

质量保证计划

**【V0.1】**

****

**项目经理：余敬**

**组员：丁磊，张伟鹏，陈建伟，唐子煜，**

**日期：2016年10月30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 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[质量保证计划 1](#_Toc465590628)

[1. 引言 3](#_Toc465590629)

[1.1 目的 3](#_Toc465590630)

[1.2 参考资料 3](#_Toc465590631)

[2. 管理 3](#_Toc465590632)

[2.1 机构 3](#_Toc465590635)

[2.2 任务 4](#_Toc465590636)

[2.3 职责 4](#_Toc465590637)

[2.4 人员分工 4](#_Toc465590638)

[3. 文档 4](#_Toc465590639)

[3.1 基本文档 4](#_Toc465590641)

[3.2 其他文档 5](#_Toc465590642)

[3.3 文档质量的度量准则 5](#_Toc465590643)

[4. 评审和检查 6](#_Toc465590644)

[4.1 软件需求规格说明书评审 6](#_Toc465590646)

[4.2 可行性分析报告评审 6](#_Toc465590647)

[4.3 软件总体设计评审 6](#_Toc465590648)

[4.4 软件详细设计 6](#_Toc465590649)

[4.5 软件数据库设计 6](#_Toc465590650)

[4.6 代码清单 6](#_Toc465590651)

[5. 软件配置管理 7](#_Toc465590652)

[5.1 版本管理 7](#_Toc465590654)

[5.2 微小改正时的变更控制 7](#_Toc465590655)

[5.3 较大变动时的变更控制 7](#_Toc465590656)

[5.4 配置状态报告 8](#_Toc465590657)

[5.5 配置审核 8](#_Toc465590658)

[6. 媒体控制 8](#_Toc465590659)

[7. 记录收集、维护和保存 9](#_Toc465590660)

[8. 风险管理计划 9](#_Toc465590661)

[8.1 风险评估 9](#_Toc465590665)

[8.2 风险控制 11](#_Toc465590666)

# 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，以保证交付文档能够满足项目委托书或合同中规定的各项需求。

## 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. 子系统的软件质量保证人员负责测试复查和文档的规范化检查工作

## 人员分工

组长：唐子煜

配置管理人员：唐子煜

子系统的软件质量保证人员：余敬，张伟鹏，丁磊

# 文档

本章给出了在项目需要编制的文档名称及其要求，并且规定了评审文档质量的通用的度量准则。



## 基本文档

《软件需求规格说明书》

《可行性分析报告》

《软件总体设计》

《软件详细设计》

《软件数据库设计》

《项目章程》

《项目计划》

《代码清单》

## 其他文档

《会议记录》

《软件质量保证》

《工作分解结构》

《甘特图》

《代码规范》

## 文档质量的度量准则

文档是软件的重要组成部分，是软项目各个不同阶段的产品描述。评审文档质量的度量准则是有以下几条：

* + 1. 完备性：应按照以保证在开发阶段结束时其文档是齐全的。
    2. 正确性：在软件开发各个阶段所编写的文档的内容，必须真实的反映阶段的工作且与该阶段的需求相一致。
    3. 简明性：在项目所编写的各种文档的语言表达应该清晰、准确简炼，适合各种文档的特定读者。
    4. 可追踪性：在软件开发各个阶段所编写的各种文档应该具有良好的可追踪性。 可追踪性包括：1）文档变更可追踪。2）文档内容可追踪
    5. 自说明性：在软件开发各个阶段所编写的各种文档应该具有较好的自说明性。

文档的自说明性是指在软件开发各个阶段中的不同文档能独立表达该软件其相应阶段的阶段产品的能力。

* + 1. 规范性：在软件开发各个阶段所编写的各种文档应该具有良好的规范性。

文档的规范性包括文档的封面、大纲、术语的含义以及图示符号等符合有关规范的规定。

# 评审和检查



## 软件需求规格说明书评审

可行性研究报告应做到内容齐全、是否符合用户需求，有没有多次找用户询问需求，如果需求有修改文档是否修改，是否可以追溯。

## 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。三个可行性论证是否论据充分，是否合适，是否可以追溯。

## 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，是否可以追溯。

## 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，是否可以追溯。

## 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，目录名由“小组代号”+“项目名”构成，然后在此项目组目录下按照所属每个项目建一个子目录，同一项目的开发文档存放在一个目录下，项目编号紧跟项目名就是目录名，使得所有开发文档能分门别类的组织存放，便于查询。

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，通过了该阶段评审的这些开发文档申请添加到仓库master分支，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.在各个评审阶段产生的所有评审报告和修改报告都要进行保存，并且可以进行追述。

## 微小改正时的变更控制

1.在评审或测试后发现的问题由评审组组长或项目经理形成《软件问题报告单》或《源代码修改记录单》，并通知配置管理员。

2.由配置管理员将需要修改的软件的备份从项目配置数据库（master）中检出，开发人员执行修改后根据9.2再次上传。

3.修改完毕后进行修改测试，编程错误累计到了一定的量或者测试时间已满一个月（从上一次入配置库后算起），凭《源代码修改记录单》及修改后的源代码，通知配置管理员，配置管理员确定测试报告的完备性，并在核对软件修改内容和修改人员填写的《软件修改报告单》或《源代码修改记录单》中的修改描述一致后，将文件登入项目配置数据库中，生成新版本。

4.配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况。

## 较大变动时的变更控制

1.开发人员或用户提出影响较大的修改要求（这是指要增加或删除某些功能或者是发现错误的阶段在造成错误的阶段的后面等）。

2.配置管理员在收到这类修改要求时，必须组织有项目经理以及开发人员参加的修改评审会，讨论修改的影响范围，修改的必要性、可行性以及修改方法、步骤和实施计划。

3.在修改方案通过并经项目经理审核后，要由产品开发部经理签字批准。涉及重大技术方案的修改时，修改方案必须由总工程师或技术总监签字批准。以决断修改工作中各项活动的先后顺序及各自的完成日期，以保证整个开发工作按原定计划日期完成。

4.配置管理员在接到修改批准——由项目经理或产品开发部经理或总工程师或技术总监签字同意的《软件问题报告单》后才可将需修改的软件的备份从项目数据库中检出，开发人员执行修改。

5.修改完毕后，交客户服务部进行测试和评审，测试和评审都通过后，交配置管理员处理。

6.配置管理员检查测试报告和评审报告是否完备，核对《软件修改报告单》中的修改描述和修改后的软件是否相符。核查结果符合要求，配置管理员将修改后的软件登入项目数据库中，生成新版本。

7.配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况对受影响的软件做出相应的修改。

## 配置状态报告

1.两份配置状态报告——《软件配置状态表》和《软件变更记录表》分别以电子表格的形式存放在项目分目录下，以便项目开发人员随时查询，了解软件的修改变化情况。

2.《软件配置状态表》由配置管理员负责填写，主要反映项目中各软件项的配置情况。开发人员通过查阅该表可及时全面的了解项目中软件项的配置使用情况。

3.《软件变更记录表》由配置管理员负责填写，主要记录软件开发过程中所有的修改情况，该表以修改时间排序，以便开发人员及时了解软件项最新的变化。

## 配置审核

为保证各项产品在技术上和管理上的完整性，总经理室在软件开发过程中的详细设计阶段和测试阶段完成时，对配置情况进行抽查。总经理室先提出要审核的内容和各项指标，逐项审核完成后要作好记录，形成《配置审核报告》。

# 媒体控制

为了保护计算机程序的物理媒体，以免非法存取，意外损坏或自然老化，相关软件数据以及文档都应设计软件配置管理人员妥善管理和存放各个阶段的文档及其专用支持软件的数据

# 记录收集、维护和保存

在研制与开发期间，要进行各种软件质量保证活动，准确记录、及时分析并妥善保存有关这些活动的记录，是确保软件质量的重要条件。在软件质量保证小组中，应有专人负责收集、汇总与保存有关软件质量保证活动的记录。

# 风险管理计划



## 风险评估

#### 过程方面的问题

1. 各项目之间的需求过程和文档模板不一致，所采用的需求过程无效
2. 承担分析任务角色的人并不清楚如何分析任务
3. 需求管理工具使用不充分

#### 规划方面问题

1. 需求不完整，需求详细程度不够
2. 需求工作的完成存在差距，多个人完成相同的需求活动
3. 在可以用的时间和资源约束下，所规划的需求超出了所能实现的需求
4. 没有编写项目范围文档或编写了范围文档但很糟糕

#### 交流方面问题

1. 访谈之前制定决策
2. 项目参与者没有公用相同的词汇

#### 需求获取方面的风险

1. 客户参与程度不高，开发人员对要实现的东西做了许多猜测
2. 不适当的用户介入
3. 客户对产品需求意见不一致
4. 用户不能明确定义他们的需求
5. 参与需求获取的人太多
6. 需求约束太多
7. 遗漏了必要需求
8. 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

#### 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
7. 开发人员发现需求含糊不清和不明确
8. 出处不同的需求来自不同用户类的需求有冲突，涉众之间难以对需求达成共识
9. 需求中包含有“待确定”之类的标记和需求中还有信息空缺

#### 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

#### 需求确认方面的风险

1. 产品没有达到业务目标或不满足用户期望，存在未陈述的，假定的或隐含的客户需求没有得到满足
2. 没有指定的质量属性和性能目标产品没有达到性能目标，或不满足用户对质量的其他期望

#### 变更管理方面的风险

1. 频繁变更需求，在开发过程后期发生了许多需求变更
2. 频繁添加新需求
3. 需求属于项目范围之内还是项目范围之外，反反复复来回变
4. 开发工作已经开始之后，项目范围又发生了变更
5. 需求变更没有传达给收影响的所有涉众
6. 涉众没有遵循变更控制过程，客户直接向开发人员提出需求变更
7. 与预期相比，变更影响到了更多的系统组件
8. 变更危害到其他需求

#### 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

#### 过程方面的控制

1. 对当前需求过程编写文档，对期望的过程创建一个提议的描述，对所有文档的编写统一模板与规范，收集并共享优秀的文档范例
2. 为需求分析编写工作描述和技能列表，为新的分析人员建立指导计划
3. 安排一名人员来管理工具并指导其他用户

#### 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
3. 在做出承诺之前，编写产品前景和项目范围文档，使其与业务目标一致，根据需求衍生开发时间表，在进度上要考虑培训时间和学习曲线时间，根据实际要求适当调整项目范围
4. 编写前景和范围文档必须获得关键涉众的认同，如果范围定义不良，不要开始着手项目

#### 交流方面的控制

1. 确定由哪些人对项目需求做出决策，并定义他们的决策制定过程确定产品代言人，对需求被拒绝，推迟或取消的历史原因编写文档
2. 定义专用术语，定义数据字典中的数据项，对开发团队提供业务领域的培训，对用户代表提供需求工程方面的培训

#### 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. 确定那些主要的客户，并采用产品代言人的方法，保证有足够的客户代表的积极参与
4. 构建原型，让用户来评估这些原型
5. 明确定义用户类别，将政治优先级与业务优先级和技术优先级区分开，将项目重点放在需要优先考虑的用户上
6. 用户可能并不了解需求由哪些信息构成，因此应该让发展技术水平高的分析人员进行需求获取或对用户进行有关需求工程的教育
7. 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
8. 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. 通过需求优先级明确价值高的功能
3. 请测试人员或质量保证工程师审查需求的可测试性
4. 开发一个协作的过程，以便设定需求优先级
5. 采用增量开发或分阶段发布产品，尽早将价值大最大的特性交付用户
6. 进行更多的市场调研，确定需要优先考虑的用户和市场，确定代表不同产品的产品代言人。
7. 对开发人员提供培训，教他们如何编写优秀的需求，在编写需求规格说明书时要避免使用主观的，不明确的术语。
8. 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
9. 跟踪每一个待确定的问题，直到问题得到解决

#### 2.2.6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

#### 2.2.7需求确认方面的控制

1. 需求过程一开始，今早让客户参与需求文档审查，明确用户的验收标准
2. 在需求获取期间让分析人员讨论非功能性需求，明确指定性能目标与质量属性

#### 需求变更方面的控制

1. 改进需求获取实践，实施并遵循一个变更控制流程，在接受变更前需对可能带来的风险进行分析并与客户沟通，成立变更控制委员会对提议的变更进行决策
2. 定义并交流项目范围，在需求获取活动中要有管理层参与；在制定进度计划时，要考虑意外情况并预留一定的时间；采用增量开发方法，快速响应新需求
3. 用范围陈述来确定所提议的需求是属于范围之内还是范围之外，记录下对某一提议的需求否认的理由
4. 重新协商项目的进度计划和所需资源
5. 为每个需求制定负责人，变更控制过程需要包括交流机制，需求交流要包括所有影响部门和涉众
6. 获得管理层的支持并让所有涉众都严格参与需求变更控制过程
7. 变更过程中要有不合适变更的影响分析
8. 将变更可能带来的风险传达给所有受影响的涉众，使用跟踪信息来评估提议变更的影响分析

#### 2.1.9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