**暨 南 大 学**

**项目开发情况说明书**

**项目名称： “吃货”交友中心网站开发**

**学 院：** **电气信息学院**

**专 业：**  **软件工程**

**课程名称： 软件项目管理**

**小组人员：** **陈仁湛 朱才华 荆俊彤**

**指导教师：** **陈双平**

**2014年 12 月 15 日**

|  |
| --- |
|  |
| 吃友垂直社交网 |
| 网站原型说明书 |
|  |
|  |

# 

目 录

**[1 引言 4](#_Toc408342889)**

**[2 开发过程论述 4](#_Toc408342890)**

[2.1项目简介 4](#_Toc408342891)

[2.2软件开发过程 4](#_Toc408342892)

[2.3分工模式 5](#_Toc408342893)

[2.4管理模式 5](#_Toc408342894)

[2.5交流模式 5](#_Toc408342895)

**[3 项目配置 5](#_Toc408342896)**

[3.1组织及职责 5](#_Toc408342897)

[3.2配置管理环境 6](#_Toc408342898)

[3.3配置管理活动 7](#_Toc408342899)

**[4.协同工作 9](#_Toc408342900)**

**[5.交流情况 10](#_Toc408342901)**

**[6.项目开发执行过程 10](#_Toc408342902)**

**[7.项目开发总结 10](#_Toc408342903)**

# 1 引言

为了概述项目开发过程的论述依据和开发过程中产生的配置事项、协同方式、项目计划执行情况及项目开发总结，引用该文档对上述情况进行记录。

# 2 开发过程论述

## 2.1项目简介

随着人民生活水平的改善，越来越多的人开始关注生活质量，“吃得好”变成了大多数人的追求，但是“吃什么好”和“什么好吃”是每个人会遇到的问题，本产品“吃友”垂直社交网站主要是提供给广大“吃货”一个吃货交友、美食讨论和美食组团的平台，同时还可以为美食商家提供广告位。产品的潜在客户数量巨大，技术要求简单发展前景广阔。而且产品所需开发时间短，维护简单，可以很好赶上并且满足社会及用户的需求。

## 2.2软件开发过程

**软件开发过程主要分为以下六个阶段：**

1. **软件规划**

包括：项目规划、计划评审；

1. **需求开发**

包括：用户界面设计、用户需求评审、修改需求和用户界面、编写需求规格说明书，需求验证；

1. **设计**

包括：概要设计、数据库ER图编制、建库等；

1. **实施**

包括：安全模块、用户管理模块、好友管理模块、前台展示模块、广告位模块、系统管理模块；

1. **系统集成**

包括：系统集成测试、环境测试；

1. **提交**

## 2.3分工模式

本次课程项目的开发过程全程由3位成员负责，属于小团队开发的形式，因此在项目开发过程中将存在工作界限模糊化的分工模式，譬如说某个子功能仅由一位成员全部负责，而不分设美工，网站逻辑等工作内容。

原型开发过程将是每个同学负责某些界面，最终的原型产品则是这些界面的组合，鉴于网站逻辑需要统一，整合工作将由一位成员完成，其余成员负责协助。

## 2.4管理模式

鉴于开发团队的规模较小，开发过程全程由3位成员负责，同时管理项目开发过程将受一定环境因素影响(校内)，管理权限上将有所约束，整个项目的开发过程将会是弱管理，即仅对定期团队交流、组内分工责任、交付成果进行管理，不对工期和细致的产品质量进行限制和管理。

## 2.5交流模式

组内成员的交流将主要依赖于任务驱动的小组会议进行交流，每次会议前每位成员将自备方案，会议结束后将对会议成果进行记录共享于小组交流群内。工作成果的交付和协同将依赖于GitHub和日常交流。

# 3 项目配置

## 3.1组织及职责

**表1：配置管理角色职责表**

|  |  |  |
| --- | --- | --- |
| **角色** | **人员** | **职责和工作范围** |
| 配置管理者 | 朱才华 | 1. 制定《配置管理计划》 2. 创建和维护配置库 |
| SCCB负责人 | 陈仁湛 | 1. 审批《配置管理计划》 2. 审批重大的变更 |
| SCCB成员 | 陈仁湛（项目经理）  荆俊彤（质量保证人员）朱才华（配置管理者） | 审批某些配置项或基线的变更 |

## 3.2配置管理环境

开发团队的配置管理主要通过GitHub及例会讨论进行确认，项目配置文档存放于各配置管理委员会成员的个人电脑上，同时GitHub存放有汇总配置项。

#### 3.2.1配置库目录结构

**表2：配置库的目录结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **内容** | **说明** | | **路径** |
| 1 | TCM | 技术合同管理  （即招标书和建议书） | | squirrel\docs\TCM |
| 2 | RM | 需求管理 | | squirrel\docs\RM |
| 3 | SPP | 软件项目规划 | | squirrel\docs\SPP |
| 4 | SCM | 软件项目跟踪与管理  （即项目开发过程） | | squirrel\docs\SPTO |
| 5 | SCM | 软件配置管理 | | squirrel\docs\SCM |
| 6 | SQA | 软件质量保证 | | squirrel\docs\SQA |
| 7 | SPE | 软件产品工程 | 设计 | squirrel\docs\SPE\DESIGN |
| 8 | 源代码 | squirrel\docs\SPE\SOURCE |
| 9 | 目标代码 | squirrel\docs\SPE\BUILD |
| 10 | 测试 | squirrel\docs\SPE\TEST |
| 11 | 发布 | squirrel\docs\SPE\RELEASE |

#### 3.2.2用户及权限

**表3：配置库的用户权限**

|  |  |  |
| --- | --- | --- |
| **类别** | **人员** | **权限说明** |
| 配置管理者 | 朱才华 | 负责项目配置管理，拥有所有权限 |
| 项目经理 | 陈仁湛 | 访问、读 |
| 质量保证人员 | 荆俊彤 | 访问、读 |
| 开发人员 | 陈仁湛  朱才华  荆俊彤 | 访问、读 |
| 高层管理 | 陈双平老师 | 访问、读 |

## 3.3配置管理活动

#### 3.3.1 配置项标志

3.3.1.1 命名规范

本项目配置项命名规范由5个字段组成，从左到右依次为：小组、项目、类型、编号和版本号。

**Group-School–RM–SRS-version 1.0.0**

小组：group5

项目：吃友

类型：最长5个字符

编号：最长8位数字/字符

版本号：version x.（每实现一个内容在原来的基础上+0.1）.（当前时间，如1312131239）

**图1：配置项命名规范**

3.3.1.2主要配置项

**表4：配置项列表**

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **主要配置项** | **标识符** | **预计正式发表时间** |
| 技术合同 | 项目立项建议书 | GR5-School-TCM-Contract-VERSION1.0.0 | 2014-10-25 |
| 项目招标书 | GR5-School-TCM-SOW-VERSION 1.0.0 | 2014-11-24 |
| 计划 | 《项目开发计划》 | GR5-School-SPP-PP-VERSION 1.0.0 | 2014-12-12 |
| 《质量保证计划》 | GR5-School-SPP-SQA-VERSION 1.0.0 | 无 |
| 《配置管理计划》 | GR5-School-SPP-SCM-VERSION 1.0.0 | 2014-12-12 |
| 需求 | 项目需求分析说明 | GR5-School-RM-SRS-VERSION 1.0.0 | 2014-12-12 |
| 设计 | 《总体设计说明书》 | GR5-School--Design-HL-VERSION 1.0.0 | 无 |
| 《数据库设计》 | GR5-School-Design-DB-VERSION 1.0.0 | 无 |
| 《详细设计说明书》 | GR5-School-Design-LL-VERSION 1.0.0 | 无 |
| 《设计术语及规范》 | GR5-School-Design-STD-VERSION 1.0.0 | 无 |
| 编程 | 网站原型 | GR5-School-Code-ModuleName-VERSIO | 无 |
| 编码规则 | GR5-School--Code-STD-VERSION 1.0.0 | 无 |
| 项目开发执行过程 | GR4-School-Code-SPTO-VERSION 1.0.0 | 2014-12-25 |
| 测试 | 《测试计划》 | GR4-School-Test-Plan-VERSION 1.0.0 | 无 |
| 《测试用例》 | GR4-School-Test-Case-VERSION 1.0.0 | 无 |
| 《测试报告》 | GR4-School-Test-Report-VERSION 1.0.0 | 无 |
| 提交 | 项目原型系统 | GR4-School-Product-Exe-VERSION 1.0.0 | 2014-12-25 |
| 项目使用说明书 | GR4-School-Product-Specification-VERSION 1.0.0 | 2014-12-25 |

3.3.1.3 项目基线

**表5 项目基线**

|  |  |  |
| --- | --- | --- |
| **基线名称/标识符** | **基线包含的主要配置项** | **预计建立时间** |
| 需求 | 需求分析说明 | 2014-12-12 |
| 总体设计 | 小组内部在工作分解过程中讨论确定 | 无 |
| 项目实现 | 网站原型设计、项目开发执行过程 | 2014-12-25 |
| 系统测试 | 《测试用例》、《测试报告》 | 无 |

3.3.1.4 配置项的版本管理

本项目采用分布式版本控制工具git进行版本管理。由于项目产生的文档、原型、项目管理文件都不是纯文本的，无法进行合并操作。因此，不使用分支工作流，仅保持默认的主分支(master).由项目负责人统一管理，根据各开发阶段的实际情况制定相应的版本选取规则，来保证开发活动的正常运作。

#### 3.3.2 变更管理

**变更管理的流程是：**

1. 由请求者提交变更请求，SCCB会召开复审会议对变更请求进行复审，以确定该请求是否为有效请求。典型的变更请求管理有需求变更管理、功能无法实现等。
2. 配置管理者收到变更请求时，进行配置文件的变更。
3. 配置管理者将此变更文件提交给SCCB，由SCCB确定是否需要修改，如果需要修改，SCCB应根据具体需求以及功能实现，确定需要修改的具体文件。
4. 配置管理者按照出库程序从配置库中取出需要修改的文件。
5. 项目人员将修改后的文件提交给配置管理者。
6. 配置管理者将修改后的配置项按入库程序放入配置库。
7. 配置管理者按SCCB标识出的修改文件，并按入库程序放入配置库。

#### 3.3.3 配置状态统计

利用配置状态统计，可以记录和跟踪配置项的改变。状态统计可用于评估项目风险，在开发过程中跟踪更改，并且提供统计数据以确保所有必需的更改已被执行。为跟踪工作产品基线，配置管理者需收集下列信息：

● 基线类型

● 工作产品名称

● 配置项名称

● 版本号

● 更改日期/时间

● 更改请求列表

● 需要更改的配置项

● 当前状态

● 当前状态发生日期

每当需求变更，管理者需要记录该变更并提交存档。

# 4.协同工作

项目开发过程中，所有小组成员在工作电脑上都配置了工具GitHub，任务分配主要根据甘特图WBS的分配进行。如果分配出现变动，小组成员主要通过主动线下交流来进行通知协商。每个单独页面作为一个可交付的工作包，但由于Git中无法对二进制文件进行协同工作，因此只能一个人写一个页面，交付成果的协同将依赖于组员交流和独立的判断是否有工作冲突。假如有工作冲突将交由项目管理者判断采用哪种更改抑或合并。整个项目的合并需要重复劳动，将所有页面文件拼为一个页面文件，因此在合并过程中将没办法协同工作。

# 5.交流情况

在整个课程项目过程中，小组成员的线下会议，基本上是一次任务一次会议，甚至一次任务多次会议。每次会议结束后，小组组长整理会议内容，然后在讨论群内共享会议内容，保证每个小组成员每次会议后都清晰明白所要做的任务。在项目需求获取阶段，通过多次会议，使项目需求基本明确清晰，然后对于每次的需求变更都开会讨论取舍和增添，而项目开发过程中，组内成员也开过多次会议进行讨论。对于网站原型的风格统一，由于每个人的观念看法不一样，在分工做出来之后，每一部分的风格差异较大，甚至逻辑冲突，最后也是通过小组会议，达成一致。

# 6.项目开发执行过程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **WBS** | **任务名称** | **工期** | **开始时间** | **完成时间** | **实际开始时间** | **实际完成时间** | **影响因素** |
| **1.1** | **软件规划** | **2d** | **2014-11-24** | **2014-11-25** | **2014-11-24** | **2014-11-25** | **无** |
| 1.1.1 | 项目规划 | 1d | 2014-11-24 | 2014-11-24 | 2014-11-24 | 2014-11-24 | 无 |
| 1.1.2 | 计划评审 | 1d | 2014-11-25 | 2014-11-25 | 2014-11-25 | 2014-11-25 | 无 |
| **1.2** | **需求开发** | **14d** | **2014-11-26** | **2014-12-15** | **2014-11-26** | **2014-12-20** |  |
| 1.2.1 | 用户界面设计 | 1d | 2014-11-26 | 2014-11-26 | 2014-11-26 | 2014-11-28 | 风格不统一 |
| 1.2.2 | 用户需求评审 | 10d | 2014-11-27 | 2014-12-10 | 2014-11-29 | 2014-12-12 | 需求变更 |
| 1.2.3 | 修改需求、修改用户界面 | 1d | 2014-12-11 | 2014-12-11 | 2014-12-13 | 2014-11-15 | 需求变更、原型逻辑冲突 |
| 1.2.4 | 编写需求规格说明书 | 1d | 2014-12-12 | 2014-12-12 | 2014-12-16 | 2014-12-16 |  |
| 1.2.5 | 需求验证 | 1d | 2014-12-15 | 2014-12-15 | 2014-12-17 | 2014-12-17 |  |

# 7.项目开发总结

总体而言，整个项目开发过程中，团队合作交流融洽，每个小组成员都能积极参与，整个过程主要按照经典的软件过程来做，但是由于经验不足，还是走了不少弯路。

团队的成员都没有这个经验，都对github工具不熟悉，而且项目产生的文档，项目计划，原型都是无法用github进行管理，故版本控制基本形同虚设。在原型设计的时候，由于每个人的观点看法不同，原型的风格有很大差异性，而且在逻辑上也有冲突不足的地方。最后通过多次会议才形成统一的观点。

还有就是软件过程的问题，系统设计和用例描述有很多细节没想到，导致后期实现原型的时候要重新设计一些细节，导致工作量增大；项目计划和甘特图是针对完整开发而设计，而且是虚构场景，故实现原型时无法按项目计划执行。

虽然有很多不足的地方，但是这也算是难得的宝贵经验，下次我们一定可以做得更好。