

迭代计划

制定日期：2022/02/27

组号	15	项目名称	在线文档工具
迭代名称	界面原型迭代	计划起止日期	2022/02/27-2022/03/14
任务、进度安排和人员分配：			
No	任务	起止日期	人员
1	调研、分析和定义需求	2022/02/27-2022/03/02	高立欣、闫涵、王乐峰
2	初选语言、工具和框架，并学习新技术	2022/02/27-2022/03/14	高立欣、闫涵、王乐峰
3	进行界面设计，实现界面原型	2022/02/29-2022/03/11	高立欣、闫涵、王乐峰
4	小组内部评审和改进需求文档和界面原型	2022/03/05-2022/03/11	高立欣、闫涵、王乐峰
5	编写《迭代评估报告》	2022/03/11-2022/03/14	高立欣、闫涵、王乐峰
预期成果： [列出本迭代计划交付的文档、模型、源代码、安装包等]			
文档： 《Vision 文档》 《软件需求规约》 《迭代评估报告》			
模型： use-case 模型			
源代码： 在线文档工具软件的全部源代码			
安装包： 未来将会开发能直接安装在用户电脑上的安装包			

主要的风险和应对方案:

[分析当前项目风险, 列出最大的 3~5 个风险, 自高到低排列, 并列出每个风险缓解和应急的措施。注: 随着项目的进行, 风险会不断变化, 故每个迭代要重新对风险评估和控制。

1. 进度。开发进度紧迫, 可能会牺牲质量换取时间。

应急措施: 1.详细分析需求。2.把优先级高的需求放在前面的迭代中实现, 宁可取消优先级低的需求, 也不牺牲质量。3.采用基于复用的软件开发方法, 在需求、设计、编码、测试和管理等多个方面复用已有的成果。4.建立切合实际的进度计划。

2. 需求误解。遗漏需求, 修改需求都会导致大量的返工。

应急措施: 1.采用问卷调查、访谈等多种方式开展充分的需求调研, 尽可能全面完整地获取需求。2.细致分析需求, 采用形式化语言进行需求建模。3.编写用户手册和测试用例。4.建立界面原型, 尽可能早地发现问题。5.与老师同学沟通, 尽早发现需求的缺陷。

3. 技术风险。开发人员对新技术的陌生会导致效率降低, 开发时间拉长。

应急措施: 向老师或助教咨询, 使项目组快速学习和掌握新的知识。

4. 缺乏足够的用户参与。开发人员同时担任测试角色, 可能会导致需求误解和更多的 Bug 未被发现。

应急措施: 1.列出所有项目干系人和用户, 进行分类, 选派合适的代表, 并确保代表的能力和空闲时间。2.让用户代表早期就直接参与到开发中, 参与需求的调研和评审, 对原型和发布版本进行测试和体验, 提供及时的反馈。3.用户代表和开发人员必须安排经常见面, 当面沟通。

5. 质量低劣。在进度紧张、预算和资源不够等情况下, 开发人员常常会减少质量保证措施, 牺牲质量。开发人员能力不足也是重要原因。

应急措施: 1.开展多层次的测试, 特别是易错模块应重点进行测试。2.对需求、设计、源代码、计划和测试用例等进行评审。3.采用迭代开发过程, 让测试提前进行, 及早发现问题。4.采用静态分析、质量审核、形式化证明等技术, 可以有效保证质量。