

**wink开发文档**

新闻分享app开发文档



**指导教师：**

**杜庆峰**

**小组成员：**

**1752026 李航**

**1652977 王星洲**

**1650350 乔宇**

**1753499 潘小逸**

目录

[修订历史 2](#_Toc29339911)

[1. 介绍 3](#_Toc29339912)

[1.1 目的 3](#_Toc29339913)

[1.2 软件团队 3](#_Toc29339914)

[2. 开发具体内容 3](#_Toc29339915)

[2.1 敏捷开发 3](#_Toc29339916)

[2.2 需求分析 4](#_Toc29339917)

[2.3 前端开发过程 4](#_Toc29339918)

[2.4 后端开发过程 5](#_Toc29339919)

[3. 软件过程管理工具 5](#_Toc29339920)

[4. 版本控制工具 7](#_Toc29339921)

# 修订历史

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 | 评审时间 | 评审人员 | 批准日期 | 签字人员 |
| 2019/11/15 | wink | 1.0.01115\_alpha | 初稿，第一次Scrum例会 | 王星洲 | 2019/11/15 | 李航 | 2019/11/15 | 李航 |
| 2019/11/26 | wink | 1.0.11126\_alpha | 搭建前端框架完成 | 李航 | 2019/12/03 | 王星洲 | 2019/12/03 | 王星洲 |
| 2019/12/03 | wink | 1.1.01203\_alpha | sprint1冲刺完成 | 王星洲 | 2019/12/03 | 李航 | 2019/12/03 | 李航 |
| 2019/12/10 | wink | 1.121210\_alpha | 需求大变更，代码重构 | 王星洲 | 2019/12/10 | 李航 | 2019/12/10 | 李航 |
| 2019/12/17 | wink | 1.2.01217\_alpha | sprint2完成 | 王星洲 | 2019/12/18 | 李航 | 2019/12/18 | 李航 |
| 2019/12/23 | wink | 1.4.51223\_alpha | sprint3完成，需求变更，代码重构 | 潘小逸 | 2019/12/24 | 李航 | 2019/12/24 | 李航 |
| 2019/12/27 | wink | 2.0.01227\_release | sprint4完成，集成测试完成，发行版本 | 李航 | 2019/12/27 | 潘小逸 | 2019/12/27 | 潘小逸 |
| 2020/01/07 | wink | 2.0.30107\_release | 修改部分bug，最终版本 | 王星洲 | 2020/01/07 | 李航 | 2020/01/07 | 李航 |

# 1. 介绍

## 1.1 目的

本文档是wink软件的开发文档，旨在说明开发软件的过程中采用的软件工程方法，总结涉及到的软件工程知识。在本文档中，我们会说明随着开发进程，我们所采用的开发模式、开发框架、会议内容、版本迭代、使用的软件管理工具、版本控制工具、软件过程等一系列内容。

## 1.2 软件团队

本项目中我们的软件团队共有四人：李航（组长）、王星洲、乔宇、潘小逸，我们团队的每一名成员都有个人责任感和优秀的抗压能力。我们都有强烈的目标意识，参与意识，每个人都在该项目中发挥自己擅长的技能、做出了贡献。

由于我们的团队只有四名成员，管理、沟通起来均较为容易，我们采用开放模式作为组织模式，注重管理与创新共同发展，同时，我们是一个敏捷团队，每个人实力较为平均，且能胜任全栈工作，这样，在框架的选择、结对编程各个过程中，我们具有充分的灵活性。同时，我们具备及时通讯工具，保证了实时通讯。我们的团队必将胜任这个项目。

# 2. 开发具体内容

## 2.1 敏捷开发

在小组的第一次讨论中，我们确定了项目的开发过程采用敏捷开发，我们融合了XP（极限编程）过程模型以及Scrum过程模型，并采用软件管理工具Leangoo（领歌）作为软件过程工具（具体介绍在后文提及）。在XP编程中，我们吸取了解对编程，单元测试，持续集成，用户故事，迭代计划等思想，同时，我们采用Scrum过程模型的backlog（待定项）、Scrum例会（由于学业繁忙、我们只能保证一周有一次到两次例会）、sprint（冲刺）的思想作为我们开发过程中的指引。团队领导由李航担任。

这样做的好处是，作为4人小组，前后端分离开发的同时，我们仍然可以采用结对编程的方法来控制代码的质量，并在每一个模块开发完毕后及时进行单元测试。采用用户故事也便于我们分析需求。同时，Scrum的过程模型可以为我们的开发过程给予帮助，我们可以根据backlog以及sprint制定短期内的开发目标，同时使用Scrum例会可以保证每一位组员保持参与意识和贡献意识，这将对我们的开发提供极大的帮助。

## 2.2 需求分析

由于我们的敏捷开发过程模型，我们的需求分析是迭代进行的，通过用户故事来进行需求分析迭代。以下为通过Scrum例会我们分析得出的一些需求：（具体需求请见需求规约）

在第一次例会中，我们讨论到了市场上的新闻类app，这是一个极大的市场，其中有一些出类拔萃的软件拥有超高的市场占有率。

李航作为会议的主持人，由于我们的讨论主要集中在新闻类软件项目之中，他希望我们对该类应用分析痛点并思考市面上存在的软件有什么缺陷或者缺少细分行业龙头。我们以今日头条、腾讯新闻等软件作为参考，发现他们的新闻囊括极广、几乎无懈可击，作为侦查员（分析市场，并弄清潜在对手，收集信息）的王星洲提出了一个痛点，如今市面上的新闻app大多含有很多无用新闻，同时，由于更新过快，用户很难浏览到相同新闻，新闻下方的评论由于用户的意识形态不同变得很糟糕（乌烟瘴气），为后台管理人员带来极大困扰。于是，针对这一痛点，我们选择吸取另一类软件，社交平台（如微博、twitter等）的有点，采用动态的方式对新闻进行评论，并允许用户之间建立好友关系，这样就可以解决上述的问题，并且通过用户之间的联系增加软件的下载量和用户人数。

在细化该需求，并建立需求模型的阶段，我们首先确定了该项目应该是一个手机app，因为新闻应该易于随时随地浏览，同时，我们将需求细化为多个故事：

（1）用户应该可以浏览新闻，并且新闻应有分类

（2）用户可以评论新闻

（3）用户可以观看一些用于放松获有趣的视频，日后也可以作为推广的手段或广告植入手段

（4）为了实现发动态等故事，用户需要注册登录

（5）用户可以发表动态，并且用户对于新闻的评论应该同样以动态形式显示

（6）用户可以与其他用户成为好友

（7）用户可以查看自己发过的动态

（8）用户可以像浏览微博一样，浏览所有人的动态，来从中发现潜在的好友

（9）用户可以仅查看自己和好友的动态，由此来与与自己兴趣相同的人沟通

（10）为了便于用户交互，动态应可以评论

//（11）用户应该可以看视频来放松

（11）用户应该可以搜索新闻和动态

最后，我们根据需求设计了用例图，在这一过程中，我们避免了特性偏好、灵活性偏好以及性能偏好，只专注于软件需求和用户故事，并得出了分析结果。

第二次例会中我们选定了软件管理工具Leangoo，并且决定由它进行敏捷模型管理，于是，我们把需求分析及用户故事作为backlog录入了管理工具中，我们通过讨论，取消了对动态进行转发的功能，因为故事中并无此要求，同时，我们把用户间的关系更正为关注、被关注这两种关系，取消了单纯的互为好友制。

## 2.3 前端开发过程

由于在需求分析中，我们确定了该项目应该是一个手机app，我们选择了Android（安卓）端作为前端的开发框架。这样该项目就可以适配安卓手机。在第一次会议中，我们决定由李航和王星洲作为前端开发的前期负责（由于敏捷开发模式，我们的组队都是自发临时组建的，意味着随时可能变动），并且保证结对编程和及时的测试与集成。

当天我们学习了很多市面上以及开源的新闻app的架构并开始了探索，我们先是选用了较老版本的Android SDK（由于担心新版本适配可能会有欠缺，同时老版本拥有更多可参考文档和支持的第三方库），同时在xml中建立了前端的大体框架，包括新闻页、视频页、动态页、个人页。查找了可用的新闻api，建立了http库，并进行了组件设计和调试。（由于结对编程，采用李航的电脑，并基本由李航编写代码，王星洲负责确定代码是否满足测试要求以及是否规范）在开发结束后，我们马上进行了软件测试，主要是针对新闻页面的单元测试，开始测试时效果不错，但后来出现了问题，（可能是由于api有访问限制，也可能是http库的问题）在多次访问后我们无法再获得新的新闻（即页面无法刷新），我们记录了这个问题，但尚未思考出解决方案。

第二次例会结束后，后来的开发中，我们将测试中发现的问题统一保存于leangoo软件管理工具。同时我们选用github作为版本控制工具，我们建立了git库，并保证了代码的同步。我们为开发的便利制定了很多sprint（冲刺）。

在第一次sprint冲刺结束后，我们完成了所有最基本的功能的实现，大约占了全部功能的60%，完成这些功能让我们无比振奋。

我们迎来了第一次大规模的需求变更，首先是关于动态方面和评论方面，我们由于之前的沟通问题，导致前后端开发小组对新闻/动态/评论的概念出现了混淆，当时的规定是(News/comment/comments)，同时对于新闻的评论与动态生成的关系没有达成统一(Comment for news/ Comment)之后又出现新的问题，对于动态的评论(comments for comment)产生了极大的歧义。从此，我们规定，三个单词分别为(news.trend,comment)同时加入评论动态，评论新闻的概念，确定了评论新闻等同于发送一条动态。并以此重构代码，进行第二轮冲刺。

二轮冲刺完成了许多功能，单元测试基本通过。

三轮冲刺前我们试着安装了sonarcube等工具对代码质量进行检测。

三轮冲刺结束，程序功能大体上完全实现。

第二次需求变更，我们去掉了之前认为可以使用户放松心情的视频模块，我们将它定义为“非必需的”“与核心功能无关的”需求，取而代之的，我们采用了搜索模块代替了它，并将搜索细化为新闻搜索和动态搜索。

最后一轮冲刺结束，测试结束，完成第一个发布版本。

更改了一些显示上的bug，版本更新到了2.0.30107\_release

## 2.4 后端开发过程

在需求分析过程中，我们逐步明确了本项目后端所提供的功能，首先最基本的是为前端提供新闻。最初，我们讨论了在每次访问时调用API获取新闻，和提前从API获取新闻并存入数据库，每次访问时直接从数据库中获取两种方式，并最终选择了后者。为此我们编写了一个简单的脚本，定期从指定API获取新闻并将获取到的新闻内容写入数据库，以便后续可以直接从数据库中提取新闻返回给前端。其次，项目后端主要是为了与数据库创建连接，并向前端提供所需要展示的数据，同时将前端各个功能修改的数据写入数据库中。因此，我们选用了Springboot框架实现快速开发，由于项目数据库体积较小，我们选择了更为简洁的MySQL数据库。后端编程主要有潘小逸和乔宇负责，依据功能模块分工，使用Github实现代码的同步。同时保证结对编程，即互相对每部分编码完成的功能进行审查。

我们在项目初期，根据数据库设计的结果首先建立了数据库，在后续项目迭代的过程中，逐步对数据库的表设计也进行了迭代，删除了一些多余不必要的表或属性，修改一些不明确的字段名，添加缺少的属性等。

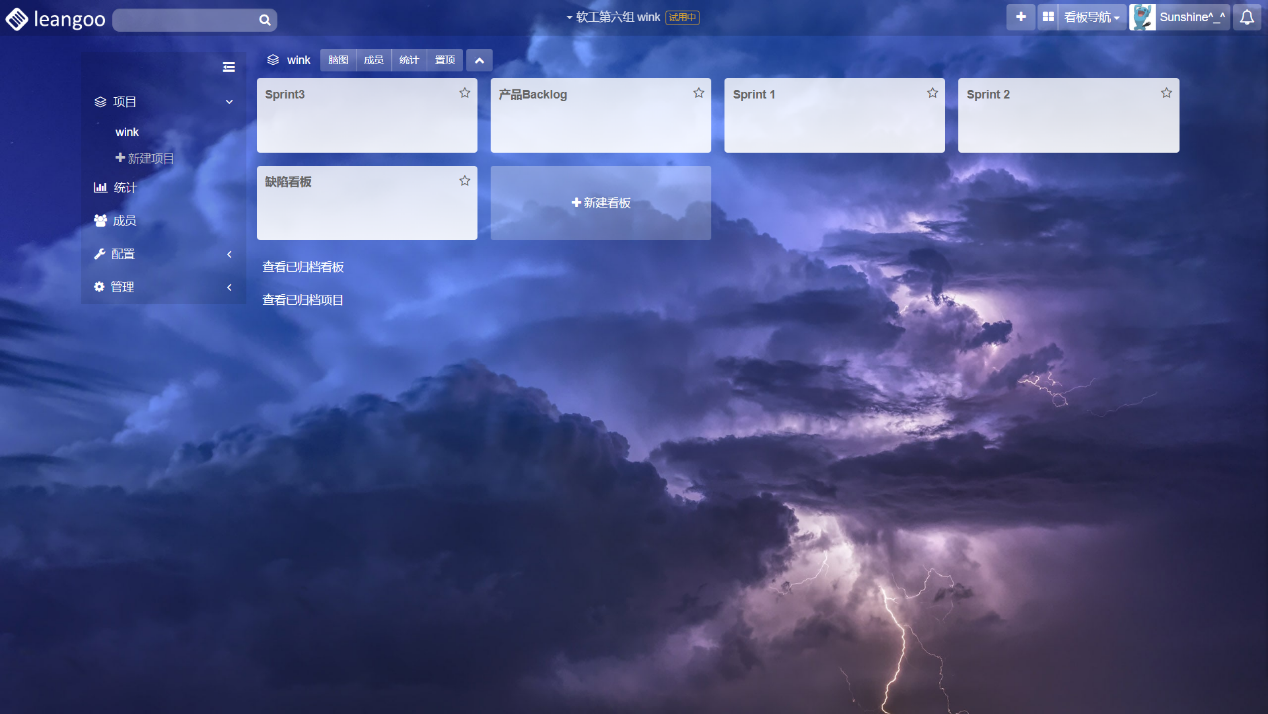
后端代码在开始编写之前，就首先统一了命名规范等问题，有效地保障了后续代码风格的一致性，同时也便于了代码审查工作的展开。针对后端每个实现的功能，我们都先使用Postman进行接口测试，保证接口功能的正确性，然后编写相应的API文档并后续发布。

后端的工作阶段主要是先完成数据库的建立，然后编写接口。由于采用敏捷开发模型，接口的实现依据功能的重要性，从最主要的功能先开始实现，以便前端可以快速地查看到结果，例如我们最先实现了查看新闻的接口。在所有功能实现的编码工作完成后，由潘小逸将所有的代码进行了重构，使后端整体的代码显得更加规整。

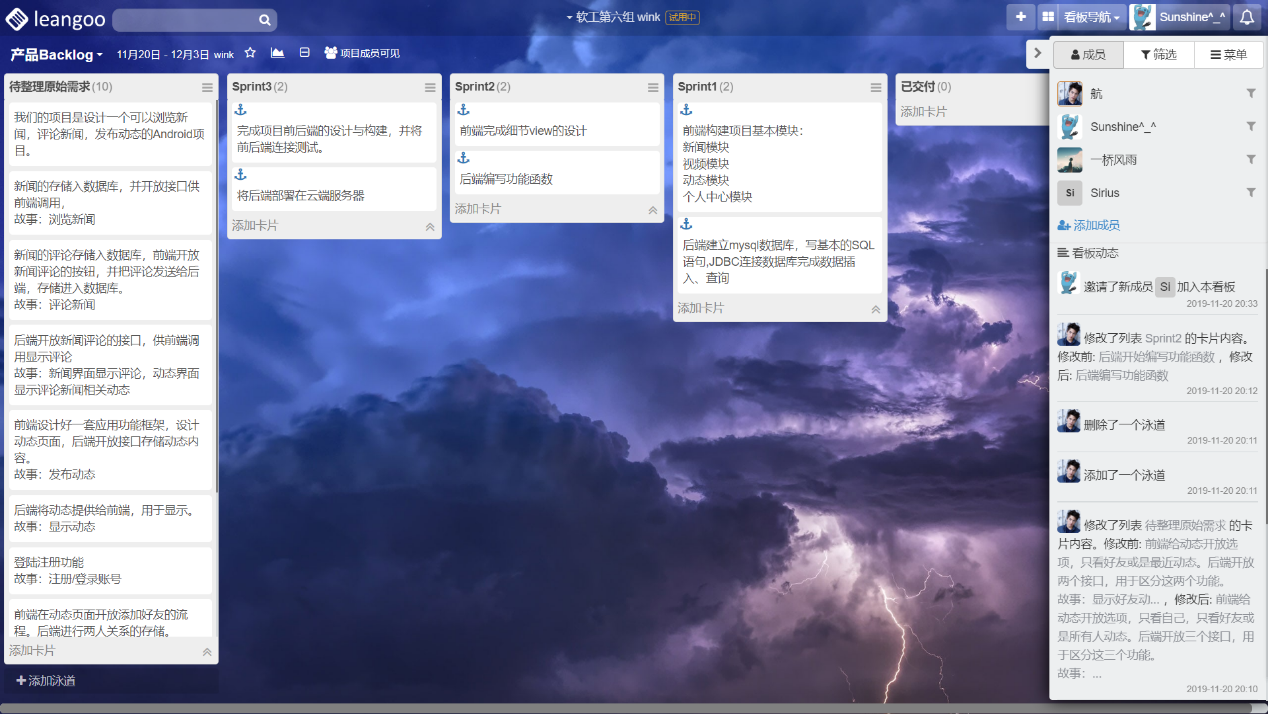
# 3. 软件过程管理工具

Leangoo（领歌）是我们在第二次会议结束后采用的软件过程管理工具，这是一个适用于Scrum过程模式的管理工具。它具有看板功能、评论功能、泳道图功能、sprint记录、backlog记录、缺陷看板功能等核心功能，并且在企业版支持软件过程进度，燃尽图，甘特图等，便于我们进行软件管理。

首先是我们的看板界面：

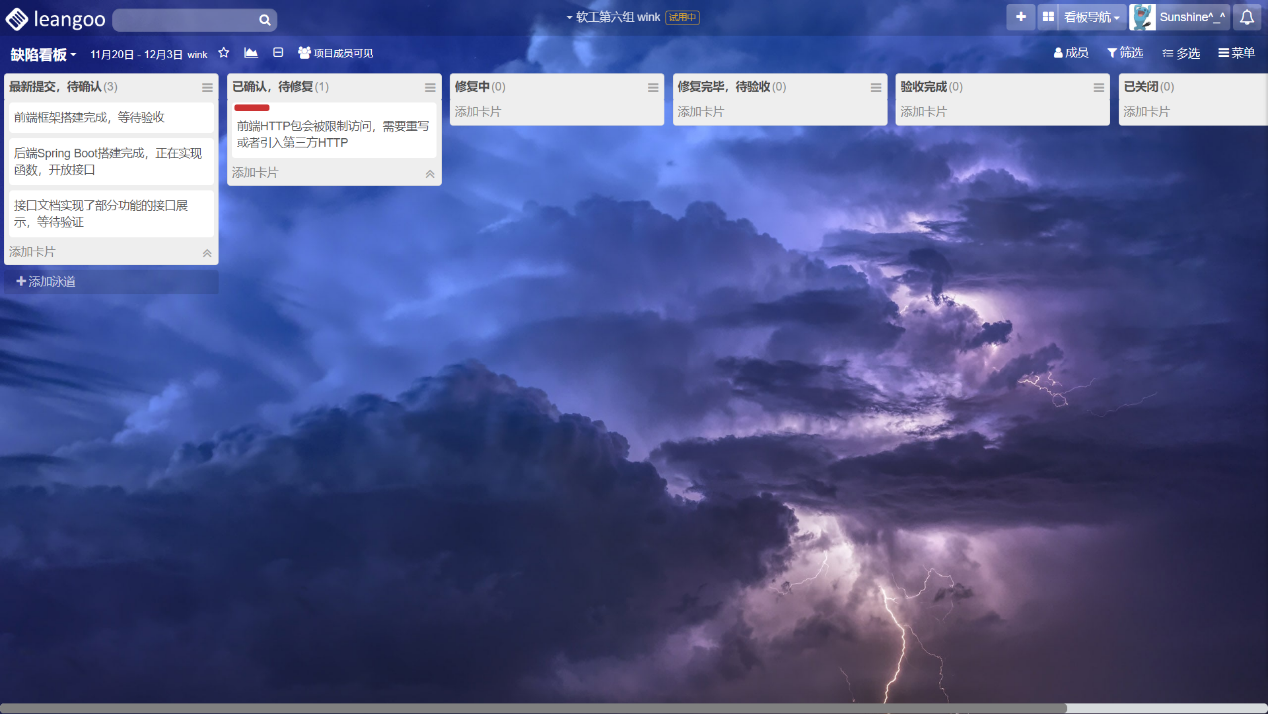


接下来是我们的backlog：

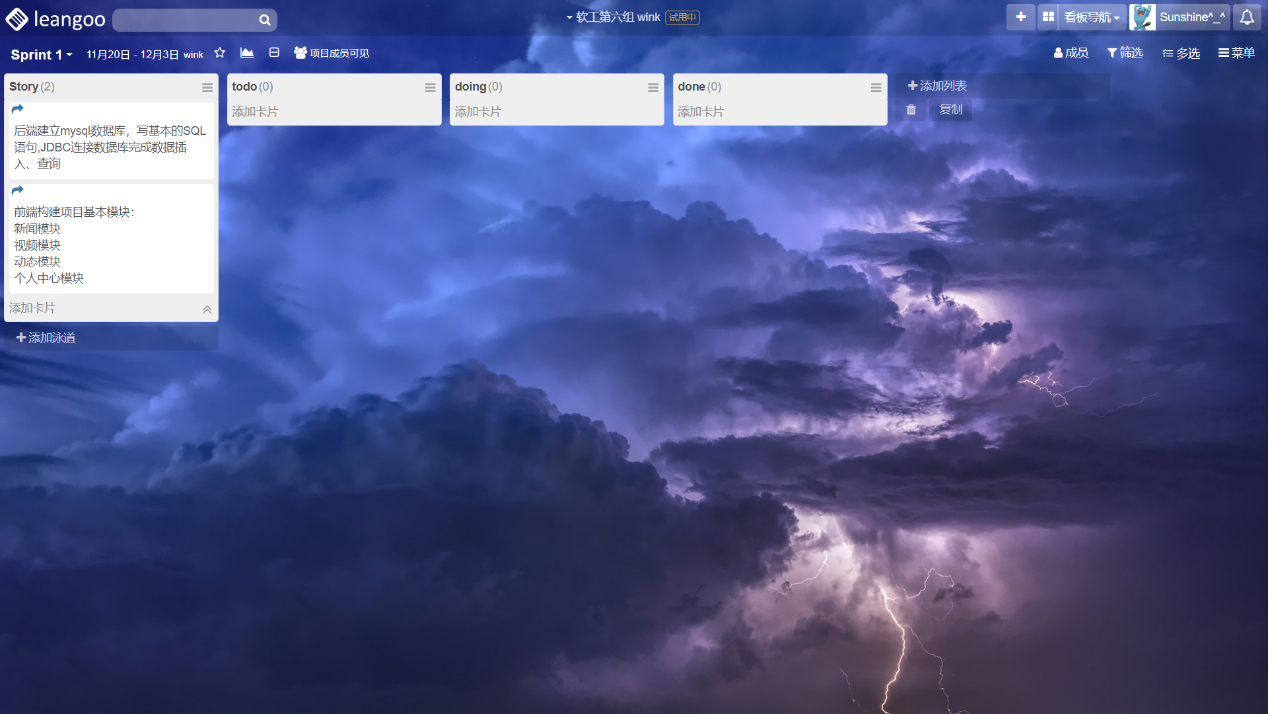


我们可以给成员添加每一个看板的访问权限。

我们的缺陷看板记录测试中发现的问题：



最后是sprint（冲刺），我们会定义冲刺的时间结点，和具体内容：



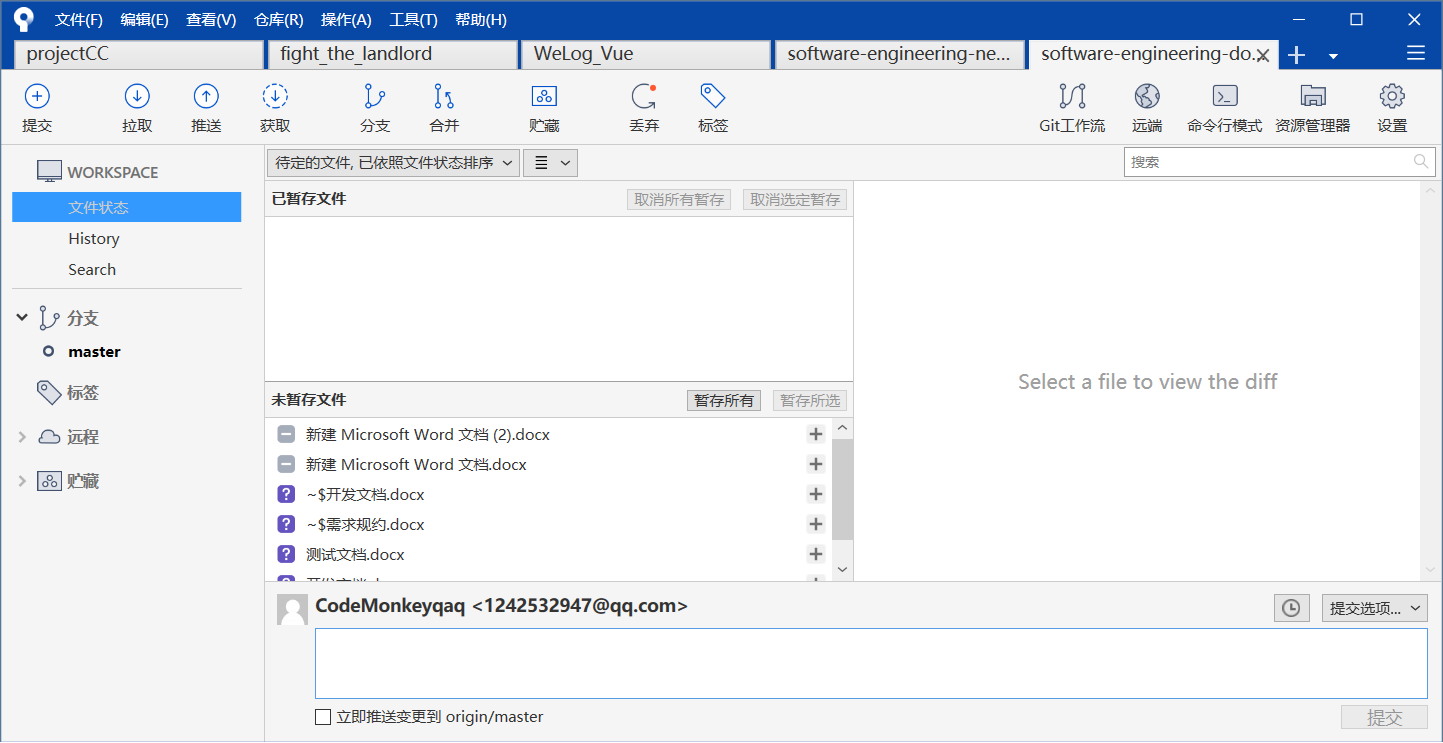
# 4. 版本控制工具

我们选用GitHub作为版本控制工具。我们不仅用git保存代码，还用来保存所有的文档和其他课程相关内容。同时我们使用了Sourcetree作为图形化界面。

我们的前端库位于<https://github.com/whasfuck/software-engineering-newsAPP>

我们的文档库位于<https://github.com/whasfuck/software-engineering-documents>

sourceTree的页面：



接下来我们使用了vs code 和 intellij idea/Android Studio自身集成的git工具来进行版本管理，避免额外安装图形化界面。

