**GitHub管理方法的改进计划**

* **文档组织结构：**

由于GitHub平台是一个代码托管平台，它提供了代码追踪机制，但是对于文档之类的非代码类文件，GitHub并不能明确快捷地让用户来进行追踪。所以，对于文档的分类组织一定要有一个清晰合理的组织结构。通过总结半学期以来对GitHub的使用，针对本课程的要求，我们对文档组织结构的建议如下：

|\_\_**会议记录**

| |\_\_\_\_\_\_课堂评审

| |\_\_\_\_\_\_日常讨论

|

|\_\_**项目文档**

| |\_\_\_\_\_\_文档正式版

| | |\_\_\_\_\_\_第一周(备注)

| | |\_\_\_\_\_\_第二周(备注)

| | ……

| |\_\_\_\_\_\_文档

| |\_\_\_\_\_\_第一周(备注)

| |\_\_\_\_\_\_第二周(备注)

| ……

|\_\_**参考资料**

| |\_\_\_\_\_\_项目相关

| |

| |\_\_\_\_\_\_软件工程

|

|\_\_**项目代码**

|

|\_\_**实验**

|

|\_\_\_\_\_\_实验一

|

|\_\_\_\_\_\_实验二

以上结构图列出的均是目录，具体文件可根据所属类别组织在该结构图里，命名方法见下文。

* **命名方法：**

由于GitHub在文档管理方面的不足，所以对文档的命名也要尽可能地清晰规范。

对于代码文件，可以根据不做要求，但是对于文档文件需要有着统一并且让组员知晓的约定。

* + 对于正在更新中的文档，可以在文件名中增加‘-更新’来表明其状态

例如：正在更新中的《需求规格说明书.docx》可命名为《需求规格说明书-更新.docx》

* + 对于已经确定最终版的文档，可在其后添加’-最终版’，并将其放在项目文档中的文档正式版文件夹中。例：《需求规格说明书-最终版.docx》
  + 文档命名应准确清晰，并在末尾附上主要负责人的姓名
* **更新说明**更新说明应该附上具体周数，commit人姓名，更新文件，更新说明。
  + 例子：3-张三-需求规格说明书-添加非功能需求
  + 这种命名方法鼓励组员没修改完成一个文件立刻进行提交。
* **一点建议**

GitHub的README是介绍展示一个项目的很重要的文档，但是本课程中同学们都没有利用起来，建议每组同学都写一个README,利用很简单的markdown语法，便能写出很整洁的README,建议每组同学都添加一下关于本课题的大概内容的README，方便老师与别的同学进行查阅。