**渔乐生活APP**

**配置管理子计划**



版本号：[0.1.2.20210327]

拟制人：朱邦杰 31801305

刘 哲 31801318

童峻涛 31801341

牛旷野 31803199

徐 任 31801346

审核人：朱邦杰 31801305

批准人： 杨 枨 老 师

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ] 正式发布  [√] 正在修改 | 文件标识： | PRD2021-G10-配置管理子计划 |
| 当前版本： | 0.1.2 |
| 作者： | 朱邦杰、刘哲、童峻涛、徐任、牛旷野 |
| 完成日期： | 2021-03-27 |

文档修订记录

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 版本 | 修订人 | 参与者 | 修订日期 | 修订状态 | 修订说明 | 审批日期 | 审核人 |
| 0.1.0 | 朱邦杰 | 全组人员 | 2021-03-12 | 首次编写 | 初始版本 | 2021-03-12 | 全体组员 |
| 0.1.2 | 朱邦杰 | 全组人员 | 2021-03-13 | 编写 | 修正版本 | 2021-03-13 | 全体组员 |
| 0.1.3 | 刘哲 | 全组人员 | 2021-03-27 | 修订 | 里程碑修订 | 2021-03-27 | 全体组员 |
|  |  |  |  |  |  |  |  |

目录

[1引言 3](#_Toc67744301)

[1.1标识 3](#_Toc67744302)

[1.2系统概述 3](#_Toc67744303)

[1.3文档概述 3](#_Toc67744304)

[2.配置系统管理 4](#_Toc67744305)

[2.1管理 4](#_Toc67744306)

[2.1.1机构 4](#_Toc67744307)

[2.1.2组织及组成成员职责 4](#_Toc67744308)

[2.1.3角色职责 4](#_Toc67744309)

[2.2项目成员分支操作权限 4](#_Toc67744310)

[2.3辅助用于配置管理的软硬件资源 5](#_Toc67744311)

[2.4 配置目录管理 5](#_Toc67744312)

[2.5 变更控制 5](#_Toc67744313)

[2.5.1执行条件： 5](#_Toc67744314)

[2.5.2执行流图： 5](#_Toc67744315)

[2.5.3执行角色职责： 5](#_Toc67744316)

[2.5.4配置变更提交规范： 5](#_Toc67744317)

[3.版本命名策略 6](#_Toc67744318)

[3.1版本格式 6](#_Toc67744319)

[3.2版本更新 6](#_Toc67744320)

[3.3文件命名规范 6](#_Toc67744321)

[4.Git使用策略 6](#_Toc67744322)

[4.1基础知识 6](#_Toc67744323)

[4.2注意点 7](#_Toc67744324)

[4.3使用场景 7](#_Toc67744325)

[4.4提交日志书写规则 8](#_Toc67744326)

[4.5图示 9](#_Toc67744327)

# 1引言

## 1.1标识

标题：《配置管理子计划》

当前版本号：0.1.2.20210327

说明：本文档支持Office、WPS等word文档阅读软件。

## 1.2系统概述

1. 项目名称：移动端记录分享服务型App应用程序

2. 项目用途：

制作一款为钓鱼发烧友开发的APP，它将提供一个以地理信息位置为基础的社交平台，含有钓鱼地点的上传、分享钓鱼经历、朋友圈社交、相约钓鱼等功能，该系统将给钓鱼爱好者提供一个良好的交流的平台，不同于当前钓鱼APP系统，我们的产品目前不带有商用、获利目的。将吸引一批钓鱼发烧友进行钓鱼活动、分享。

3. 任务提出者：杨枨老师

4. 项目开发者：朱邦杰、刘哲、童峻涛、徐任、牛旷野

5. 用户：杨枨老师、广大钓鱼发烧友

6. 课程名称：《软件需求分析原理与实践》

7. 承办小组：SRA2021-G10小组

8. 项目里程碑：

2021年03月11日 项目选题完毕

2021年03月14日 可行性分析完毕

2021年03月18日 项目计划完毕

2021年04月11日 第一次需求获取完毕

2021年04月30日 第二次需求获取完毕

2021年05月18日 需求变更完毕

2021年06月01日 UML反转课堂完毕

2021年06月14日 项目收尾完毕

2021年07月01日 项目总结完毕

9. 具体文档：项目的提出和介绍；软件项目计划；软件可行性分析报告；软件需求分析报告；软件总体设计报告；软件详细设计报告；软件测试报告；项目总结报告。

## 1.3文档概述

本配置管理子计划对本次项目中的各类配置进行把控和管理。同时对可能出现的问题做好分析、研究和探讨。本文档的使用除了项目开发团队的内部使用外，还需提交用户和客户组织负责人审查批准。

本文档的使用应遵守国家先关法律法规，未经允许不得对外公开，需要有一定的保密性和私密性。

# 2.配置系统管理

## 2.1管理

### 2.1.1机构

TBD

### 2.1.2组织及组成成员职责

#### 2.1.2.1软件配置控制委员会（SCCB）

会长：朱邦杰

会员：徐任

#### 2.1.2.2软件配置管理组（SCM组）

组长：朱邦杰

组员：徐任

### 2.1.3角色职责

#### 2.1.3.1软件配置管理员，软件变更控制管理员

软件配置管理员：徐任

软件变更控制管理员：朱邦杰

#### 2.1.3.2开发人员

开发人员：刘哲、牛旷野、童峻涛

#### 2.1.3.3测试人员

测试人员：刘哲、牛旷野、童峻涛

## 2.2项目成员分支操作权限

SRA-G10(非受控)：

master:朱邦杰、刘哲、牛旷野、童峻涛、徐任

SRA-G10：

master:朱邦杰

UML作业：朱邦杰、刘哲、牛旷野、童峻涛、徐任

UML基础知识测试：朱邦杰、刘哲、牛旷野、童峻涛、徐任

软件需求变更文档：朱邦杰、刘哲、牛旷野、童峻涛、徐任

项目计划：朱邦杰、刘哲、牛旷野、童峻涛、徐任

项目总结：朱邦杰、刘哲、牛旷野、童峻涛、徐任

## 2.3辅助用于配置管理的软硬件资源

软件：GitHub Desktop

硬件：笔记本电脑

## 2.4 配置目录管理

TBD

## 2.5 变更控制

### 2.5.1执行条件：

一切文档的更新。

### 2.5.2执行流图：

TBD

### 2.5.3执行角色职责：

完成本地和gitee仓库内文档的更新。

### 2.5.4配置变更提交规范：

依照配置管理文档的要求变更文件版本号，并在提交时写明更新内容，首先提交至非受控文档仓库中，经过组长同意后，视情况由对应负责人提交至受控文档仓库所属阶段分支，最后合并至master。

#### 2.5.4.1文档修改：

前文已提及。

#### 2.5.4.2 BUG修补：

TBD

#### 2.5.4.3 新功能提示：

提交时写入更新说明中。

# 3.版本命名策略

## 3.1版本格式

每一个文档的版本格式为[主版本号.子版本号.修正版本号]。

示例：0.1.1

文档的初始版本为0.1.0，正式发布版本为1.0.0。

同时修改详细版本号例如[主版本号.子版本号.修正版本号.日期] 例如 [0.1.1.20210301]

## 3.2版本更新

当文件内容有了重大的变化或改进，主版本号加一。

当文档的内容有了模块的增加、补充、新的阶段等，子版本号加一。

当文档的内容有了小修改，如修正了纰漏等，修正版本号加一。

## 3.3文件命名规范

所以文件命名格式为“SRA-G10-文件名”或“PRD-G10-文件名”。

期中PRD属于阶段性文件，SRA属于项目最终文件。

若文件需要保存迭代则在文件后面添加版本号 v a.b.c,详见3.1，例如SRA-G10-WBS和SRA-G10-WBSv0.1.0。

非受控仓库中同一子版本的分工任务则在当前版本后加“（）”备注注释，若有其他备注也以此形式备注，如分工到的文档初始文件为PRD2021-G10-项目计划PPT，完成1-4章制作后，上传到非受控仓库的文件名为 PRD2021-G10-项目计划PPT（1-4）。

# 4.Git使用策略

## 4.1基础知识

在使用之前，我希望每位组内成员都能明白git的基本用法与术语，在此，我对几个关键术语做出解释，如果不能理解，可以网上搜索资料或者问陈。

* 仓库：可以简单的理解为一个文件夹
* 多版本：如果一个仓库是多版本的，那么我们可以随时把它切换成某个时间段的某个样子，即不同版本。
* 分支：一个文件目前是A状态，甲将这个文件从A状态修改到了B状态，乙将这个文件从A状态修改为了C状态，那么从A这个时间点分叉出了两个不同版本（B、C）,即分支。
* 分支合并：将B、C两个状态相对于A的改动合并到一起。注意，如果B、C对于A都只是增加内容，那他们可以轻易的合并到一起，如果对同一个部分有了修改操作，会造成“冲突”，需要人工合并，应该尽力避免这种情况。
* 远程仓库：即我们放在码市或者github的仓库，对于组员来说是共用的，上面的内容大多数应是可发行的版本（做完的）。
* 本地仓库：就是你自己电脑上从远程仓库克隆下来的文件夹，如果你只是在本地做了修改，是不会影响远程仓库的，其他组员是看不到你做了什么的，除非你push了改动。
* 远程分支：即远程仓库上不同的分支，所拥有的不同版本，对所有组员可用。
* 本地分支：你为自己在本地的仓库建立的分支，你可以选择是否push它，使它成为远程分支。
* push：将本地仓库的改动（包括你建立的本地分支）推送到远程仓库上，使其他组员也能看到你的修改。
* pull：将远程仓库上的内容同步到本地仓库上。
* fetch：可以检测出远程仓库对于你的本地仓库有哪些更新。
* master分支：主分支，上面的所有内容应保证是可用的、可发行的。

## 4.2注意点

* push之前请先fetch，看看远程仓库目前是不是最新版本，如果是的话先pull下来，再push，防止冲突。
* 对于push时，备注应该详细，比如对哪些文件的哪些部分做了何种修改，而不要笼统的说修改了某个文件

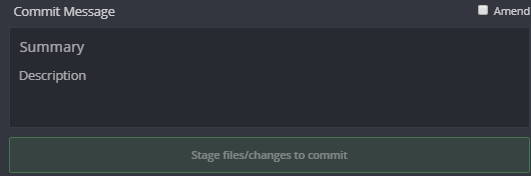
## 4.3使用场景

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 场景 | 权限 | 操作分支 | 所在目录 | 准备工作 |
| 提交个人作业/文档 | 项目组所有成员 | 组员自己的分支 | 非受控文件/组员名 |  |
| 新增文件 | 项目组所有成员 | 组员自己的分支 | 非受控文件/组员名 | 根据项目经理的指示完成文档的编写。 模板由项目经理指定、或者小组协同确定模板。  增加文件的目录是在组员的文件夹内的，在完成起草之后，并入ModifyingMaster分支的时候由配置管理员把文件转入受控文档。 |
| 协同编写文档 | 项目组所有成员 | 组员自己的分支 | 非受控文件/组员名，  该组员是文档的主要编写者 | 和新增文件流程一样，文件的主要编写的组员在自己的工作分支上编写文件，其他编写组员也pull下该分支后协同工作。 |
| 更新现有文件 | 项目组所有成员 | 新分支/ModifyingMaster | 受控文件对应的文件夹。 | 和配置管理员协调，确定基准线，然后开一个新分支。名字格式：“modify\_文档标识”，然后向该分支上提交记录。在完成修改之后，通知配置管理员（给他一个便签啥的）让他合并入ModifyingMaster中，如果想自己合并，记得在合并前拉取最新的ModifyingMaster分支。  请勿在自己的组员文件夹中修改，这样就不能优美的体现出文档改动了。 |
| 提交整合完的文档，里程碑文档 | 配置管理员 | master | 受控文件对应阶段文件夹。 | 在提交前拉取远端的最新master，并以此为基础再提交 |
| 对配置管理系统中的文件命名进行整改 | 配置管理员 | master | 受控文件夹 | 在提交前拉取远端的最新master，并以此为基础再提交 |

* 只有配置管理员有权限新建分支、合并分支。
* 项目最新的进度体现在ModifyingMaster分支上，Master是拿来保存里程碑快照的。

## 4.4提交日志书写规则

提交日志分为Header和Body：



图中的Summary就是Header，Description就是Body。

* **Header的期望格式：**

**Header模板：**

<type>(<scope>):<subject>

**字段的必须性：**

Type:必须填；

Scope：可选填；

Subject：必须填；

**字段的用途：**

Type：说明commit的类别：请使用下面几个标识：

* modify
* add
* delete

Scope：说明commit影响的范围，这是第几周的作业或者是因为评审不通过而改动。

Subject：是commit的简短的描述，直接写改动的文档的标识，PRD\_XXXXXv1.2.3。

**例子：**

*modify(组内评审不通过)配置管理计划v1.2.3*  括号要用()英语中的括号。

或者：

*modify\_配置管理计划v1.2.3*

* **Body的期望格式：**

Body是对本次commit的详细描述，需要分成多行。

**例子：**

*增加封面*

*增加提交日志书写规则*

*修改版本命名规则*

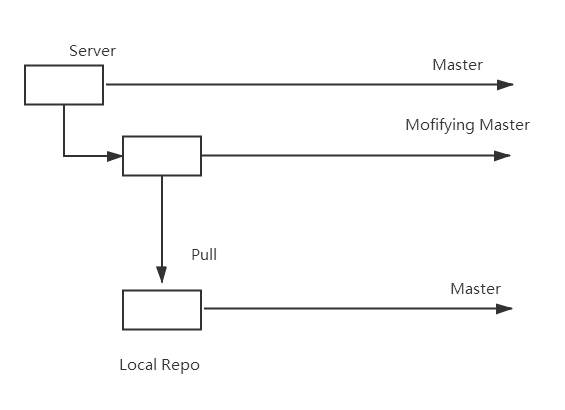
#本条规则参考：

http://www.ruanyifeng.com/blog/2016/01/commit\_message\_change\_log.html

## 4.5图示

小组成员初始化本地仓库：

请先把origin上的modifyingmaster分支pull到本地，获取仓库的基本内容。



小组成员提交文档：

当你做完改动commit到本地的仓库之后，想要把内容push到服务器分享给他人的时候，不要push到ModifyingMaster和Master分支上。

请在服务器上建立一个用自己名字命名的分支，然后push到这个属于你的分支上。

注意：如果你要查看其他组员的工作进度，请pull下对方的分支到你自己的本地仓库中，最好不要乱push到别人的分支。