# 项目配置

# 1.1导言

# 1.1.1目的

## 为“暨南大学毕业论文选题系统”项目的配置管理过程提供一个实施规范，作为项目配置管理实施的依据和指南。

# 1.1.2范围

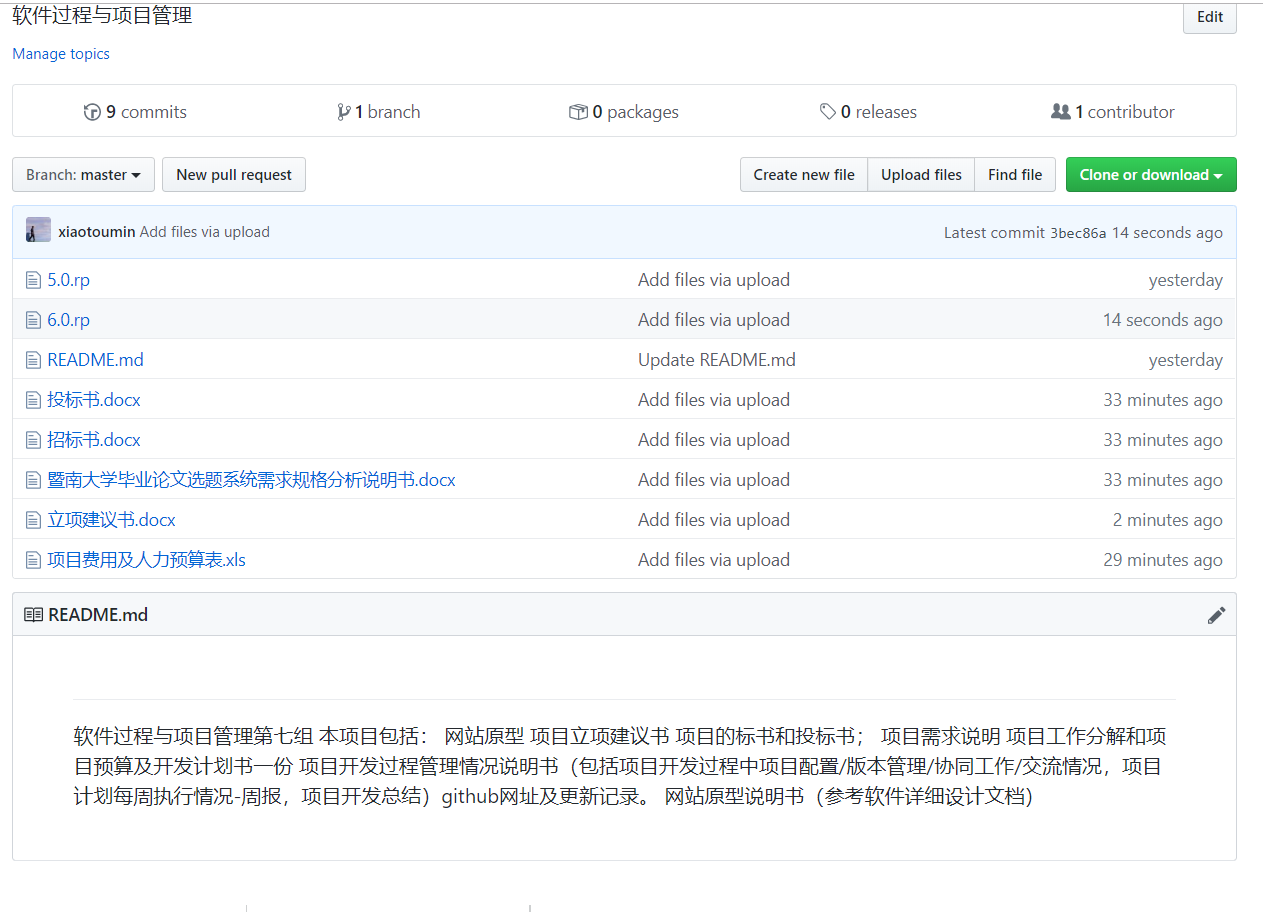
## 仅适用于“暨南大学毕业论文选题系统”项目的配置管理过程。

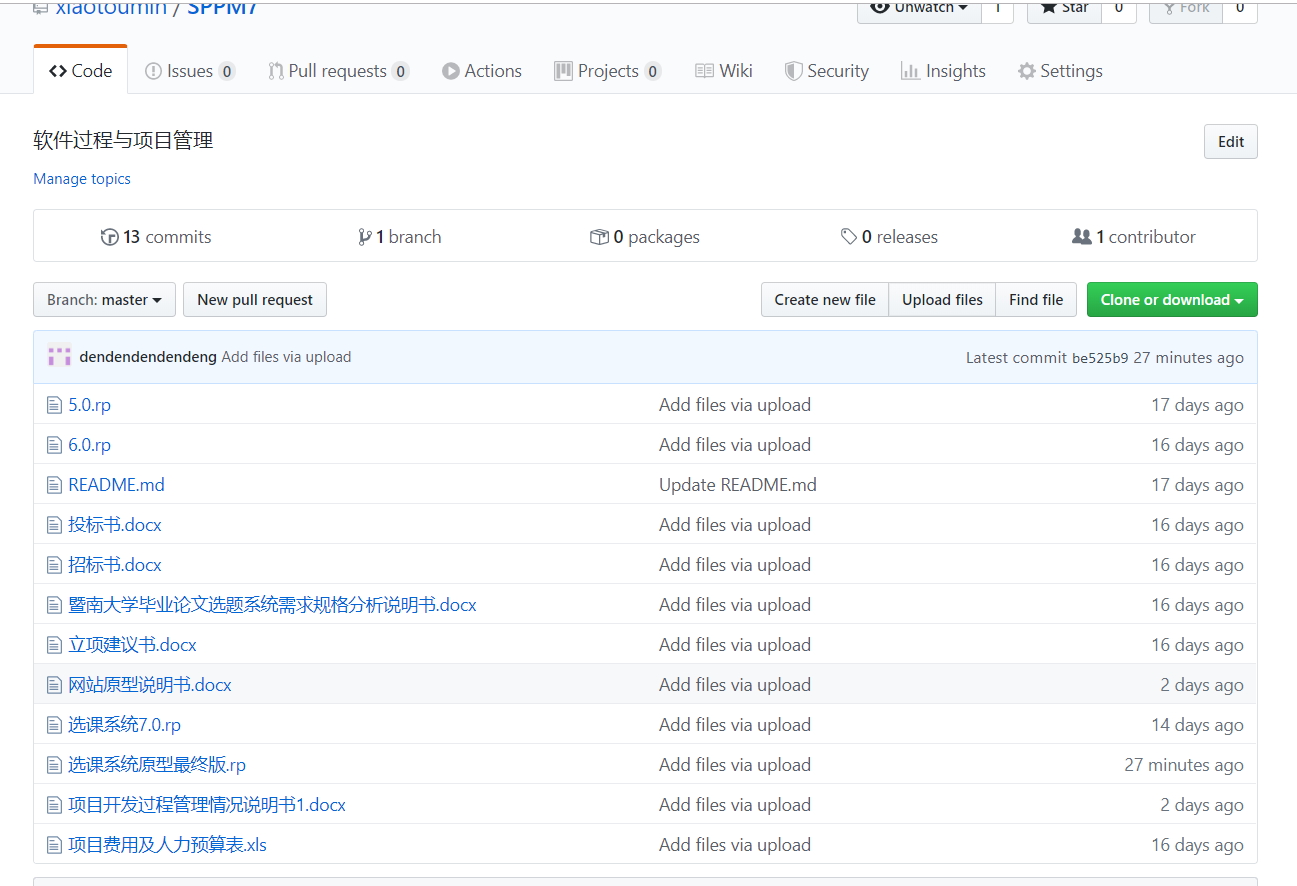
# 1.1.3版本更新记录

我们的Github网址：<https://github.com/xiaotoumin/SPPM7>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改内容 | 修改人 | 审核人 | 日期 |
| 1.0 | 初始版 | 吴英华、谢安、丁晓辉 | 李婧 | 2019.11.10 |
| 2.0 | 管理员界面 | 吴英华、丁晓辉 | 李怡菲 | 2019.11.27 |
| 3.0 | 风格统一 | 吴英华 | 谢安 | 2019.11.28 |

更新截图：





# 1.2配置管理流程

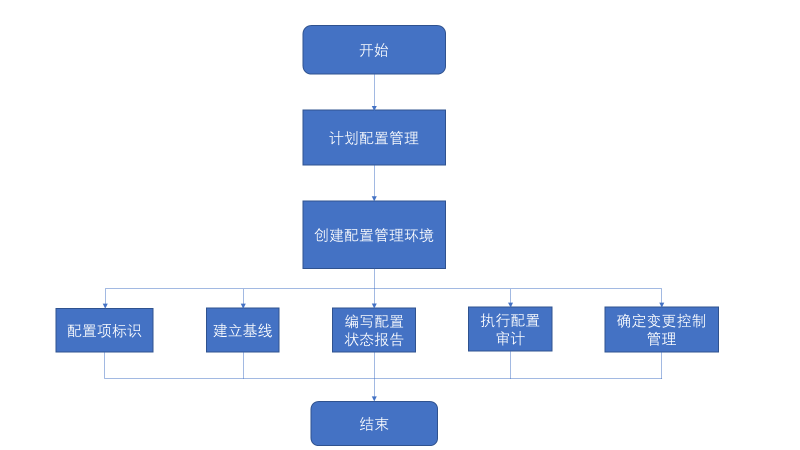
如图1所示，首先进行配置管理计划，并据此搭建配置计划中所需要的配置环境，同时确定配置管理活动，包括配置项标识，建立基线，编写配置状态报告，执行配置审计，确定并更控制管理。

图1 配置管理流程图

# 1.3配置项标识

项目文档命名规范：

<项目名>V<发布版本号><文件种类>\_<子系统名称>[<模块名称>]

# 1.4配置库建立

# C:\Users\李婧\AppData\Local\Temp\1575794367(1).png1.4.1 库结构

# 1.4.2基线建立及发布过程

# （1）基线所属的配置项，全部经过评审并解决了评审中提出的问题。

# （2）CCB对申请进行审批，审批通过后由配置管理员执行配置检查，然后可以建立并产品基线。 一般项目要建立的基线见下面的基线分类表。

# 表1 基线分类表

# 1.4.3用户权限

# 表2 用户权限表

# 1.5配置控制

# 1.5.1配置变更范围

（1）新增需求

（2）用户问题

（3）缺陷报告

（4）其他变更请求的处理

# 1.5.2配置变更流程

图2 配置变更流程图

# 1.6备份

为了保证文档的最大可恢复性，要随时及定期地进行备份工作。

（1） 随时备份：

1. 开发人员每天都要将自已当日修改的源文件在本地机器上进行备份。
2. 每天要将所有源文件在本地机备份。
3. 建议备份采用循环备份。

（2）定期备份

1. 备份形式为硬盘备份和光盘备份。硬盘备份时，要备份在独立的硬盘上；光盘备份时，要将光盘存放在可靠的地方。
2. 备份周期视具体情况而定。如果处于开发阶段，每周应对所有的源程序项进行备份，一般为每周周五；如果处于其它阶段，根据具体情况而定，但周期不能超过两周。
3. 备份原则应是保证文档的最大可恢复性。

# 版本管理

# 2.1正式版本

软件版本号由四部分组成：X.Y.Z.DATA\_希腊字母，

X：主版本号，用来表示提供给客户的产品功能的主要增强。在一个极端的例子中，主版本号的上升用来说明产品现在已经拥有了一个全新的功能类。从市场和许可权的角度来看，主版本号的升级相当于购买一个完全独立的产品。从开发者角度来看，一个主版本号的迭代差不多总是反映了一个新的独立分支或是其主干还可以延续主版本的生命期。

Y：特征版本号，用来表示产品新增了一些特征，或者是在原来文档中描述的特征上作了重要的修改。用来确定特征版本号什么时候需要修改的一个衡量标准就是产品功能说明书。产品的特征版本升级是在主版本之间保持产品竞争力的一种重要机制。

Z：缺陷修复版本号，用来表示在该版本上所做的缺陷维护行为的等级。版修复版本是稳定市场和最小化客户技术支持费用负担的一种重要机制。

Alpha版: 此版本表示该软件在此阶段主要是以实现软件功能为主，通常只在软件开发者内部交流，一般而言，该版本软件的Bug较多，需要继续修改。

Beta版: 该版本相对于α版已有了很大的改进，消除了严重的错误，但还是存在着一些缺陷，需要经过多次测试来进一步消除，此版本主要的修改对像是软件的UI。

RC版: 该版本已经相当成熟了，基本上不存在导致错误的BUG，与即将发行的正式版相差无几。

Release版: 该版本意味“最终版本”，在前面版本的一系列测试版之后，终归会有一个正式版本，是最终交付用户使用的一个版本。该版本有时也称为标准版。一般情况下，Release不会以单词形式出现在软件封面上，取而代之的是符号(R)。

例如：1.1.1.051021\_beta.第一个1为主版本号，第二个1为子版本号，第三个1为阶段版本号，第四部分为日期版本号加希腊字母版本号，希腊字母版本号共有5种，分别为：base、alpha、beta、RC、release。

# 2.2版本升级

版本升级应严格纳入版本管理的控制之下。应当谨慎地控制版本的升级，保障高版本的向下兼容性，或提供严格定义的升级方法。

主版本号(1)：当功能模块有较大的变动，比如增加多个模块或者整体架构发生变化。此版本号由项目决定是否修改。

子版本号(1)：当功能有一定的增加或变化，比如增加了对权限控制、增加自定义视图等功能。此版本号由项目决定是否修改。

阶段版本号(1)：一般是 Bug 修复或是一些小的变动，要经常发布修订版，时间间隔不限，修复一个严重的bug即可发布一个修订版。此版本号由项目经理决定是否修改。

日期版本号(191110)：用于记录修改项目的当前日期，每天对项目的修改都需要更改日期版本号。此版本号由开发人员决定是否修改。

希腊字母版本号(beta)：此版本号用于标注当前版本的软件处于哪个开发阶段，当软件进入到另一个阶段时需要修改此版本号。此版本号由项目决定是否修改。

每次版本升级，要填写版本升级记录表。

记录表样例如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 主版本号 | 子系统名称 | 子系统版本 | 发布日期 | 变更功能描述 | 发布人 | 批准人 | 备注 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

主版本号：记录当前发布的版本

发布日期：该版本批准发布的日期

修改文件：版本修改记录，版本修改日志

# 协同工作

1. 吴英华、丁晓辉：使用Axure进行毕业论文选题网站原型开发。
2. 谢安：参与毕业论文选题网站原型开发并将项目进展上传至GitHub。
3. 李怡菲：负责编写项目计划执行周报和项目开发总结。
4. 李婧：负责项目配置、版本管理文档以及更新项目进度。

# 交流情况

1. 首先进行分工，每个人认领自己的任务，并预设好任务DDL；
2. 在各自完成任务期间，如涉及到与其他成员任务有交叉的部分，共同协商完成；
3. 在DDL之前，每个成员将自己的任务完成并向其他成员展示，其他成员提出改进意见，该成员将对自己的工作成果进行修正；
4. 各个成员的最终工作成果汇总，得到完整项目及其相关说明文件。

# 项目周报

|  |  |  |
| --- | --- | --- |
| **周数**  **（以项目开始为准）** | **完成内容** | **总结** |
| 第一周 | 经过小组讨论，根据要求完成立项建议书一份以及招标书一份。 | 本课题立项建议书的重点应放在做这样一个项目能给学校带来的利益好处上，毕竟该建议书是由领导审核。 |
| 第二三周 | 小组讨论，深刻理解招标书，在此基础上完成初步功能需求分析。最后合作完成一份开发投标书。 | 前期写的时候基本是按照模板来的，注重逻辑上的分析。  老师点评认为有以下可以有所完善的地方：   1. 实施计划应更清晰地标明软件开发各个阶段的日期等。 2. 招标书的需求可以更丰富完整。 |
| 第三四周 | 小组讨论，上网找了一些模板，分工完成需求分析。 | 这次的作业做得比较赶，没有太细化。  主要需要完善的是用例，其功能需要具体化到每一个用例，比如学生注册需要填写哪一些功能，经过哪一些步骤等。 |
| 第五周 | 小组讨论，完成了软件项目WBS分解。选择使用ones project管理项目计划，统计规模、估算成本及时间，制作出甘特图。  本周任务是分工完成。 | 新的内容不多，重点还是分析细化。 |
| 第六七周 | 小组讨论，对之前的WBS分解进行进一步修改，并用ones project管理起来。  分配每个人网页制作的部分，分工制作最后整合在一起。  继续更新甘特图的每周进度。  在github上建立了项目。 | 本周任务量较大。讨论完WBS的进一步修改后，我们先是在b站上学习了Axure的基础入门用法，而后分工完成了网页制作。  也是因为网页制作过程中讨论不充分，导致了一些需要改善的问题：   1. 因为是分工完成的所以网页整体风格不够统一。 2. 学生界面查看不了自己的选择情况。 3. 管理员界面有些需求之外的不必要的功能。 |
| 第七八周 | 小组讨论，根据上次出现的问题进一步分工完善需求及网页。  继续在GitHub上更新项目进度。 | 老师反映思路上有一些小的问题：应该是管理员给老师分配论文数目，老师去完成。而我们理解的是，老师提交申请，管理员对内容进行审核。 |

# 项目开发总结

第一次学习项目软件过程管理，大家都不是很熟练。但是在一次次的小组讨论之后，我们一步步的完成了许多文档的书写以及最重要的，一个较为完整的网页的搭建，实现了最初需要的功能，这还是让我们挺有成就感的。中间确实会遇到各种各样的一些问题，但是更重要的是一次次发现问题之后去解决去完善的过程。我们都在这个课程里学到了很多。