

# 风险分析及解决文档

## 目录

1、风险分析.....	1
2、解决方案.....	2

## 1、 风险分析

### 1.1、 团队内部风险分析

检查表 1-1 团队内部风险检查表

#### 团队内部风险检查表

- ☐ 团队成员技术背景能否在第四循环开始时达到此次软件开发的要求？
- ☐ 如果达不到，是否可以降低此次软件开发的要求？
- ☐ 是否能给出一个让团队成员在规定期限内学会所需知识的计划？
- ☐ 在团队成员知识不足以应对开发任务时，是否可以及时找到提高手段？
- ☐ 是否会出现团队成员学习任务重，导致没有足够的时间投入此次软件开发的情况？
- ☐ 如果出现，是否能够通过调整任务分配，使成员都能得到适当的开发任务？
- ☐ 多次调整任务后是否会造成开发任务分配失衡
- ☐ 开发进度差距是否会出现过大
- ☐ 是否有全体认可的团队沟通协调机制和条例？能否贯彻执行？
- ☐ 团队成员之间是否会出现性格不合、交流困难的情况？
- ☐ 是否有团队建设活动计划？
- ☐ 团队成员之间的信任是否随着开发过程的不断深入而不断加深？
- ☐ 是否能适当地给每个成员分配合适的开发任务？
- ☐ 是否有合理有效的方式检查每个成员的任务完成情况？
- ☐ 是否有全体成员认可的奖惩条例？能否贯彻执行？

### 1.2、 用户相关风险分析

检查表 2 用户相关风险检查表

#### 用户相关风险检查表

- ☐ 是否对“天天爱消除”原系统进行了充分的调查学习？
- ☐ 是否对相关用户的背景做了充分的调查了解？
- ☐ 是否选择了正确的用户代表？
- ☐ 与用户面谈是否采用了合适的面谈技巧？
- ☐ 面谈过程中，是否保持了话题的一致性？
- ☐ 是否设置了太多开放式问题，以至于用户的回答过于发散，甚至无法归纳出本质的需求？
- ☐ 是否设置了太多封闭式问题，以至于无法调动用户的积极性，甚至无法获得积极有效的信息？
- ☐ 是否主动引导用户回答问题，得到想要的信息？
- ☐ 是否从多个侧面设置问题，得到积极有效的信息？
- ☐ 对需求的内容是否进行了及时的记录并整理？
- ☐ 是否与用户确认了整理的需求获取结果，并获得他们的认可？

## 2、 风险解决

### 2.1、 开发团队内部风险解决

### 2.1.1 团队成员技术实力与开发任务所需有差距

解决方案：在需求开发和软件原型构建完成后，团队成员须认真分析自身技术实力的欠缺，找到不足之处，并制定学习目标计划，尽最大努力在第四循环开始时达到软件开发需求。如果没能达到，可在软件开发过程中由易到难进行开发，边开发边学习。若实在无法完成，课向技术经理或其他团队成员求助。当成员感到实在无法完成任务是，可申请重新调整开发任务，使大家都得到能力范围内的任务。如果项目开发难度超过团队能应付的极限，可通过团队会议讨论更改开发目标，保证此次软件开发能否继续进行。

### 2.1.2 团队成员没有足够的时间精力投入此次开发

解决方案：出现此种情况，其他成员若有若有余力，应在学业、生活上积极提供帮助，提高队友的学习效率，是其得到更多的时间投入开发过程。如果情况得不到改善，无时间的队员可向团队提出申请减少一部分开发任务，团队经过评审会酌情调整开发任务。如果团队中出现大范围的无足够时间和精力开发的情况，团队将通过会议决定适度调整开发进度目标。

### 2.1.3 团队成员交流沟通不畅

解决方案：（1）改善团队交流沟通计划，如加入更多的 coffee break、团队聚餐、团队活动和素质拓展等，让大家在项目开发之外有更多的交流活动，放松大家的身心，增加彼此的熟悉与亲切感，加强团队内部的交流。（2）制定团队的交流规范，如规定日常的交流方式以 QQ 群为主，文档的撰写严格按照《项目文档格式撰写规范》中的规定格式。

### 2.1.4 开发任务分配失衡

解决方案：出现此种情况后，应尽快查明原因。如果是由于任务分配不合理，应立即调整并重新分配；如果是团队成员推卸开发任务，应立即按照奖惩条例对其进行惩罚，并重新调整任务分配，督促其完成应完成的任务；如果是由于团队陷入“主程序员结构”，团队成员应立即反思，并立即调整并重新分配任务。

### 2.1.5 开发进度差距过大

解决方案：团队成员应按时汇报自己负责部分的开发进度，保证开发进度信息的透明。当有成员的开发进度有落后趋势时，团队应分析原因，如果是因为成员未能认真对待所分配的任务，那么应提醒并督促其抓紧完成所负责的任务；如果是因为其他原因，团队应结合实际得出解决方案，如重新调整任务分配。

## 2.2、 与用户有关的风险解决

### 2.2.1 对“天天爱消除”原游戏系统了解不够充分

解决方案：规定团队在需求获取前玩“天天爱消除”的累计时间限制。达到时间限制后在团队内部就“天天爱消除”这款游戏的优点、缺点等方面进行交流，并思考如果继承优点，改正缺点的初步构思。成员提出自己的困惑或看法，大家相互解答和质疑，不断挖掘原游戏系统的特点。

### 2.2.2 面谈的问题设置太过开放性或封闭性

解决方案：在面谈之前，团队应该先设计出面谈时要问的问题，并且团队内部应对这些问题进行评审，知道团队全体都认可这些问题是合适的。之后可再找团队外部的人进行测试，当大家都觉得测试结果可以之后，再进行面谈。

### 2.2.3 需求获取过程结束后仍没得到本质的用户需求

解决方案：（1）重新确认需求获取是否结束：确定已经让所有相关的用户代表都表达了自己的意见；用户已经没有更多的用例；用户想出的用例都是导出的用例；用户只是在讨论重复的问题；用户提出的新特性、需求都在项目范围之外。（2）如果确定需求获取过程结束，检查是否做到了一下几点：对需求获取过程进行了仔细的梳理和分析；细化了用户提出的抽象和模糊的需求；使用了多种方法表达获取到的需求信息；利用了不同的分析技术为相同的

需求进行建模，通过分析不同的关注点，考察需求是否完整，让真正的需求显露出来。