

表 1: 风险列表

#	风险陈述		发 生 概 率 (0 ~ 100%)	危害程度 (1 ~ 10)	应对方案	责任人
	情况	后果				
1	模块编码时间超出预期	项目不能如期完成	50%	7	定期召开例会，把握开发进度	曾 锴 鹏 (开 发 工 程 师)
2	产品定义不明确，项目实现阶段修改产品定义	项目大改，周期拖长	30%	9	在定义阶段召开组内会议，确定出明确的产品定义，并不在后续阶段动摇产品定义	李 语 焱 (项 目 经 理)
3	服务器部署时遇到困难	导致软件主要功能不能实现	30%	8	事先制定好备用方案	曾 锴 鹏 (集 成 工 程 师)
4	开发过程中遇到技术性困难	项目停滞不前，部分设计目标不能实现	30%	8	边开发边学习，查阅资料，请教专业人士	李 梦 宁 (开 发 工 程 师)
5	产品规模编码难度比预想的更大	周期延长	60%	4	全员加班	李 语 焱 (项 目 经 理)
6	测试阶段未能发现隐藏的漏洞	为软件正常运行埋下隐患，导致后续可能需要返工	40%	6	测试阶段合理设计测试方式，合理设计测试用例	李 嘉 鸿 (开 发 工 程 师)
7	库的功能和效果不能达到预期	开发难度增大，部分功能难以实现	30%	7	提前做好调研，选用熟悉的稳定可靠的库，提前做好备选方案	曾 锴 鹏 (开 发 工 程 师)
8	不熟悉开发环境和开发工具	开发效率较低，影响编码进度	50%	4	边开发边学习，组员互相交流技术问题	刘 彦 铭 (开 发 工 程 师)
9	实现阶段进一步扩展项目范畴	项目复杂度和编码耗时超过预期	60%	3	进入实现阶段前，明确产品需求，明确项目设计方案	吕 天 砚 (分 析 工 程 师)
10	项目开发任务分工不合理	开发效率低下	30%	6	在项目开发计划中明确团队组织结构以及各成员的项目角色	吕 天 砚 (分 析 工 程 师)

续表

#	风险陈述		发 生 概 率 (0 ~ 100%)	危害程度 (1 ~ 10)	应对方案	责任人
	情况	后果				
11	部分成员不能全力投入开发	开发进度受影响,周期延长	30%	5	合理安排进度,合理分配任务,各成员之间互相帮助	李 语 焱 (项 目 经 理)
12	开发工具、平台选择不当	增加实现难度,实现效果不佳	20%	7	前期充分调研,研究各开发工具的特点,权衡利弊	李 梦 宁 (开 发 工 程 师)
13	系统架构设计不合理	设计目标难以实现	20%	7	充分考虑系统的各种设计目标、与其他系统的集成、将来的维护需求、技术风险等因素	李 梦 宁 (开 发 工 程 师)
14	对象设计不合理	增加编码的难度,影响架构的灵活性等	20%	6	充分考虑软件的性能、可重用性、未来可能的变化等因素。兼顾设计的简洁性和架构的灵活性	刘 彦 铭 (开 发 工 程 师)
15	数据管理设计欠佳	影响程序读写性能,浪费存储空间等	20%	5	认真分析内存、文件和数据库等存储方式的特点,学习数据库相关技术	刘 彦 铭 (开 发 工 程 师)
16	运行设计不合理	影响软件性能	20%	3	认真分析并发需求,借鉴参照已有的类似项目的设计方案	吕 天 砚 (开 发 工 程 师)
17	前期部分源代码质量不高	代码越发臃肿,难以拓展新功能	10%	5	适当的重构部分代码	李 嘉 鸿 (开 发 工 程 师)
18	系统部署遇到困难	软件不能正常打包发布	10%	5	学习系统部署的方法,请教专业人士	曾 锴 鹏 (集 成 工 程 师)
19	代码管理出现问题	导致进度的损失回退	10%	5	编写过程中注意及时备份,采用科学的版本管理方法	李 语 焱 (项 目 经 理)
20	项目急需有特定技能的人	部分设计目标可能难以实现	10%	3	应事先提前联系好可能需要的相关具有特定技能的人员	李 语 焱 (项 目 经 理)