

# 变更控制和版本管理方案

## 1. 实验目的

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. 实验方法与工具

### 2.1 方法

#### (1) 变更控制

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。(如下图)



9-实验六方案 ...		
sswang1117 authored 7 hours ago		latest commit 8a816d62c7
1.会议记录	添加第一次会议记录	2 months ago
2.评审意见	9-评审意见记录	2 days ago
3.项目PPT	7-课堂演示ppt	21 days ago
4.项目提交文档	9-实验六方案	7 hours ago
5.资料	5-Redis源码	28 days ago
6.检查单	Revert "4-需求分析检查单的修改"	a month ago
7.工作日志	9-工作日志更新	9 hours ago
8.工作量统计	9	12 days ago
Team-A 评审材料	10	3 days ago
README.md	Initial commit	2 months ago

## （2）版本管理

在每一个项目文件中，我们都会在文件头部加入版本变更历史的表格，每一个成员在完成自己的部分后，都要在表格内填入版本变更的信息，包括版本号，修改人和时间等，方便对历史版本的查看（如下图）。同时，小组成员分别各自提交 `github` 也有助于对版本的管理。

版本变更历史

版本	提交日期	编制人	说明
Version 1.0	2015-05-17	陈志伟、林璐	初步完成测试分析报告的第一版
Version2.0	2015-05-19	王志鹏	初步完成测试分析报告的第二版

## 2.2 工具

Github

版本表格

## 3. 实验要点

- 1) 每个成员完成自己的任务后及时将文件 `push` 到 `github`，以便小组成员间的及时同步
- 2) 编写完自己的文件后，不能忘记填入版本变更历史表格

## 4. 实验依据（输入）

输入文件：

版本更新表格，`github` 项目文件

## 5. 实验结果（输出）

变更与管理分析报告