

软 件 工 程 实 验

**基于区块链EOS平台的美食点评系统**

**-----小众点评**

**配置过程管理分析报告**

**编写人：20TeamD**

**编制时间：2020年4月23日**

变更记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2020/04/23 | 叶明林 | 文档初稿 | 许栋亮、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |
| v1.1.0 | 2020/06/03 | 叶明林 | 更新统计数据 | 许栋亮、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |

# 1 引言

软件配置管理是一种标识、组织和控制修改的技术。软件配置管理应用于整个软件工程过程。在软件建立时变更是不可避免的，而变更加剧了项目中软件开发者之间的混乱。配置管理活动的目标就是为了标识变更、控制变更、确保变更正确实现并向其他有关人员报告变更。从某种角度讲，配置管理是一种标识、组织和控制修改的技术，目的是使错误降为最小并最有效地提高生产效率。

撰写本文档是为了对各配置文档进行迭代变更的统计与分析，进一步追踪在进行项目开发过程中进行配置管理活动中发现的问题并进行改善。

# 2 实验过程分析

在对配置管理过程进行记录和分析时，主要记录和分析如下信息：

配置文档总表：记录项目中所有被管理的文档列表以及各个文档总变更次数。

配置文档变更列表：记录各个实验的产出物——各个配置文档的变更统计，各迭代中变更次数；分析各配置文档的变更情况，包括可量化的字数、图表数、引用数等信息以及不可量化的内容信息等。

各迭代的变更分析：针对上述记录信息对各个文档进行变更原因分析。

## 2.1 配置文档总表

本项目所有被管理的文档列表，以及各个文档在Github上的总变更次数如下：

|  |  |  |
| --- | --- | --- |
| 文档名称 | 版本变更次数 | 统计日期 |
| 项目开发计划书 | 5 | 2020/06/03 |
| 需求规格说明书 | 10 | 2020/06/03 |
| 需求评审表设计 | 1 | 2020/06/03 |
| 需求规格说明书检查单 | 1 | 2020/06/03 |
| 测试需求规格说明书 | 5 | 2020/06/03 |
| 测试报告 | 5 | 2020/06/03 |
| 测试需求评审检查单 | 1 | 2020/06/03 |
| 进度计划与监控报告 | 4 | 2020/06/03 |
| 配置管理计划 | 2 | 2020/06/03 |
| 配置管理过程分析报告 | 2 | 2020/06/03 |
| 工作量分析报告 | 8 | 2020/06/03 |

## 2.2 实验一：软件需求分析

实验一的需求分析阶段产出《需求规格说明书》，截止至统计时间，在实验一中，我们共产出了十个版本。《需求规格说明书》主要内容包含软件的各项需求，其变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| 1.0.0 | 2020.3.26 | 许栋亮  陈瀚清 | 文档初稿 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯 |
| 1.0.1 | 2020.3.27 | 陈瀚清 | 修改项目名称、将用例图与用例说明放在一起 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯、许栋亮 |
| 2.0.0 | 2020.4.1 | 许栋亮 | 添加参考文献、修改目录结构 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |
| 2.0.1 | 2020.4.5 | 许栋亮 | 修改项目名称、需求规格说明书检查 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |
| 2.0.2 | 2020.4.6 | 叶明林 | 修改了需求规格说明书段落的行间距、统一了部分专业术语的大小写 | 许栋亮、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |
| 2.1.0 | 2020.4.9 | 许栋亮  陈瀚清  黄雨鹤 | 根据需求评审，对需求规格说明书进行了修改 | 刘思远、刘新宇、  林讯、叶明林 |

## 2.3 实验二：软件需求评审

实验二需求评审主要针对其他四个小组进行评审，并根据其他四个组的评审修改需求文档，需求文档部分的变更已经在上一节给出，在这里不加以赘述。评审报告产出对B、C、E、F组的需求评审，共计四份。由于对其他小组的评审文档都是经过会议讨论后再进行撰写的，所以都只有一个版本，即初稿。

需求评审表设计变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2020/04/08 | 许栋亮 | 初稿 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |

## 2.4实验三：软件设计与实现

实验三的软件设计与实现阶段产出《软件设计说明书》，截止至统计时间，在实验三中，我们共产出了一个版本。《软件设计说明书》变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2019/05/06 | 叶明林、刘新宇、林讯 | 初稿 | 许栋亮、刘思远、黄雨鹤、  陈瀚清 |

## 2.5 实验四：软件测试

实验四的软件测试阶段产出《测试需求规格说明书》和《测试报告》。《测试需求规格说明书》主要内容包含软件的各项测试需求，截止至统计时间，共产出了五个版本，其变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2020/05/15 | 黄雨鹤  刘思远 | 文档初稿 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.0.1 | 2020/05/18 | 黄雨鹤  刘思远 | 完善测试用例 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.1.0 | 2020/05/19 | 黄雨鹤 | 增加测试方法部分 | 刘思远  许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.2.0 | 2020/05/24 | 黄雨鹤 | 根据评审意见修改本文档 | 刘思远  许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.3.0 | 2020/05/30 | 刘思远  黄雨鹤 | 根据评审意见修改本文档 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |

《测试报告》主要内容包含对软件测试的结果，截止至统计时间，也共产出了五个版本，其变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2020/05/15 | 黄雨鹤、刘思远 | 文档初稿 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.1.0 | 2020/05/18 | 黄雨鹤、刘思远 | 第二次测试结果 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.1.1 | 2020/05/19 | 黄雨鹤、刘思远 | 更新回归测试结果 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.2.0 | 2020/05/24 | 黄雨鹤 | 根据评审意见修改本文档，根据修改后的测试用例得到测试结果 | 刘思远  许栋亮  陈瀚清  叶明林  林讯  刘新宇 |
| v1.3.0 | 2020/05/30 | 刘思远、黄雨鹤 | 根据评审意见修改本文档，根据修改后的测试用例得到测试结果 | 许栋亮  陈瀚清  叶明林  林讯  刘新宇 |

## 2.6 实验五：软件测试评审

实验五测试评审主要针对其他四个小组的测试过程及文档进行评审，并根据其他四个组的评审修改我们组的测试需求文档和软件，测试需求文档部分的变更已经在上一节给出，在这里不加以赘述。评审报告产出对A、H、I、G组的需求评审，共计四份。由于对其他小组的评审文档都是经过会议讨论后再进行撰写的，所以都只有一个版本，即初稿。

测试需求评审表设计变更细节如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v1.0.0 | 2020/05/19 | 陈瀚清 | 初稿 | 叶明林、刘思远、刘新宇、黄雨鹤、  林讯、许栋亮 |

## 2.7 实验六：软件项目计划与监控

实验六产出《项目计划说明书》和《进度计划与监控报告》，《项目计划说明书》变更细节如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 |
| v0.1 | 2020/03/016 | 刘新宇 | 添加项目概述 | 刘思远、许栋亮、黄雨鹤、林讯、叶明林、陈瀚清 |
| v0.2 | 2020/03/016 | 叶明林 | 添加项目团队组织 | 刘思远、许栋亮、黄雨鹤、林讯、刘新宇、陈瀚清 |
| v0.3 | 2020/03/016 | 刘思远 | 添加项目简介 | 叶明林、许栋亮、黄雨鹤、林讯、刘新宇、陈瀚清 |
| v0.4 | 2020/03/017 | 许栋亮、黄雨鹤 | 添加项目进度计划 | 刘思远、刘新宇、林讯、叶明林、陈瀚清 |
| v0.5 | 2020/03/017 | 陈瀚清 | 添加实施计划 | 叶明林、许栋亮、黄雨鹤、林讯、刘新宇、刘思远 |

《进度计划与监控报告》变更细节如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 |
| v1.0.0 | 2020/03/26 | 刘思远 | 撰写文档总体框架 |
| v1.0.0 | 2020/03/26 | 许栋亮、黄雨鹤 | 撰写项目总体计划 |
| v1.1.0 | 2020/04/2 | 刘思远 | 修改项目准备阶段与撰写需求分析阶段的执行情况。 |
| v1.2.0 | 2020/04/23 | 刘思远 | 撰写软件需求评审与软件需求复评审阶段的执行情况 |

## 2.8 实验七：软件配置管理

实验七产出《配置管理计划》和《配置管理过程分析报告》。

《配置管理计划》的变更细节如下所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 主要编制人 | 版本说明 | 审核人 |
| V1.0.0 | 2020/3/23 | 叶明林 | 初稿 |  |
| v1.0.1 | 2020/4/1 | 叶明林 | 添加了文档命名规范 | 刘新宇、黄雨鹤、刘思远 |
|  |  |  |  |  |

《配置管理过程分析报告》的变更细节如下所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 主要编制人 | 版本说明 | 审核人 |
| v1.0.0 | 2020/04/23 | 叶明林 | 初稿 | 刘思远、许栋亮、黄雨鹤、林讯、刘新宇、陈瀚清 |
| v1.1.0 | 2020/06/03 | 叶明林 | 更新统计数据 | 许栋亮、刘思远、刘新宇、黄雨鹤、  林讯、陈瀚清 |

## 2.9 实验八：软件工程实验追踪分析

实验八产出《工作量分析报告》，变更细节如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 备注 |
| v1.0.0 | 2020/3/17 | 许栋亮 | 目录结构 |
| v1.1.0 | 2020/3/23 | 刘新宇 | 增加评分指标工时记录，修改文档整体结构 |
| v1.2.0 | 2020/4/1 | 许栋亮 | 增加评分标准 |
| v1.2.1 | 2020/4/2 | 刘新宇 | 汇总需求分析第一阶段及以前工时记录 |
| v1.2.2 | 2020/4/9 | 刘新宇 | 汇总需求分析第二阶段工时记录 |
| v1.2.3 | 2020/4/16 | 刘新宇 | 汇总需求初评审阶段工时记录 |
| v1.2.4 | 2020/4/23 | 刘新宇 | 汇总需求复评审阶段工时记录 |
| v1.2.5 | 2020/5/7 | 陈瀚清 | 汇总设计阶段工时记录 |

# 3 配置管理过程问题与改进

## 3.1 远程协作导致修改冲突

问题描述：由于客观原因，导致组员间不能直接进行面对面的交流，这直接导致了交流效率和分工效率的下降。在交流效率低的情况下，需要有一个明确的文档变更流程来较少不必要的版本冲突。

解决方法：在《配置管理计划》的初稿（即V1.0.0版本）中，对文档的变更及向Github提交的流程做出了明确的规定。

## 3.2 文档提交不及时

问题描述：同时采用金山在线文档和Github对文档进行管理，很容易导致组员们在在线文档中修改后忘记将其同步至Github上，这导致了有文档没有按时提交的问题。

解决方法：在《配置管理计划》的初稿（即V1.0.0版本）中，明确了每个文档的负责人，由每个文档的负责人负责在每个文档的截至提交日期前将文档提交到Github。

## 3.3 文档命名不规范

问题描述：由于每个实验的负责人不一样，相应文档初稿的撰写者也不一样，导致提交时文档的命名风格不一致。

解决方法：在《配置管理计划》V1.0.1版本中添加了文档命名规范相关内容。

# 4 配置管理过程总结

1. 使用git等版本控制软件能很方便、有效地对关键文档进行版本控制及变更追踪，但这需要软件项目小组事前制定好版本控制流程，否则版本冲突、负责人员和审核人员不明确等问题会大大降低效率。
2. 命名规范、变更记录规范等规范有利于清楚地、唯一地标识软件的各个版本，提高审查校验的效率。
3. 实施基线管理很重要，能明确各个开发阶段的职责，使上游阶段为下游阶段提供稳定的工作基础，也为软件产品的维护和新版本开发提供了良好的基础。