|  |  |  |
| --- | --- | --- |
| **文件状态**  **[√] 草稿**  **[ ] 讨论稿**  **[ ] 正式发布** | **文档类型** | **软件测试计划** |
| **文件标识** |  |
| **版 本** | **1.0** |
| **作 者** | **刘嘉伟，高浩然** |
| **完成日期** | **2016.4.19** |

项目名称： Running cat

文档名称：软件测试计划

文档修订

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **更改人** | **描述（注明修改的条款或页）** |
| 1.0 | 2016.4.19 | 刘嘉伟 | 初始版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

批准人签字

|  |  |  |
| --- | --- | --- |
| **职务** | **姓名** | **日期** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**目录**

[1 引言 4](#_Toc7613)

[1.1 编写目的 4](#_Toc5511)

[1.2 背景 4](#_Toc21256)

[1.3 定义 5](#_Toc21213)

[1.4参考资料 5](#_Toc30082)

[2 测试计划 5](#_Toc1428)

[2.1 软件说明 5](#_Toc4619)

[2.2 测试内容（分模块测试） 7](#_Toc2193)

[2.3 测试类型 8](#_Toc7017)

[2.4 测试目标 9](#_Toc24853)

[2.5 测试资源 9](#_Toc12584)

[2.5.1测试工具 9](#_Toc19101)

[2.5.2 测试环境 9](#_Toc23144)

[2.5.3 测试人员计划 10](#_Toc27382)

[2.6 测试组织结构 10](#_Toc15401)

[2.6.1 角色和职责 10](#_Toc14314)

[2.6.2 测试估算 11](#_Toc20174)

[2.7 测试策略 12](#_Toc8987)

[2.7.1 功能测试策略  12](#_Toc29942)

[2.7.2 性能测试策略 13](#_Toc1557)

[2.7.3 接口测试策略 13](#_Toc9687)

[2.7.4 用户界面测试策略 13](#_Toc22433)

[2.7.5 安全性测试策略 13](#_Toc12227)

[2.8 测试计划 13](#_Toc15568)

[3 测试管理 14](#_Toc31614)

[3.1 培训计划 14](#_Toc23398)

[3.2 测试资源的跟踪和管理 14](#_Toc16264)

[3.3 进度的跟踪和管理 15](#_Toc16517)

[3.4 缺陷的跟踪和管理 15](#_Toc19090)

[3.5 测试完成标准 16](#_Toc1483)

[4 测试风险分析 16](#_Toc1713)

[4.1需求风险 16](#_Toc28733)

[4.2测试用例风险 16](#_Toc8972)

[4.3缺陷风险 16](#_Toc24504)

[4.4代码质量风险 16](#_Toc23)

[4.5测试环境风险 17](#_Toc23233)

[4.6测试技术风险 17](#_Toc6816)

[4.7回归测试风险 17](#_Toc8673)

[4.8沟通协调风险 17](#_Toc21552)

[4.9其它不可预计风险 17](#_Toc20222)

# 1 引言

## 1.1 编写目的

本测试计划文档作为指导此测试项目循序渐进的基础，帮助我们安排合适的资源和进度，避免可能的风险。本文档有助于实现以下目标：

1）确定现有项目的信息和应测试的软件结构。

2）列出推荐的测试需求

3）推荐可采用的测试策略，并对这些策略加以详细说明

4）确定所需的资源，并对测试的工作量进行估计。

1. 列出测试项目的可交付元素，包括用例以及测试报告等。

## 1.2 背景

随着人们对健康的重视，运动人群已成几何状递增。由此衍生的跑步类软件都在推出各自的特色功能。清晰的目标管理、专业的数据分析、丰富的训练计划、有趣的社交功能，令人眼花缭乱。很多人在尝试各种各样的app后，并没有真正的将跑步坚持下来，反而被各色各样的附加功能弄得晕头转向。

Running cat是一款能将宠物养成与跑步健身结合的一款app。软萌的宠物养成模式，清新简洁的风格更能够吸引青少年用户的眼球。

说明：

开发软件名称：Running Cat

项目开发者：北京理工大学软件学院软件项目管理开发小组第5组：

用户单位：暂未定

## 1.3 定义

主键（Primary Key）：每一笔资料中的主键都是表格中的唯一值。换言之，它用于独一无二地确认一个表格中的每一行资料。

外键（Foreign Key）：设表t1,t2中都有一个name字段，而且该字段是t1的主键，那么如果设t2中的name为外键的话，向t2中添加数据的时候，如果name值不在t1之中就会报错。

SQL: Structured Query Language（数据库结构化查询语言）。

## 1.4参考资料

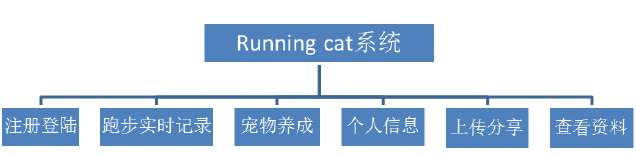
1. 《软件工程基础》 胡思康 清华大学出版社
2. 《软件需求》 劳森 (作者) 刘晓晖 (译者) 电子工业出版社
3. 《GBT 8567-2006 计算机软件文档编制规范》 中国质检总局
4. 测试使用的国家标准、行业指标、公司规范和质量手册等等
5. 《Running Cat项目开发任务书》 项目管理开发小组第5组
6. 《Running Cat软件需求规格说明书》 项目管理开发小组第5组
7. 《Running Cat软件概要设计文档》 项目管理开发小组第5组

# 2 测试计划

## 2.1 软件说明

Running Cat是一个具有基本跑步健身功能的移动互联网app，同时加入通过跑步养成宠物功能，用户还可上传分享。

软件整体框架：



本软件初步设计有五个界面。

1) 第一个界面是登陆界面，用户可以在登录界面输入账号和密码进行登陆，登陆一次之后就可以记住密码，第二次登陆时可以一键登录。

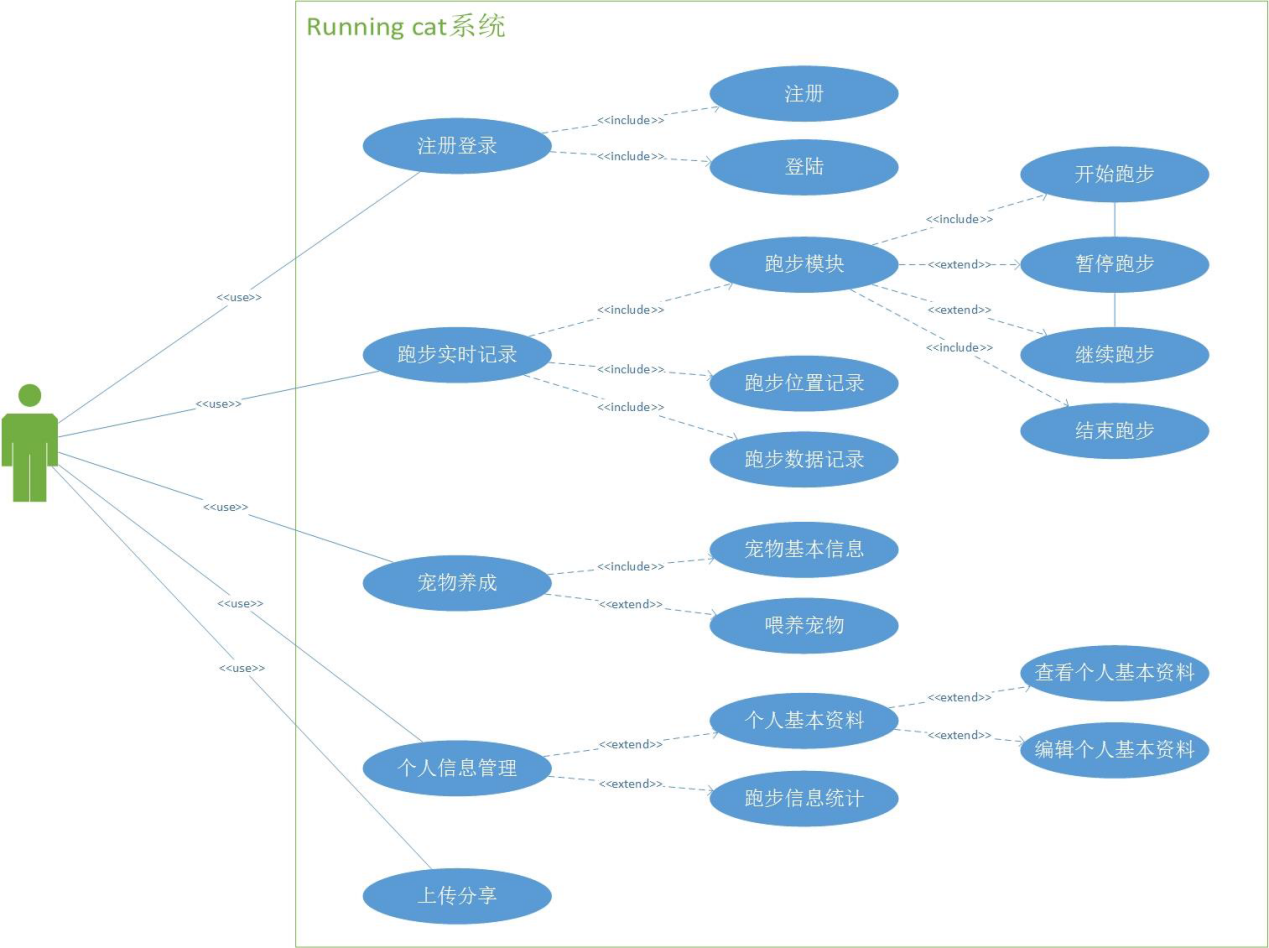
2) 第二个界面是注册界面，对于第一次使用本软件的用户需注册一个账号，从登陆界面的注册账号链接点进来，就可以注册

3) 第三个界面是登陆成功之后，进入主界面，其中会出现宠物挑选，其中有许多种类的小猫供用户挑选。用户可以根据自己的喜好选取一只进行饲养。以后本界面就会出现这只小猫的家，小猫会对主人撒娇和对话。同时旁边也会显示小猫的成长状态，没有猫粮是会题型跑步换取猫粮

4) 第四个界面是跑步界面，主界面右划就到了跑步界面。本界面会有一个地图，在跑步时地图上会出现用户的定位并且显示已跑里程，并且会自动将里程转化为猫粮

5) 第五个界面是好友和我的动态，跑步界面右划就到了我的界面。本界面会显示近期跑步里程的直方图，以日为单位，同时也可以查看好友排行，点开可以看到好友的跑步里程和宠物状态

系统用例图：



## 2.2 测试内容（分模块测试）

测试1： 名称：注册登录模块测试

目的：测试系统注册和登录模块

模块描述：

（1）注册：用户通过邮箱，输入密码进行注册，若注册邮箱合法且之前未被注册过，则注册成功，并获得一只宠物猫。

（2）登录：用户输入用户名和密码登录，查询数据库是否存在该用户，并且用户名和密码均正确，则登录成功。

测试内容：用户名、密码、身份输入、合理性检查、合法性检查，系统操作界面显示控制。

测试2：名称：跑步实时记录模块测试

目的：测试系统跑步实时记录模块

模块描述：

用户开始跑步，同时开始记录位置信息和实时数据。跑步过程中，用户可以选择暂停跑步（PAUSE），同时暂停记录数据，之后可以选择继续跑步，或者直接结束本次跑步。

测试内容：实时记录位置信息，数据同步，暂停/继续跑步功能。

测试3：名称：宠物养成模块测试

目的：测试系统宠物养成模块

模块描述：

新用户第一次注册登陆成功之后，进入主界面，其中会出现宠物挑选，其中有许多种类的小猫供用户挑选。用户可以根据自己的喜好选取一只进行饲养。以后本界面就会出现这只小猫的家，小猫会对主人撒娇和对话。同时旁边也会显示小猫的姓名和成长状态，即以柱状图和数值方式显示饥饿度、经验值、等级。在该界面也会以图像和数值的方式显示猫粮剩余量，当没有猫粮时会提醒跑步换取猫粮。

测试内容：新用户领取宠物，已有用户查看宠物，查询宠物属性，提示猫粮不足，给宠物喂食。

测试4：名称：个人信息模块测试

目的：测试个人信息模块

模块描述：

用户登录账号后，可进入个人信息模块，查看个人基本资料，也可编辑个人资料并保存修改。同时在该模块可以显示用户之前的跑步信息，主要包括：累计距离、累计时间、平均时速。

测试内容：显示个人资料，编辑/修改个人资料，显示跑步信息：累计距离、累计时间、平均时速。

测试5：名称：上传分享模块测试

目的：测试上传分享模块

模块描述：

该模块中有四个功能。第一个是查看好友列表，点进去以后可以看到已经添加的好友。第二个是好友动态，点进去可以看到好友的最新动态。第三个是排行榜，可以看到近期跑步里程以及宠物状态的排行。第四个是扫一扫，可以通过扫二维码的方式添加好友。

测试内容：查看好友列表，添加好友，查看好友动态，排行榜信息，扫码功能测试。

## 2.3 测试类型

|  |  |  |
| --- | --- | --- |
| 序号 | 测试类型 | 是否执行 |
| 1. | 功能测试 | 是 |
| 2. | 性能测试 | 是 |
| 3. | 用户界面测试 | 是 |
| 4. | 安装性测试 | 否 |
| 5. | 安全性测试 | 是 |
| 6. | 稳定性测试 | 是 |
| 7. | 兼容性测试 | 是 |
| 8. | 接口测试 | 是 |

备注：请在需要执行的测试类型之后填“是”

## 2.4 测试目标

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 领域 | 指标 | 指标值 | 组织级 | | 备注 |
| 上限 | 下限 |
| 质量 | 漏测率 | 0 | 10% | 0 | 系统上线后用户发现的缺陷/（测试发现的缺陷+用户发现的缺陷） |
| 测试用例有效性 | 100% | 100% | 80% | 测试用例发现的缺陷数量/总的缺陷数量 |
| 测试用例覆盖率 | 100% | 100% | 80% | 测试用例覆盖需求特征项的比率 |

## 2.5 测试资源

### 2.5.1测试工具

Quality Center9.2：测试管理工具，主要用于编写测试用例和进行缺陷的记录跟踪管理。

LoadRunner8.0：性能测试工具，主要用于性能测试。

### 2.5.2 测试环境

1）硬件测试环境

应用服务器使用HP rx4640 Itanium 2 CPU: ItaniumII 1.6G  内存：8G

系统要求运行在10/100M快速以太网。

 局域网通信协议使用TCP/IP，Internet通信协议使用HTTP。

2）软件测试环境

服务器端环境：

操作系统使用 安卓系统

数据库使用SQL Server 2012和MySQL

客户端环境：

操作系统使用 安卓系统

### 2.5.3 测试人员计划

|  |  |  |
| --- | --- | --- |
| 角色 | 技能及经验要求 | 数量 |
| 测试设计人员 | 熟悉测试技能，对系统功能有一定的了解 | 2名 |
| 测试执行人员 | 熟悉测试技能，对系统功能有一定的了解 | 2名 |
|  |  |  |

## 2.6 测试组织结构

### 2.6.1 角色和职责

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 人员 | 职责 | 备注 |
| 测试中心经理 | 颜振松 | 1. 负责与开发组之间的沟通，安排测试人员。 2. 对测试过程中遇到的问题进行协调沟通。 |  |
| 测试组长 | 刘嘉伟 | 1. 负责与开发组之间的沟通，接受项目经理提出的测试任务单，协调测试所需的资源和环境，并检查是否已经具备测试的条件。 2. 制定项目的系统测试计划。参与系统测试用例的评审。 3. 监督和跟踪整个系统测试的执行。 4. 测试的反馈。审核测试报告，在阶段测试结束时对测试活动进行总结。 |  |
| 测试工程师 | 高浩然 | 1. 负责根据系统测试计划和设计文档编写系统测试用例。 2. 参与对测试用例的评审，并按照评审要求对测试用例进行修改，直至通过。 3. 负责根据测试用例执行测试，对测试中发现的问题填写缺陷问题跟踪表，编写测试报告。 |  |

### 2.6.2 测试估算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阶段名称 | 任务名称 | 参与人员 | 工作量估计（人日） | 备注 |
| 第一阶段 | 编写测试计划、测试进度计划 | 2 | 1.5 |  |
| 编写测试用例 | 1 | 2 |  |
| 接收测试任务、功能测试、回归测试 | 1 | 2 |  |
| 编写测试总结报告、缺陷分析报告 | 2 | 1 |  |
| 第二阶段 | 接收测试任务、功能测试、回归测试 | 1 | 2 |  |
| 录制性能测试脚本，执行性能测试 | 1 | 2 |  |
| 编写测试总结报告、缺陷分析报告 | 2 | 1 |  |
| 编写出厂测试报告 | 1 | 1 |  |
| 编写用户手册 | 1 | 2 |  |

## 2.7 测试策略

根据前面在测试范围中列出的测试类型，在这里具体的描述对这种测试类型所要执行的测试策略。

### 2.7.1 功能测试策略

功能测试要覆盖系统中各个模块中的每个功能。

在功能测试时采用等价类划分法、边界值分析法和错误推测试法等多种测试方法，尽可能地做到测试的充分性。测试时既要输入正确的数据，测试功能是否满足，也要对每个功能中的每个数据输入域故意输入错误的数据，测试系统的健壮性。比如：要求输入字符的测试是否可以输入数值，要求输入数值的测试是否可以输入字符，是否有输入长度的限制、大小写的限制、最大最小值（边界值）的限制等等。

在每个功能测试通过的基础上，可以开始业务流程的测试，根据每个业务流程中所涉及到的多个功能，将各个相关的功能、模块按一定的顺序连接起来进行测试。

### 2.7.2 性能测试策略

在接收到性能测试任务后，测试组人员根据测试需求进行准备性能测试脚本、测试环境的搭建和基础数据的录入等准备工作。

性能测试往往针对软件的一部分功能，进行专项测试。比如，系统能承受的并发用户数量，系统的操作响应时间等。但却很难把握和度量，很容易导致测试过程很快结束或测试似乎永远都结束不了这两种极端，所以针对每一个测试过程要定义可度量的状态点，执行完一组工作后，及时检查是否已达到预定目标，是否已执行完该过程所有的步骤等，如实际情况与计划出入较大，应及时调整计划。

### 2.7.3 接口测试策略

通过验证提取数据的准确性进行系统接口的测试。

### 2.7.4 用户界面测试策略

用户界面测试包括以下内容：导航测试、图形测试、内容测试、表格测试、整体界面测试。

### 2.7.5 安全性测试策略

通过系统登录、密码保护等功能进行系统安全性测试。

## 2.8 测试计划

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 周 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 事件 | 第五周 | 第六周 | 第七周 | 第八周 | 第九周 | 第十周 | 责任人 | 关键里程碑 |
| 分析系统需求，准备系统测试相关工作 |  |  |  |  |  |  | 颜振松 | 确定测试方向 |
| 查阅软件文档编制规范，编写项目本软件测试计划 |  |  |  |  |  |  | 刘嘉伟 | 形成软件测试计划 |
| 根据软件测试计划，开展测试工作 |  |  |  |  |  |  | 刘嘉伟 | 开始测试 |
| 总结测试结果，形成软件测试报告 |  |  |  |  |  |  | 高浩然 | 完成软件测试报告 |

# 3 测试管理

## 3.1 培训计划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 培训的专题 | 培训方式 | 培训时间 | 参与角色 | 培训人 | 达成的目标 |
| 可靠性业务知识培训 | 内部培训 | 需求阶段前期 | 全体组员 | 高浩然 | 使团队所有成员了解领域内相关业务 |
| 生产相关业务知识 | 内部培训 | 一周一次 | 系统分析师、软件工程师、测试工程师 | 刘嘉伟 | 使团队的成员了解生产领域的其他相关业务知识 |
| 性能测试工具培训 | 演示 | 性能测试 | 测试人员 | 刘嘉伟 | 使团队成员更好地完成性能测试 |

## 3.2 测试资源的跟踪和管理

接收测试后，由测试组长制定进度计划，并将工作分配给测试组成员。测试组成员在完成工作后必须及时填报完成情况、进度安排的实际开始和结束时间。项目发生需求变更或者进度落后时，测试组长必须及时更新相关的测试进度安排以符合实际情况。

系统测试用例在项目组的QC库里进行编写。首先测试人员需要熟读并理解系统的需求，遵循《系统测试用例编写规范》上的具体要求，编写系统测试用例。以《需求规格说明书》需求特征项作为主要的依据，设计文档、界面原型等作为编写的辅助材料。

测试文档是测试工作开展的重要依据，也是整个项目文件的重要组成部分，所以必须做好评审工作。需要进行评审的测试文档至少为：系统测试计划、系统测试进度计划、系统测试用例；对测试文档的评审或走查分为两种情况，一种是项目组层面上需要对测试文档进行评审，

这种情况主要由项目组进行主导，具体的过程可参见《技术评审规程》；另一种是测试组内部的评审，主要是对测试用例的设计、编写情况进行内部评审，评审时要使用“系统测试用例检查表”。项目组对测试用例的评审是非常关键的，评审时要有熟悉需求和设计的人员参加，确保测试用例的正确性和完整性。

每周测试组长跟踪测试进度计划，有变化及时调整。

测试文档如系统测试计划、系统测试进度计划、系统测试用例的走查和评审。

需求变更和用户问题反馈的跟踪，测试培训计划的跟踪和管理。

## 3.3 进度的跟踪和管理

使用项目工作管理系统进行进度安排。根据开发进度计划制订测试进度计划。测试人员在完成工作后必须及时填写进度安排的实际开始和结束时间。当项目发生需求变更或者进度落后而改变开发计划时，也必须及时更新测试进度安排以符合实际情况。

## 3.4 缺陷的跟踪和管理

缺陷跟踪管理系统，必须注意权限分配的问题。缺陷记录作为软件开发过程中的重要数据，不能轻易被删除；对于已经关闭的缺陷，也不能随意进行修改。因此，缺陷跟踪管理系统必须设置严格的管理权限，非相关人员不得进行相应操作，修改相应数据。

## 3.5 测试完成标准

项目结项。

# 4 测试风险分析

在软件测试过程中，可能会出现以下9类风险，深入分析这些风险以及制订合理的规避策略对测试项目的顺利进行非常重要。

4.1需求风险

原因：对软件需求理解不准确，导致测试范围存在误差，遗漏部分需求或者执行了错误的测试方式；另外需求变更导致测试用例变更，同步时存在误差。

规避策略：测试人员应仔细阅读《Running Cat项目开发任务书》和《Running Cat软件需求规格说明书》，准确、深入、全面地理解本软件项目的需求，并确定合理的测试范围和测试用例。同时，制订需求变更的应对策略。

## 4.2测试用例风险

原因：测试用例设计不完整，忽视了边界条件、异常处理等情况，用例没有完全覆盖需求；测试用例没有得到全部执行，有些用例被有意或者无意的遗漏。

规避策略：测试人员应制订完整的测试用例，重视边界条件和异常处理等特殊情况，覆盖所有的需求；在测试执行过程中，要保证所有的测试用例都得到执行。

## 4.3缺陷风险

原因：某些缺陷偶发，难以重现，容易被遗漏。

规避策略：对于偶发且难以重现的缺陷，可以暂时忽略，并通过《软件测试报告》反馈。

## 4.4代码质量风险

原因：软件代码质量差，导致缺陷较多，容易出现测试的遗漏；

规避策略：制订相应的缺陷记录，记录遇到的所有缺陷，并通过《软件测试报告》反馈。

## 4.5测试环境风险

原因：有些情况下测试环境与生产环境不能完全一致，导致测试结果存在误差。

规避策略：测试人员应与开发人员做好沟通，争取使测试环境与生产环境一致；如做不到完全一致，需在测试过程中将这种差异纳入考虑。

## 4.6测试技术风险

原因：某些项目存在技术难度，测试能力和水平导致测试进展缓慢，项目延期。

规避策略：提前了解项目存在的技术难度，并制定好相应的应对策略。

## 4.7回归测试风险

原因：回归测试一般不运行全部测试用例，可能存在测试不完全。

规避策略：

## 4.8沟通协调风险

原因：测试过程中涉及的角色较多，存在不同人员、角色之间的沟通、协作，难免存在误解、沟通不畅的情况，导致项目延期。

规避策略：在测试过程中形成定期与开发人员交流沟通的机制，尽可能避免误解与沟通不畅。

## 4.9其它不可预计风险

原因：一些突发状况、不可抗力等也构成风险因素，且难以预估和避免。

规避策略：在本项目中，该类风险出现的几率极小，暂不考虑。