

质量保证计划

**【V0.1】**

**项目组长：张伟鹏**

**组员：丁磊，余敬，唐子煜**

**日期：2017年3月26日**

**跟踪记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **跟踪记录表** | | | | | |
| 版本 | 修改日期 | 修改问题 | 跟踪情况 | 修改人 | 审核人 |
| V0.1 | 2017.3.26 | 创建 |  | 张伟鹏 |  |
| V0.11 | 2017.3.30 | 增加了两个风险和应对 |  | 张伟鹏 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 引言 1](#_Toc478308233)

[1.1 目的 1](#_Toc478308234)

[1.2 参考资料 1](#_Toc478308235)

[2. 管理 1](#_Toc478308236)

[2.1 机构 1](#_Toc478308239)

[2.2 任务 1](#_Toc478308240)

[2.3 职责 2](#_Toc478308241)

[2.4 人员分工 2](#_Toc478308242)

[3. 基本文档的评审和检查 2](#_Toc478308243)

[3.1 软件需求规格说明书评审 2](#_Toc478308245)

[3.2 可行性分析报告评审 2](#_Toc478308246)

[3.3 软件总体设计评审 2](#_Toc478308247)

[3.4 软件详细设计 2](#_Toc478308248)

[3.5 软件数据库设计 3](#_Toc478308249)

[3.6 代码清单 3](#_Toc478308250)

[4. 软件配置管理 3](#_Toc478308251)

[4.1 版本管理 3](#_Toc478308253)

[5. 记录收集、维护和保存 3](#_Toc478308254)

[6. 风险管理计划 4](#_Toc478308255)

[7.1 风险评估 4](#_Toc478308259)

[7.2 风险控制方案 5](#_Toc478308260)

# 引言

## 目的

本计划的目的在于对所开发系统软件规定各种必要的质量保证措施，最终能成功完成产品的开发。

## 参考资料

《软件质量保证》

《软件工程导论》

《软件需求》

# 管理



## 机构

在项目整个开发期间，必须成立软件质量保证小组负责质量保证工作。由项目的质量保证组代表任组长。

软件质量保证组和软件质量保证人员必须检查和督促本计划的实施，软件质量保证人员有权直接向软件质量保证组报告软件质量状况。

## 任务

软件质量保证人员的主要任务包括：

* + 1. 制定《质量保证计划》
    2. 产品审计
    3. 过程审计
    4. 跟踪问题管理
    5. 度量和报告

## 职责

* + - 1. 组长全面负责有关软件质量保证的各项工作；
      2. 配置管理人员负责有关软件配置变动、数据文档的备份保存
      3. 任务负责人之间互相监督进度

## 人员分工

组长：张伟鹏

配置管理人员：张伟鹏

各任务负责人：余敬，唐子煜，丁磊

# 基本文档的评审和检查



## 软件需求规格说明书评审

需求规格说明书是否能满足系统功能的定义和确定。

## 可行性分析报告评审

可行性研究报告应做到内容齐全，结论明确，数据准确，论据充分，以满足决策者定方案定项目的需要。

## 软件总体设计评审

软件是否按照用户需求进行设计，软件设计是否合理，经济性是否合理，界面原型是否完善，是否有分成各个子模块，内容是否完整。

## 软件详细设计

软件是否总体进行设计，有没有描写出相关算法，对于各个子模块系统是否有进行详细设计，内容是否完整。

## 软件数据库设计

是否有详细的数据字典，是否有ER图，内容是否完整，是否可以追溯。

## 代码清单

是否有按照代码规范实现，内容是否完整，是否可以追溯。

# 软件配置管理



## 版本管理

1.首先在服务器(GITHUB)上建立一个新的仓库 ，作为项目配置数据库。在此目录下按照每个项目组建一个分目录，分目录用于区分不同的文档和代码，便于管理。

2.项目文档master分支一般只有项目经理和属于该项目的开发人员和配置管理员能够访问。配置管理员负责分配访问权限，项目组长则具有较大的权限——读取、添加和更改；一般开发人员只有关于brench的读取、添加和更改权限，对于master分支只有读取权限。

3.在项目开发的某一阶段结束时，也就是里程碑时，并且通过了内部的审查，做为正式版本的第一版——1.0版本。

4.在以后的开发中，如果软件需要修改，可以保存到各个开发人员的brench分支，经过组长同意后通过配置管理员更新到master分支

5.每次进行master变更需要在所有人的确定下进行。

# 记录收集、维护和保存

小组会议、对成果的审核记录也要放在GIT里保存。

# 风险管理计划



## 风险评估

#### 过程方面的问题

1. 任务可能发生延期的情况
2. 承担分析任务角色的人并不清楚如何分析任务
3. 可能出现部分小组成员消极处理的情况

#### 规划方面问题

1. 对工作量估计错误导致进度严重落后

#### 交流方面问题

1. 组内遇到问题未能及时沟通，导致问题花过多的时间解决
2. 小组成员不及时反映进度**，**导致任务延期

#### 需求获取方面的风险

1. 客户参与程度不高，以及不适当的用户介入
2. 客户对产品需求意见不一致
3. 用户不能明确定义他们的需求
4. 遗漏了必要需求
5. 指定需求不正确或不适当，强制接受上级管理层或外部权威指定的需求
6. 未能正确划分用户群，需求获取不全面

#### 需求分析方面的风险

1. 指定了没必要的需求
2. 指定并构建了功能，但却没使用这一功能
3. 需求不够清晰，无法编写测试用例
4. 没有设定需求优先级，所有的需求似乎都是一样重要
5. 当添加新需求时，分析人员不能做出知情的折衷取舍决策
6. 涉众之间的优先级发生冲突

#### 编写需求规格说明方面的风险

1. 需求没有编写成文档，客户向开发人员以口头方式或其他非正式渠道提供需求信息
2. 客户或开发人员认为要将现有系统中已有的功能复制到新系统中
3. 需求文档没有精确描述系统
4. 存在不同的需求版本或需求版本有冲突

#### 人员方面的风险

1. 项目经理变更
2. 团队成员退出
3. 团队人员变更
4. 团队成员临时有事或其他方面的原因请假

## 风险控制方案

#### 过程方面的控制

1. 编写《小组规章》，明确赏罚措施，避免人员消极应对造成的延后
2. 后继任务负责人监督前驱任务，实现成员之间监督
3. 进行Teambuilding提高组员的积极性

#### 规划方面的控制

1. 在充分地理解需求之前不要承诺产品的交付时间表，增强团队的需求分析能力
2. 为项目的需求开发和管理定义角色并分配其职责，指定专人负责管理需求

#### 交流方面的控制

1. 每周进行多次线上会议，确保组内沟通
2. 没周定时汇报进度，确保进度不因此延后

#### 需求获取方面的控制

1. 让技术水平高的分析人员去获取用户需求
2. 明确定义用户类别，若管理人员没派出真正用户参与分析，则应从直接用户之外的其他涉众那里获得信息
3. 构建原型，让用户来评估这些原型
4. 使用原型让用户参考，与用户进行充分的沟通，尽量能够让知识丰富的用户参与获取需求，可以适当增加分析人员的人数对用户获取需求
5. 确定有缺陷的需求会带来哪些问题，就不精确的需求可能带来的风险与高级权威人士进行交流

#### 需求分析方面的控制

1. 记录下每个需求的来源和理由
2. 组内审查需求的可测试性
3. 开发一个协作的过程，以便设定需求优先级
4. 将重点放在兼顾业务的利益上，不要维护感情和政治上的地位
5. 跟踪每一个待确定的问题，直到问题得到解决

#### 编写需求规格说明方面的控制

1. 定义并遵循一个需求开发过程，明确各个角色的职责并严格遵循
2. 对现有系统进行全面分析，在编写需求规格说明时要包括新系统的所有预期功能
3. 遵循一个变更控制流程，当接受变更时相应地更新需求，汇集换件涉众来评审修改过的需求规格说明
4. 定义并遵循需求文档良好的版本控制，将每次更新的文档都存入版本控制器中

#### 人员方面的控制

1. 尽快响应人员变更机制，新的项目经理应尽快熟悉整个管理过程，并明确每个人的职责
2. 重新安排项目进度与任务分配
3. 让新成员快速明确该项目，分配好任务使其尽快加入到该项目的开发中
4. 通过变更机制让其他人员顶替或将根据当时的情况对任务进行适当的分配