以下是可能出现的风险与解决方案：

（1）需求风险

1.需求已经成为项目基准，但需求还在继续变化；

2.需求定义欠佳，而进一步的定义会扩展项目范畴；

3.产品定义含混的部分比预期需要更多的时间；

4.在做需求中客户参与不够；

5.缺少有效的需求变化管理过程。

解决方案：事先写出产品的需求，对于有疑问的需求小组内进行讨论，讨论完毕得到结果之后进行整合。

（2）计划编制风险

1.计划、资源和产品定义全凭客户或上层领导口头指令，并且不完全一致；

2.计划是优化的，是"最佳状态"，但计划不现实，只能算是"期望状态"；

3.计划基于使用特定的小组成员，而那个特定的小组成员其实指望不上；

4.产品规模(代码行数、功能点、与前一产品规模的百分比)比估计的要大；

5.完成目标日期提前，但没有相应地调整产品范围或可用资源；

6.涉足不熟悉的产品领域，花费在设计和实现上的时间比预期的要多。

解决方案：由每个人写出对应的计划，小组内进行整合后讨论哪一些需要解耦，哪一些需要联合，最后确定每个人的工作是什么。

（3）组织和管理风险

1.仅由管理层或市场人员进行技术决策，导致计划进度缓慢，计划时间延长；

2.低效的项目组结构降低生产率；

3.管理层审查 决策的周期比预期的时间长；

4.预算削减，打乱项目计划；

5.管理层作出了打击项目组织积极性的决定；

6.缺乏必要的规范，导致工作失误与重复工作；

7.非技术的第三方的工作(预算批准、设备采购批准、法律方面的审查、安全保证等)时间比预期的延长。

解决方案：根据学校老师的实验流程进行实验，确定截止日期之后在截止日期之前完成任务。

（4）人员风险

1.作为先决条件的任务(如培训及其他项目)不能按时完成；

2.开发人员和管理层之间关系不佳，导致决策缓慢，影响全局；

3.缺乏激励措施，士气低下，降低了生产能力；

4.某些人员需要更多的时间适应还不熟悉的软件工具和环境；

5.项目后期加入新的开发人员，需进行培训并逐渐与现有成员沟通，从而使现有成员的工作效率降低；

6.由于项目组成员之间发生冲突，导致沟通不畅、设计欠佳、接口出现错误和额外的重复工作；

7.不适应工作的成员没有调离项目组，影响了项目组其他成员的积极性；

8.没有找到项目急需的具有特定技能的人。

解决方案：事先确定小组内部的成员并确定分工，这样就能保证分工明确，且学生在课程结束之前是不会轻易走人的。

（5）开发环境风险

1.设施未及时到位；

2.设施虽到位，但不配套，如没有电话、网线、办公用品等；

3.设施拥挤、杂乱或者破损；

4.开发工具未及时到位；

5.开发工具不如期望的那样有效，开发人员需要时间创建工作环境或者切换新的工具；

6.新的开发工具的学习期比预期的长，内容繁多。

解决方案：确定具体的开发环境，事先准备好需要的第三方库。

（6）客户风险

1.客户对于最后交付的产品不满意，要求重新设计和重做；

2.客户的意见未被采纳，造成产品最终无法满足用户要求，因而必须重做；

3.客户对规划、原型和规格的审核 决策周期比预期的要长；

4.客户没有或不能参与规划、原型和规格阶段的审核，导致需求不稳定和产品生产周期的变更；

5.客户答复的时间(如回答或澄清与需求相关问题的时间)比预期长；

6.客户提供的组件质量欠佳，导致额外的测试、设计和集成工作，以及额外的客户关系管理工作。

解决方案：事先与老师沟通确定自己的产品方案是否可行，不可行的话在一开始就需要更改。

（7）产品风险

1.矫正质量低下的不可接受的产品，需要比预期更多的测试、设计和实现工作；

2.开发额外的不需要的功能(镀金)，延长了计划进度；

3.严格要求与现有系统兼容，需要进行比预期更多的测试、设计和实现工作；

4.要求与其他系统或不受本项目组控制的系统相连，导致无法预料的设计、实现和测试工作；

5.在不熟悉或未经检验的软件和硬件环境中运行所产生的未预料到的问题；

6.开发一种全新的模块将比预期花费更长的时间；

7.依赖正在开发中的技术将延长计划进度。

解决方案：多次多人进行调试和测试，这样能够保证测试的时候不出bug。

（8）设计和实现风险

1.设计质量低下，导致重复设计；

2.一些必要的功能无法使用现有的代码和库实现，开发人员必须使用新的库或者自行开发新的功能；

3.代码和库质量低下，导致需要进行额外的测试，修正错误，或重新制作；

4.过高估计了增强型工具对计划进度的节省量；

5.分别开发的模块无法有效集成，需要重新设计或制作。

解决方案：事先确定每个人需要的部分，保证工作不重复，事先确定哪些部分需要整合，保证有一定的相关性。

（9）过程风险

1.大量的纸面工作导致进程比预期的慢；

2.前期的质量保证行为不真实，导致后期的重复工作；

3.太不正规(缺乏对软件开发策略和标准的遵循)，导致沟通不足，质量欠佳，甚至需重新开发；

4.过于正规(教条地坚持软件开发策略和标准)，导致过多耗时于无用的工作；

5.向管理层撰写进程报告占用开发人员的时间比预期的多；

6.风险管理粗心，导致未能发现重大的项目风险。

解决方案：遇到问题首先向组员提问，无法解决就找组长，再无法解决就找助教和老师来解决。