配置管理

1. 实验目标

在软件研发过程中,采用有效方法来记录变更和版本，根据版本的变化来和变更的进行来分析软件研发的进度和预期进行对比，以便调整软件开发的进度，实行变更控制和版本管理。

2. 实验方法和工具

2.1 实验方法

小组成员每次完成自己分配到的任务时,首先利用 github,从 github 上 pull 下来其他小组成员上传的工作内容，以完成不同小组成员的信息的同步。在 pull 的过程中，可能发现不同小组成员对同一文件的修改而造成的冲突，这时需要手动更改冲突的部分，完成小组成员工作内容的合并。接下来，小组成员把自己的工作内容 push 到 github 上，以让其他小组成员可以及时 pull 下自己的工作内容，从而完成变更控制。

2.2 实验工具

本实验选取 Github 作为实验工具。

1.会议记录	更新github目录结构	3 months ago
10.实验6-8	实验6-8记录更新	a month ago
11.测试需求说明书	更新	20 days ago
12.测试分析报告	更新	20 days ago
13.实验验收文档	验收文档添加mpp文件	19 days ago
14.软工实验个人总结	软工实验个人总结	19 days ago
2.开发计划书	系统需求规格说明书	4 months ago
3.项目进度计划	更新mpp	19 days ago
4.文档撰写记录	更新需求规格说明书	3 months ago
5.作业提交	整理实验文档	3 months ago
6.项目资源	上周文档同步更新	2 months ago
7.演讲PPT	软工实验个人总结	19 days ago
8.需求规格说明书	软工实验个人总结	19 days ago
9.评审报告及问题清单	14.实验验收文档更新	27 days ago
.gitignore	14.实验验收文档更新	27 days ago
README.md	readme	4 months ago

3. 实验要点

3.1 变更控制

每次要进行代码开发的时候，先与 github 服务器同步，保持一致，在开发完毕的时候，再与 github 服务器同步，更新服务器数据，这两步必须做，以免出现任何异常。每次分配任务时，将任务分解成不冲突的部分，这样每个人可以单独的完成自己所负责的部分。如果任务适合细分，那么每个人各自提交自己所负责的部分。如果任务较大，那么两个人或多个人负责同一个部分并一起协作。尽量不出现多个人共同编辑单个文件的情况。如果多个人编辑同一个文件，那么事先协调好编辑的时间，两个人不同时编辑一个文件。

3.2 版本管理

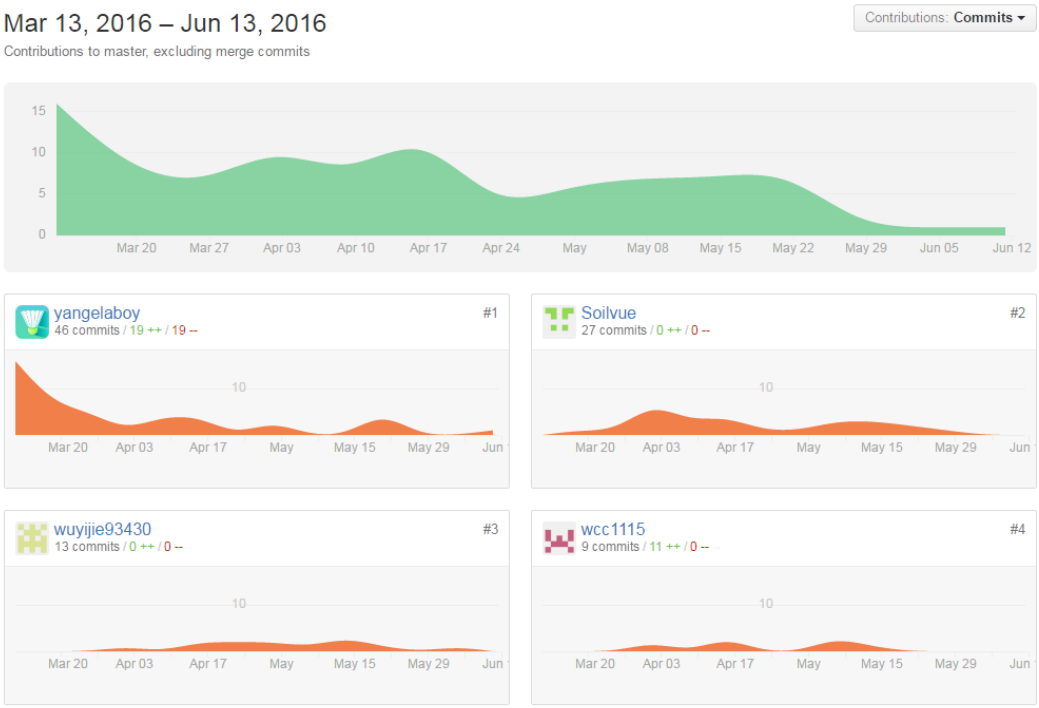
首先我们将项目分为数个模块，分别放在不同文件夹中：会议记录、源代码、开发计划书、问题清单、问题清单等等，每种类型的产出文件单独建立文件夹，避免所有文件在同一文件夹，方便查找和管理。对于每个重要的项目模块，比如计划书，需求分析之类的，每次修改我们都赋予其一个版本号，并记录版本号，修改过的版本都会保留。

3.3 变更管理分析

经过项目变更管理，项目资料和文档可以方便的检索和管理。并且，项目的提交从未出现过冲突的情况，基本完成了变更管理的目的。哪项修改由谁做出都能方便的看到。最后，代码开发过程，项目的工作进度也一目了然。

4. 实验结果

下图是本组对文档提交的历史记录，由于 github 有时候会出现上传失败的问题，所以我们组成员有时会将完成的文档交给其他成员代为提交，所以各成员的总的计划工作量较为均衡，实际完成情况也基本与计划相符合。对于文档提交共有 113 个。可以看到，组员的提交时间都呈波浪形，在个别天数比较集中，这与本课程设置的提交日期高度相关。



Github 变更记录文档附在另一个文档中。