**迭代计划（简化版）**

制定日期：2020-09-15

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 14 | 项目名称 | PC Logo |
| 迭代名称 | 界面原型迭代 | 计划起止日期 | 2020.09.15-2020.10.05 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 通过讨论、调研，比较不同桌面程序框架的优缺点，了解相关技术，确定所采用的桌面程序开发框架 | 2020.09.15-2020.09.16 | 全体成员 | | 2 | 分析软件定位、前景等，学习撰写Vision文档的相关知识 | 2020.09.17-2020.09.18 | 苏浩然、吴天龙 | | 3 | 分析软件需求，学习如何制作use-case模型，学习如何撰写软件需求规约 | 2020.09.17-2020.09.18 | 吴泽一、吴泽川 | | 4 | 初步构想、设计软件界面原型 | 2020.09.17-2020.09.18 | 张哲溢 | | 5 | 初步撰写Vision文档（前半部分1-5节） | 2020.09.19-2020.09.20 | 苏浩然、吴天龙 | | 6 | 完成use-case模型的制作 | 2020.09.19 | 吴泽一、吴泽川 | | 7 | 初步完成软件需求规约的撰写 | 2020.09.20-2020.09.21 | 吴泽一、吴泽川 | | 8 | 用相关工具初步制作出界面原型 | 2020.09.19-2020.09.20 | 张哲溢 | | 9 | 初步撰写Vision文档（后半部分5-10节） | 2020.09.21-2020.09.22 | 苏浩然、吴天龙 | | 10 | 通过课上学习，对Vision文档进行检查重构 | 2020.09.23-2020.09.24 | 苏浩然、吴天龙 | | 11 | 将Vision文档初稿完成，小组内传阅评审 | 2020.09.25 | 全体成员 | | 12 | 根据组内成员的建议，对Vision文档进行修改，完成终稿 | 2020.09.26-2020.09.27 | 苏浩然、吴天龙 | | 13 | 完善，美化界面UI，形成初步的界面原型 | 2020.09.21-2020.09.22 | 张哲溢 | | 14 | 将界面原型在小组内传阅，小组成员提出修改意见 | 2020.09.23-2020.09.24 | 全体成员 | | 15 | 根据小组内建议，对界面原型进行进一步完善修改 | 2020.09.25-2020.09.26 | 张哲溢 | | 16 | 根据Vision文档，完善界面原型中应该展示的功能或非功能需求 | 2020.09.27-2020.09.30 | 张哲溢 | | 17 | 通过课上学习，进一步完善use-case模型和软件需求规约文档 | 2020.09.27-2020.09.30 | 吴泽一、吴泽川 | | 18 | 小组对界面原型进行最终评估、改进 | 2020.09.27-2020.09.30 | 全体成员 | | 19 | 小组对各项文档、模型进行评估、改进 | 2020.09.27-2020.09.30 | 全体成员 | | 20 | 编写《迭代评估报告》 | 2020.10.01-2020.10.08 | 张哲溢 | | 21 | 对修改后的界面原型、文档等进行最终评审 | 2020.10.09 | 全体成员 | | 22 | 学习所选择的技术栈内的相关技术 | 2020.09.27-2020.10.09 | 全体成员 | | | | |
| 预期成果：  需求相关文档与模型：《Vision文档》、use-case模型、《软件需求规约》  界面原型：界面原型设计、截图、代码等  迭代：《迭代计划》、《迭代评估报告》 | | | |
| 主要的风险和应对方案：   1. 进度风险。项目有可能因为时间问题难以在截止日期前完成迭代要求。缓解应急措施：加强团队沟通与交流，及时召开小组会议，及时制订迭代计划，估算团队开发速率和项目进度，及时修正迭代计划，严格按照迭代计划执行。当进度风险十分急迫时，可以采取增加工作时长的方法来应急。 2. 技术风险。小组成员有可能对相关技术了解不够充分，导致软件开发出现困难。缓解应急措施：提前学习相关技术，对团队成员的技术能力进行充分的预判，在使用不熟悉的技术前进行充分的学习、调试。在遇到难以自己克服的困难时，及时向助教、老师或者其他小组成员寻求帮助 3. 质量风险。项目有可能无法达成某项质量指标要求。缓解应急措施：严格执行符合要求的软件开发流程，认真进行code review，及时对产品代码进行评审，进行全方面的严格的测试，及时与需求方沟通并确认质量要求。在质量问题难以改善时，尽早采取其他架构，减少沉没成本。 4. 需求风险。小组成员有可能在最初工作开始时，对需求没有理解充分，需求有可能因为种种原因在项目开发过程中发生变更。缓解应急措施：及时地召开小组会议对产品需求进行评估，认真评估项目相关的需求文档，加强小组成员之间的沟通，对项目需求充分理解。采用合理的软件开发流程，采用演化模型等相关方式，以更好地应对需求变更。 | | | |