

# 【Odometer】第三次会议

---

会议主题: Project's Requirement Analysis

参会人: 卢逸凡、沈祺昊、赵伊蕾, 王柯棣, 梁可愉

会前必读: 项目需求文档, Lec 4

## 议程1: 项目需求

- 讨论过程
  - 我们将项目需求大致分为3个目标实现
    - 对Pytorch项目的stargazer、committer、issue 人数 company 信息的增强数据展示
    - 对项目讨论话题进行展示分析, 其中话题来自于pull request、issue内容。
    - 横向对比多个项目的多项指标
  - 开发计划
    - 针对上述3个项目需求, 我们制定了开发的顺序和优先级。我们将首先完成stargazer、committer、issue人数的company这三个信息的数据可视化工作, 然后再做多个项目的横向对比, 最后完成第二部分话题展示需求。
  - 后端分工
    - 每人领了项目目标1的一块数据内容, 研究github接口, 以及数据如何获取。

## 议程2: 需求访谈会

- 讨论工作划分
  - 访谈人员
    - 卢逸凡
  - 准备访谈脚本和问题
    - 梁可愉、沈祺昊
  - 准备讨论的用例
    - 王柯棣
  - 访谈会记录员
    - 赵伊蕾
- 文档划分
  - 概述
    - 时间安排、人员安排、自我介绍、访谈目的等。
  - 评估问题
  - 对目前情况做简要回顾
  - 客户评估解决方案、确认需要实现的需求
    - 在原有功能基础之上增强数据分析功能, 可以统计显示一段时间内不同贡献者的代码提交、指出项目的核心贡献者。(目前工作重心)
    - 展示被分析项目贡献者的活跃情况和社区发展速度

- 计算项目的 stargazer, committer, issue 人数的 company 信息，并且用图表形式直观展示项目开发者的组织、公司。（目前工作重心）
- 增加对项目issue内容的分析，例如分析issue数量按照更新时间的变化、特征关键词的提取、添加用户自定义条件分析等。
- 增加横向比较多个项目的功能，支持图表和数据的时间范围选择、缩放功能、排序功能
- 实现针对数据源的缓存、优化从 GitHub 上获取信息的方式，在 GitHub 接口不可用时仍能提供项目历史数据访问。同时细化数据过滤，使得对缺失的数据有所补全，提高用户使用的友好程度。
- 客户对待开发系统进行评估
  - 可靠性评估、相应率、性能需求
- 其他需求
  - 存在特殊的许可要求吗？（比如public的项目都可以被import）
  - 对系统的UI界面是否还有非功能性方面的要求或设想？
- 总结
  - 记录员现场完成的记录

第三次会议是以线上、线下相结合的讨论形式

## 线上会议、讨论截图



