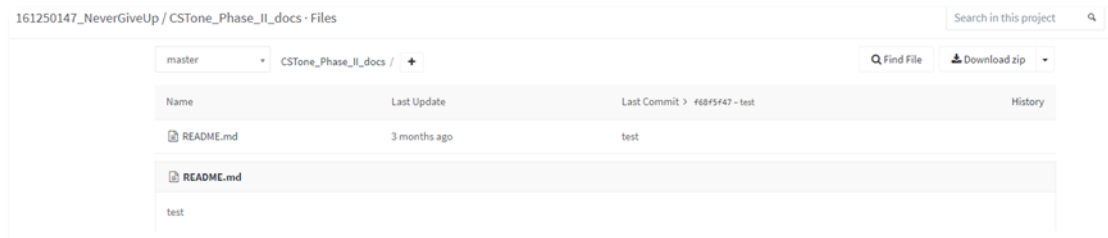
- Gitlab上，所有项目都是可见的
- 参考好的小组，重新考量自己的设计和软件工程过程
- 期末答辩的时候，请在软件工程过程部分，描述整个课程实践期间看了哪些小组的项目以及对这些项目的评价及借鉴。

3、存在一些问题的示例（1）

- 找不到部署文档的，甚至迭代二的文档仓库都没有的
 - 161250106_BestSECI**：没有迭代二文档仓库

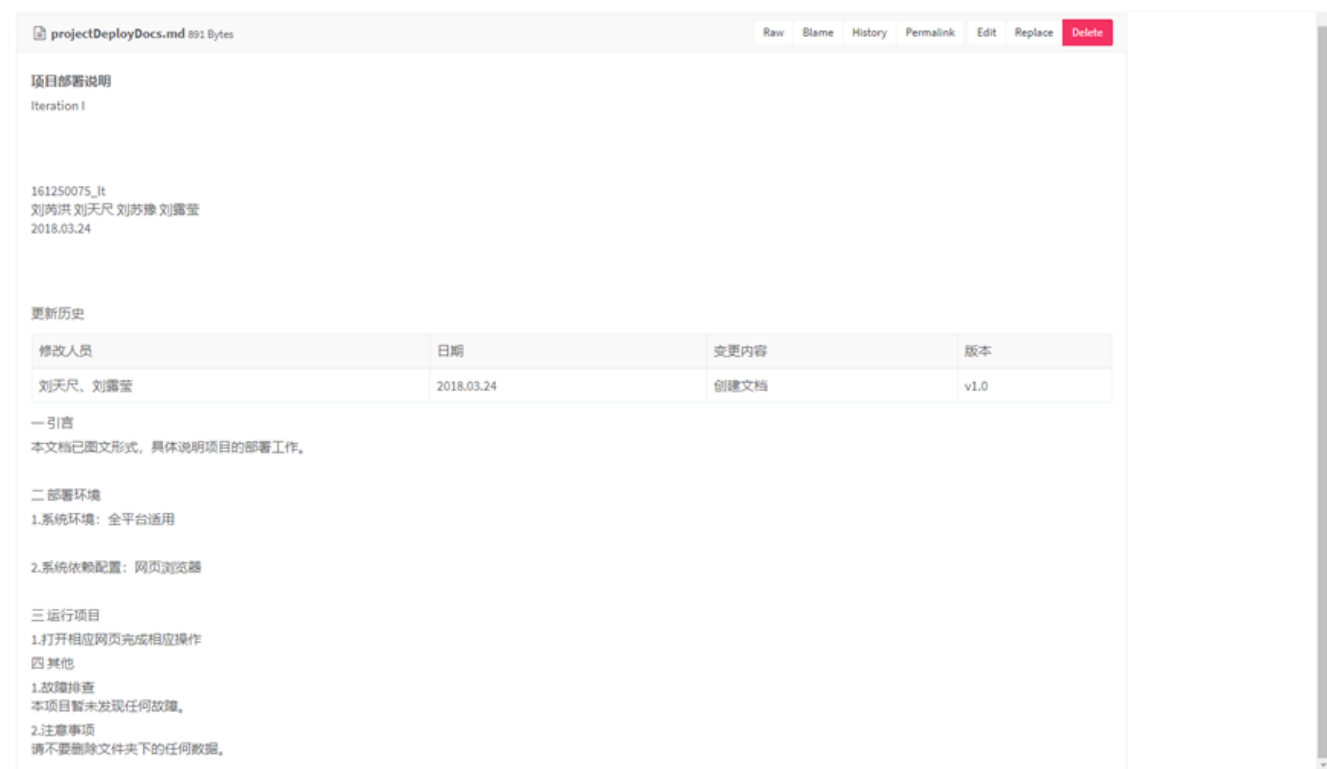


- 161250147_NeverGiveUp**：有迭代二文档仓库，但没有部署文档



3、存在一些问题的示例（2）

- 部署文档等于没写的
 - **161250075_It**: 只有部署文档的框架



3、存在一些问题的示例（3）

- 部署文档写成用户手册的

- **161250078SEHW3**: 和部署相关的只有两行，剩下的大篇幅都是系统的使用手册
- **161250177_GorgeousSECIII**: 和部署文档无关，整篇文档都是用户手册

一、引言
本文档以图文形式，具体说明项目的部署工作

二、部署环境

1. 系统配置：全平台适用

2. 系统依赖配置：系统需要安装JDK，如未安装JDK，可以先在<http://www.oracle.com/technetwork/java/javase/downloads/index.html>网站下载，然后参考<http://jingyan.baidu.com/article/6dad5075d1dc40a123e36ea3.html>网站进行安装与环境变量配置。

三、运行项目
从gitlab上下载压缩包后，解压缩，先运行SEHW_ii_server.jar,再运行SEHW_ii_client.jar
下载地址：http://114.215.188.21/161250078SEHW3/SEHWIII_Phase_II/blob/master/SEHWIII_ii.zip

四、用户手册

1. 注册

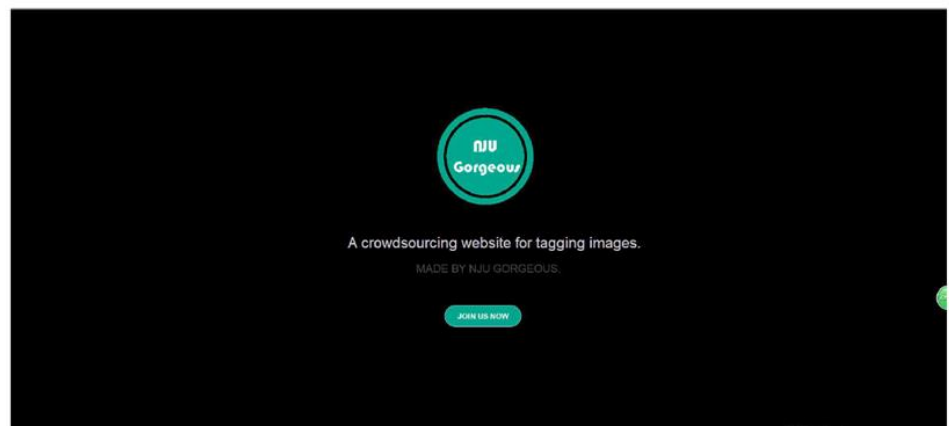


注册界面输入姓名，密码，选择身份（众包发起者/众包工人）进行注册，id自动生成，无需填写



迭代二 众包系统使用流程说明

1. 找到项目里的 PicturesApplication 并启动，在浏览器中打开网址
<http://localhost:8090/html/index.html> 到达 登 陆 注 册 界 面



3、存在一些问题的示例（4）

- 部署文档写成如何在IDE上run的，而不是jar包war包等部署文件
 - **161250178_ArgusTags**：在IDE里面run的步骤文档
 - **161250182_SE3**：部署文档要求clone仓库后导入IDE运行启动。没有提供可以部署的jar和war包以及部署脚本。

在 IDEA 本地部署此项目

1.在 IDEA 中导入项目

导入项目时，选择 Project From Version Control -> Git，点击 Clone

2.配置项目

IDEA 打开了项目之后，在 Event Log 处会弹出提示信息

关于 Maven

点击 Add As Maven Project，将项目转换为 Maven 项目

如果 Event Log 中没有出现这条信息，则在 Project 中找到 pom.xml，右击 -> 选择 Add As Maven Project

关于 sdk

点击窗口上方出现的 select sdk 选择 java1.8 并点击确认

3.运行项目

右键点击 src/main/java/argustags_phase_ii/ArgusTagsPhaselIApplication 选择 RUN'ArgusTagsPhaselIApplication' 即可运行

1.部署环境

本系统需要 Tomcat9.0 及以上版本并配置好环境变量，jdk1.8 及以上版本，支持 IE8 以上及 360 浏览器，支持 windows10 系统，默认运行 8080 端口（若占用请解除占用）
附 Tomcat、JDK 安装教程：https://blog.csdn.net/qq_32519693/article/details/71330930

2.客户端配置

客户端机器配置要求：默认文件存储在用户本机，请确保用户本地磁盘有足够的空间存储数据
客户端软件配置要求：Tomcat9.0 及以上版本、jdk1.8

3.系统安装

从 gitlab 上 clone 此项目，运行在 idea（推荐）上

3、存在一些问题的示例（5）

- 部署文档写的步骤比较简略
 - **161250162 CrowdsourcingTags**: 文档仓库结构相对不清晰，部署文档写的过于简单，没有为裸机提供详细的部署方案。按照部署文档进行部署，未成功。
 - **161250202 Discounts**: 对部署过程描述简单，没有提供可以使用的war包，可以部署，功能缺失。

CrowdTag--IterationIIFinalDraft

部署

- 安装 `Gradle`
- 在项目目录下运行 `gradle clean build`，生成的 `jar` 包位于 `build/libs` 目录下，使用 `java -jar build/libs/CrowdTag-0.0.1-SNAPSHOT.jar` 运行
- 打开 `localhost:8080`



3、存在一些问题的示例（6）

- 部署文档写的很好的，但是按照文档部署后缺失功能、或云服务器无法访问
 - **161250127 COUNTS**: 部署文档内容详尽，给出jar包和步骤在mac环境下不能成功运行，建议考虑不同系统的路径问题。
 - **161250119_ControlISECIII**: 文档步骤不够清晰，能部署，但是云服务器端无法访问



关注Moodle上的公告

• 有问题的组联系助教提交更新的部署说明

	A	B
1	组名	详细说明
2	161250033_crowdSourcing	部署及使用说明文档中，仅两行提及部署方法，未详细说明部署步骤。在其迭代二项目仓库中找不到可以部署的jar或者war文件。给出了部署好的服务器。
3	161250151_TagMakers	在迭代二项目根目录下有脚本运行，从项目重新编译jar和前端项目。编译报错。部署文档写的较为详尽。
4	161250021_Geass	无文档，无项目
5	161250068_MINECODE	有部署产物，部署文档写的也相对详尽。文档中端口号与实际需要修改的端口号不匹配。在尝试过部署文档中写的8080端口以及实际代码中的8000端口后部署未成功。项目是前后端分开部署的，给出了部署好的服务器。
6	161250147_NeverGiveUp	无部署文档，有项目，无部署产物。
7	161250091_CrowdsourcingTags	部署文档中要求将项目clone到本地后运行main方法启动项目。迭代二代码仓库中没有可部署的jar和war包以及部署脚本，给出了部署好的服务器。
8	161250128_STZ	部署文档中要求在IDEA中导入项目后运行main方法启动，没有提供可以部署的jar和war包以及部署脚本。
9	161250096_Utag	部署文档写在代码仓库的README文件里，要求clone仓库后导入IDE运行main方法。仓库太大，800多MB。
10	161250015_OiCAFEBABE	部署文档写的不够清晰，要求导入工程后运行main方法。迭代二代码仓库中没有可部署的jar和war包以及部署脚本，给出了已经部署好的服务器。
11	161250053_Horizon	部署文档写的比较详尽，项目给出的jar和运行命令可以成功的运行项目。建议给出一个运行脚本以便在后台运行以及停止项目。
12	161250179_GeeKMark	部署文档简单有效，系统能够成功运行，给出了启动脚本和停止脚本。
13	161250040_KylinSECIII	部署文档中要求在IDEA中导入项目后运行main方法启动，没有提供可部署的jar和war包以及部署脚本。
14	161250100_Conux	部署文档中依靠ng server启动前端项目，后端项目依赖于IDE启动，没有提供可部署的jar和war包以及部署脚本。
15	161250075_It	文档中说以图文形式，未见图文没有什么具体的步骤
16	161250115_CTsHiddenFigures	部署文档步骤明确清晰，项目启动后可正常运行
17	161250178_ArgusTags	文档步骤清晰，但不能算是部署文档是在IDE里面run的
18	161250156_SpotsSECIII	没有迭代二的文档仓库，迭代二的项目仓库中也未找到部署文档
19	161250207_SEC3A644	文档步骤清晰，但不能算是部署文档是在IDE里面run的
20	161250065_foursome3E	文档步骤清晰，后端使用runable jar包部署前端未说明build之后需要放入nginx/tomcat
21	161250144_CVZZ	文档步骤清晰，部署方式正确且已经成功部署在阿里云上
22	161250063_stars	只有使用文档，未找到部署文档指出已经部署在阿里云上，但是无法访问
23	161250167_Counts	文档步骤清晰，项目运行正常
24	161250004_Joker	文档步骤清晰，部署方式正确且已经成功部署在云服务器上
25	161250119_ControllerSECIII	文档步骤不够清晰，能够部署，但是云服务器端无法访问
26	161250078SEHW3	使用文档，而不是所需要的部署文档运行的步骤不够清晰且部署后并未正常运行
27	161250177_GorgeousSECIII	写的是使用文档的内容没有部署步骤，所谓的步骤只是在IDE中run
28	161250024_PIONEER	项目前后端分离，前端node打包失败，后端为导入项目运行没有提供可以部署的jar和war包以及部署脚本。
29	161250182_SE3	部署文档要求clone仓库后导入IDE运行启动，没有提供可以部署的jar和war包以及部署脚本。
30	161250169_CrowdsourcingTags	提供文件和相关jar包，文件过程详细，给出了已经部署好的服务器。
31	161250198_COUNTS	部署文档描述简单，且要求在IDEA中导入项目后运行main方法启动，没有提供可以部署的jar和war包以及部署脚本。
32	161250127_COUNTS	部署文档内容详尽，给出jar包和步骤在mac环境下不能成功运行，建议考虑不同系统的路径问题
33	161250123_SECIII	部署文档中依靠npm启动前端项目，后端项目依赖于IDE启动，未提及安装node.js，没有提供可部署的jar和war包以及部署脚本。
34	RandomGroup7	未找到项目
35	161250020_SEC3	迭代二文档仓库内无相关文档，代码仓库上传了涉及的jar包，但没有部署的jar包或war包以及部署脚本
36	161250103_YingYingMonster	部署文档流程清晰，运行相对简单，没有提供可部署的jar包，从项目重新编译jar包可以运行
37	161250171_SECIIICS	采用了前后端分开部署，部署文档描述相对详细，提供了运行的jar包，可以启动。但前端npm启动失败
38	161250143_SocialEngineers	给出了部署好的服务器，所以部署文档类似使用手册，而不是部署文档，没有可部署的jar和war包以及部署脚本
39	161250181_DebugSECIII	给出了部署好的服务器，部署文档对可能出现的部署失败情况做了较多列举和解决方案，部署文档中要求在IDEA中导入项目后导出war包运行，未提供对应的war包，且项目其他的jar包不用上传至代码仓库（建议直接提供war包和对应的部署脚本）
40	161070068_Markis	文档仓库为空，没有部署文档，没有提供可部署的jar包或war包以及运行脚本，代码仓库上传了部署失败的其他jar
41	161250112_SECIII	有详尽的部署文档，使用SpringCloud微服务框架，没有给出可以直接运行的jar，war包，有部署到自己的服务器但是无法访问
42	161250108_BazingaTurk	部署文档简单，可以通过文档实现部署，但是具体功能无法使用
43	161250162_CrowdsourcingTags	文档仓库结构相对不清晰，部署文档写的过于简单，没有为练习提供详细的部署方案。按照部署文档进行部署，未成功。
44	161250013_Tager	没有提供有效的部署文档，部署在服务器端，但是无法打开
45	161250102_AlphaCat	部署文档对功能介绍较为详细，但是没有给出部署方案，有在服务器上部署，但是无法访问.http://120.79.145.97:8080/index
46	161250057_COUNTS	文档仓库结构混乱，没有给出可以直接运行的jar，war，需要在IDE中打开，能打开，但是具体功能不齐全。
47	161250058_CrowdsourcingTag	对部署过程描述简单，没有提供可以使用的war包，并且无法在IDE中使用
48	161250022_Discounts	对部署过程描述简单，没有提供可以使用的war包，可以部署，功能缺失。
49	161250010_Trapx00	提供了详细的项目部署说明文档，给出了可以直接运行的jar包
50	161250136_10k	有详细的部署文档，已经部署到服务器，但是无法访问，没有给出直接使用的jar，war包，给出了执行脚本。
51	161250066_secIII	文档仓库没有按照规定要求构建，部署文档简单，混乱，没有给出可以执行的jar，war包，需要在ide中启动。
52	161250036_COUNTS_FX	已将项目部署到服务器上，有部署文档，部署部分过于简单，模糊，无法根据部署文档部署，没有给出可以直接运行的jar，war包
53	161250132_SECIII	步骤清晰，给出了看可以直接执行的jar包，根据部署文档，jar包无法正常使用。
54		

这次实践挖了很多坑啊.....

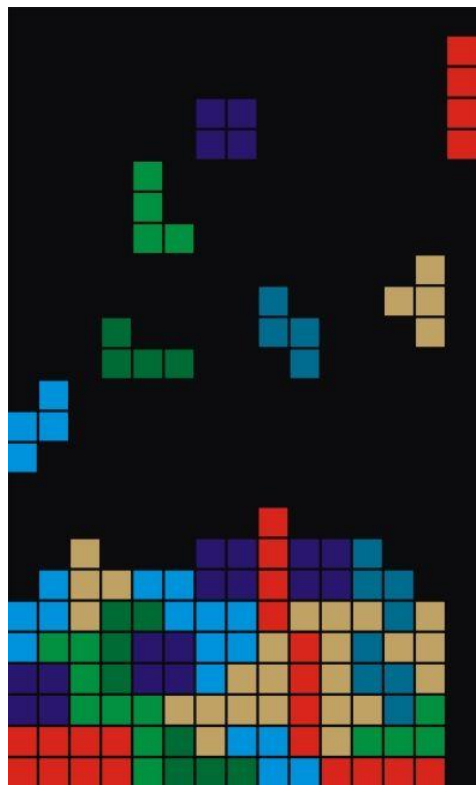
- 软件工程与计算I & 软件工程与计算II
 - 你学会了基本的程序语言（Java）、初步的程序设计以及软件工程方法
- **软件工程与计算III**
 - 团队的项目让你们体会项目中多次迭代所带来的坑
- 高年级的专业课程中，一门一门的帮你们填坑
 - 软件需求工程
 - 软件构造
 - 软件测试
 - 人机交互
 - 软件过程
 - 数据库开发技术
 -



Part 2 “产品化” 的诉求

同理心：满足、愉悦、不爽

- 用户无法像产品经理那样，分层次说出体验，他能展现的就是用户的情绪



同理心：满足、愉悦、不爽

- 这次课程实践：**面向深度学习的在线标注系统**
- **标注工人 (Worker)**，他的愉悦和不爽在哪里？
 - 是否操作简便（交互）
 - 是否等级判定（质量）
 -
- **发布图片任务者**，他的愉悦和不爽又在哪里？
 - 是否有人能完成任务（奖励）
 - 是否质量成本成比例（成本）
 -

同理心的延伸：愤怒和恐惧

- 如果要自己设计一个产品，而不是完成命题作业，应该考虑什么情绪呢？
- **愤怒和恐惧**，他们是同生的一对感觉，都是来自于“被侵犯”。
 - 恐惧是边界
 - 恐惧也是动力
- **恐惧**，才是产品的“痛点”
 - 要么做一个让人**愉悦到爽爆**的产品，要么做一个可以帮人**抵御恐惧**的产品
 - 很多人会把“难受”当痛点

产品是被动的艺术：潜意识、防御与说服

- 三大产品经理做出了全世界最大的三个社交网络产品，他们分别是：
 - 做出了Facebook的扎克伯格
 - 做出了QQ的马化腾
 - 做出了微信的张小龙
 - 马云呢.....?
- 销售和产品的区别，**意识即防御**
- 好的产品是根本不让用户启动防御，《Don' t Make Me Think》
 - 今日头条、抖音
 - 苹果、小米手机

产品中系统能力提供的确定性

- 整体流程与确定性
 - 打孔机的例子：我们生产一个打孔机，用户需要的是这个吗？
 - ATM机的例子：ATM机提供了什么服务？取现金这个确定性的服务
 - 包含战略、运营、现金、密码、硬件、客服、技术
- 我不在乎界面的美观，我在乎的是持续提供用户可依赖的确定性
- 迭代三的检查，在你们认为重要的可展开的服务上，如何提供可依赖的确定性，以及围绕这个确定性你们所展开的各个角度的设计和实现。
- 这就是，项目要求的 **“在某个点上做的足够的深入”**

产品的上瘾：激励机制

- 在确定性基础上的不确定性是产品上瘾的核心要素
- 不确定性的爽点和确定性的爽点合在一起就是用户的体验地图、情绪曲线
- 怎么激励用户
 - 打飞机的例子
- 内在激励和外在激励
 - 内在激励，就是能让人找到感觉，持续深入，把事情做到位
 - 外在激励，非常强有力的短期改变用户行为的工具
- 优化激励系统
 - 受激励点（王者荣耀的越塔、MVP、超神、五杀的设计，勋章、小红花、奖杯等等）
 - 被激励能力，定义获得能力并努力降低门槛（王者荣耀的技能、时间、设备、每日任务，减低动作门槛、降低设备门槛、思考门槛、时间门槛）

产品的迭代：核心功能的深入

- 迭代中，核心功能最重要
 - 微信1.0，熟人之间的发文本信息和图片工具
 - 微信2.0，增加了微信语音和查找附近的陌生人
 - 微信3.0，扫一扫、服务号（从工具开始连接这个世界）
 - 微信4.0，朋友圈（正式变成社交平台）
 - 之前3.5加了扫描二维码、3.6加入了微信公众号、回复评论的功能是4.2版本才做出来的
 - 微信5.0，红包和绑定了银行卡
- 课程实践的迭代也是核心功能的迭代
 - 迭代一：图片标注
 - 迭代二：发布任务和任务管理
 - 迭代三：开放式功能（各组要明确核心功能）

产品的用户体验地图

- 用户体验地图就是通过画一张图，用一种讲故事的方式，从一个特定用户角度出发，记录下他与产品或者服务进行接触、进入、互动的完整过程
- 不要用一个管理员的视角来规划你的产品，而用用户使用的路径
 - 一个画像完整的人物角色
 - 清晰描述用户的目标和预期
 - 服务触点
 - 用户使用路径
 - 用户情绪曲线

研发产品的路径

- 系统的学习
 - 我们都是站在巨人的肩膀上.....
 - 寻找开源代码是我们首先要学会的基本技能
- 细节技术的讨论
 - 服务器和UI的交互
 - 应用服务器和数据服务器的交互
 - 等等

实践项目的分析

- **自动化标注**

- 视频的自动化、实体的自动化
- 目标是如何简单的提供工人的效率

- **标注的质量评估**

- 民主的标注质量评估、带权重的标注质量评估、采样的标注质量评估.....
- 目标是如何提高发布者的资金效率

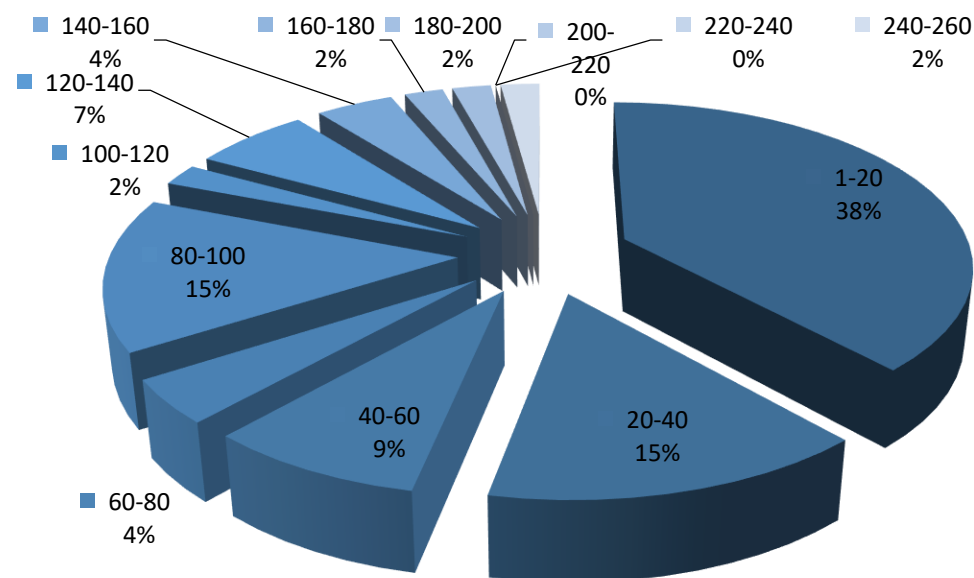
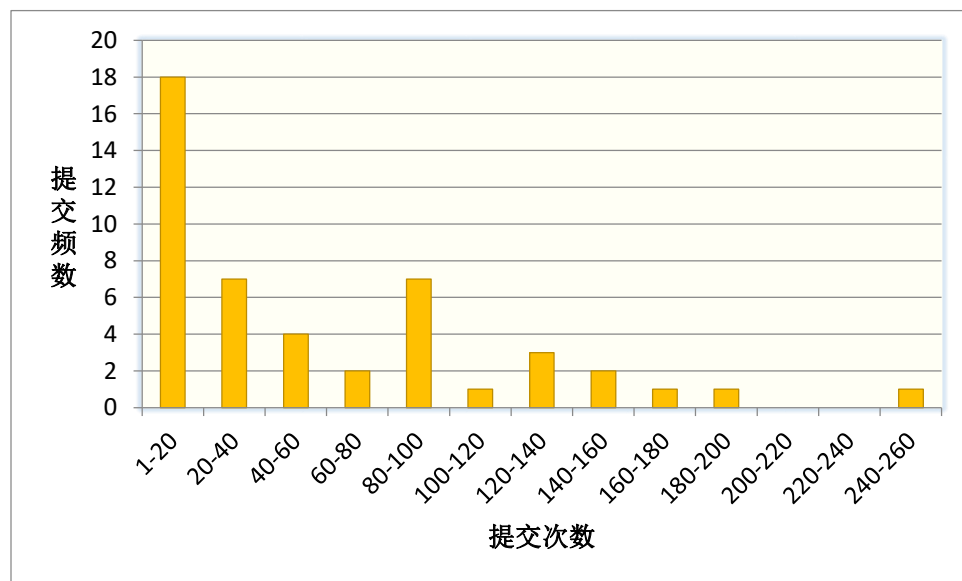
- **标注的推荐**

- 推荐算法的基本逻辑，如何选择更好的算法、如何自动化选择（A/B测试框架）
- 目标是如何提高平台的任务撮合效率

Part 3 期末检查的要求

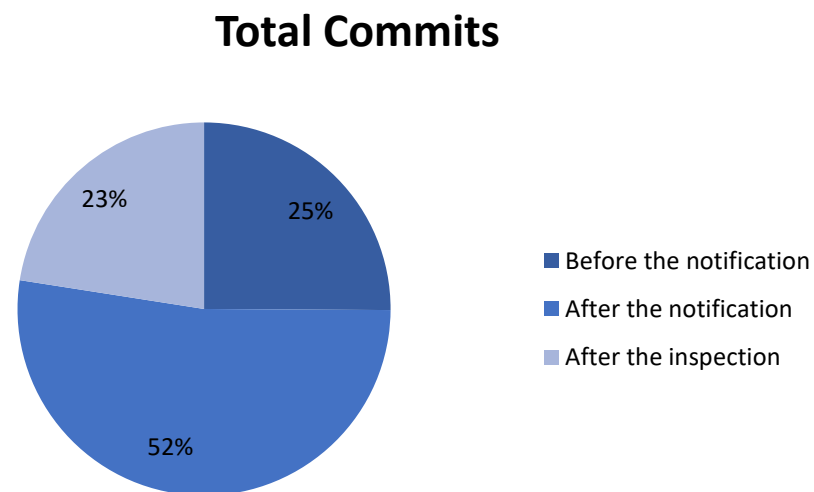
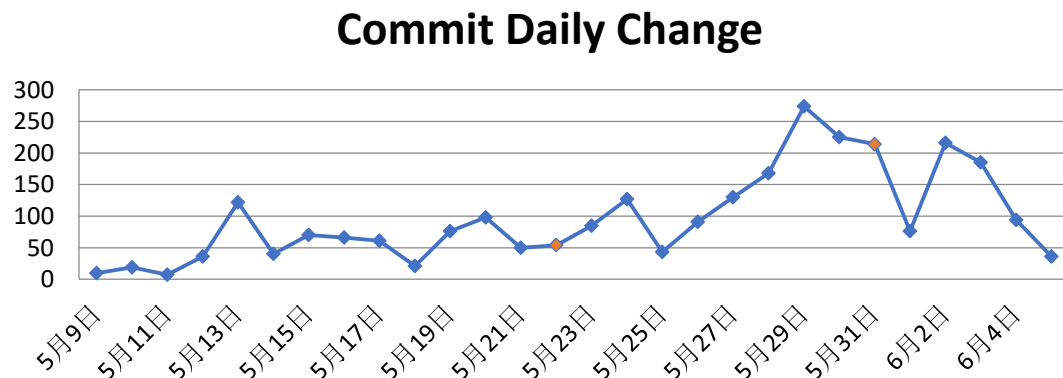
迭代三目前的情况

- 共52组，5月9日（迭代二提交截止日）至今有提交记录的一共有47组
 - 数据截止收集时间为：2018/6/5 15:45
 - 有提交记录的小组中，大部分小组（53%）提交次数没有超过40次。仅有9组提交数超过100次。



Git上提交日的变化情况

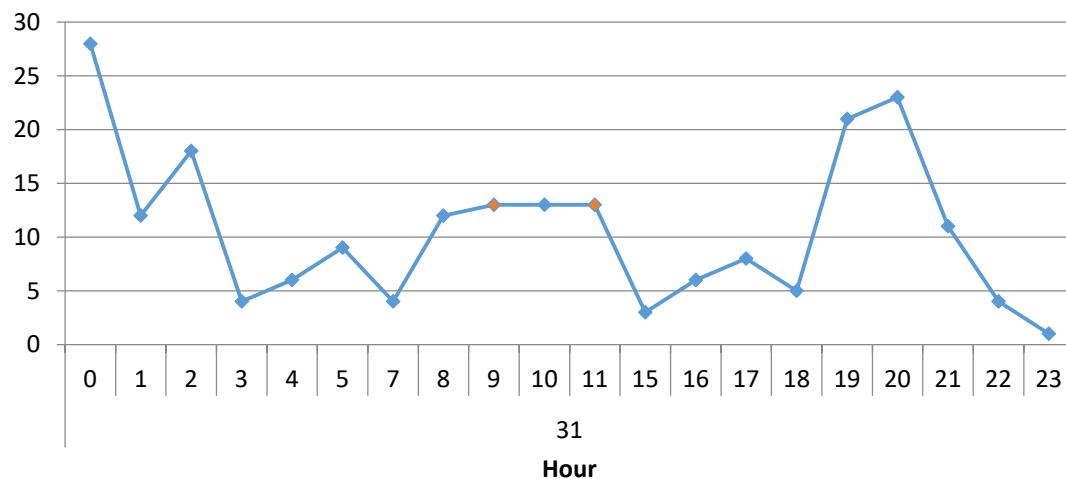
- 收到中期检查消息后，提交量急剧攀升，在接到通知到检查当天，提交量占了28天所有提交量的一半以上。下一节，关注检查日当天各个时段的提交变化。
 - 5月22日公布迭代三中期检查。5月31日开始迭代三中期检查



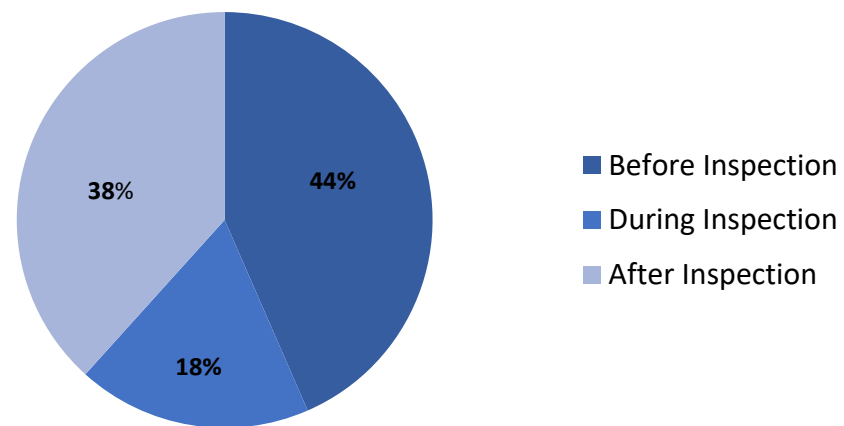
Git上中期检查日的提交情况

- 在检查日当天，在检查前有接近一半的提交，检查前和检查时间段内（50%的时间）的提交占了当天的提交量的62%。
 - 检查时间段为：5/31 09:00-5/31 12:00

Inspection day Commits Changes



Inspection Day Commits Count



Deadline驱动是永恒的驱动逻辑

检查时间安排

- 时间：预计在第16周（端午节后）详细见Moodle通知
- 地点：基础实验楼丙区某一教室（例如：524）
- 人物：全组组员全部到场（不到的组员不评分 & 整组总评扣10分）
- 事件：
 - 每组检查评价时间为30分钟
 - 需要准备ppt陈述、各迭代代码和部署好的系统展示
 - 做好答疑准备，尽可能展示亮点与特点
 - 请学习并测试投影仪连接.....

演示过程要求

Step1

Step2

ABCD

项目介绍和整体演示

- 1、项目整体介绍 (PPT, 4分钟以内)
- 2、项目整体演示 (系统演示, 整体功能, 4分钟以内)

AB

功能

- 1、项目核心和特点演示 (系统演示, 10分钟以内)
- 2、项目使用
(多准备一台部署好的电脑供使用检查)
- 3、答疑

C

代码

- 1、项目代码结构说明 (电脑演示, 8分钟以内)
- 2、答疑

D

过程

- 1、项目软件工程过程说明 (口头说明, 8分钟以内)
- 2、答疑

代码检查的额外提交要求

- 请各组同学将迭代一、迭代二、**迭代三**的源代码按照Gitlab的标准提交
- 源代码仅包含项目目录中的.java文件，请不要上传其他文件。
- 如果之前迭代的代码已被覆盖，可以按照时间点回滚。
- **截止日期为6月18日**（好吧，祝大家端午节快乐！）。

评分

- 项目介绍和整体功能 20%
- 核心功能和细节演示 20%
- 系统使用的友好稳定 20%
- 代码结构合理和清晰 20%
- 软件工程过程的评价 20%

评分

- 小组各个成员统一得分
- 检查期间，小组可以提出对小组每个成员分别评分
 - 分别评分将考察每个成员负责的程序和工作量部分以及参加汇报的表现
 - 只在答辩现场接受评分类型的请求
- **下述情况不给予成绩**
 - **成员不到场**
 - **迟到5分钟以上（不含5分钟）**
 - **无法演示**
 - **源代码没按要求上传的，代码检查部分不予评分。**

祝各位同学期末检查顺利！