1. 实施计划
   1. 工作任务的分解与人员分工

|  |  |  |
| --- | --- | --- |
| **任务名称** | **负责人** | **参与人** |
| **获取需求** | 徐双铅（除需求访谈）吕迪（需求访谈） | **黄叶轩、陈俊仁、陈苏民、吕迪、徐双铅** |
| 编写项目视图与范围 |
| 用户群分类 |
| 选择产品代表 |
| 确定使用实例 |
| 召开应用程序开发联系会议 |
| 需求访谈 |
| 分析用户工作流程 |
| 确定质量属性 |
| 检查问题报告 |
| 需求重用 |
| **需求分析** | 陈俊仁 | **黄叶轩、陈俊仁、陈苏民、吕迪、徐双铅** |
| 绘制关联图 |
| 创建开发原型 |
| 分析可行性 |
| 确定需求优先级 |
| 为需求建立模型 |
| 编写数据字典 |
| 应用质量功能调配 |
| **需求规格说明** | 黄叶轩 | **黄叶轩、陈俊仁、陈苏民、吕迪、徐双铅** |
| 采用软件需求规格说明模板 |
| 指明需求来源 |
| 为每一项需求注上标号 |
| 记录业务规范 |
| 创建需求跟踪能力矩阵 |
| **需求规格审核** | 陈苏民 | **黄叶轩、陈俊仁、陈苏民、吕迪、徐双铅** |
| 编写测试用例 |
| 编写用户手册 |
| 确定合格的标准 |
| 审查需求文档 |

详见《PRD-2018-G15-GANT》

* 1. 接口人员

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **联系方式** | | | **接口联系人** |
| **联系电话** | **邮箱** | **地址** |
| 杨枨 | 13357102333 | [yangc@zucc.edu.cn](mailto:yangc@zucc.edu.cn) | 理四504 | 徐双铅 |
| 侯宏仑 | 13071858629 | [houhl@zucc.edu.cn](mailto:houhl@zucc.edu.cn) | 理四501 | 陈苏民 |

* 1. 进度

详见《PRD-2018-G15-GANT》

* 1. 预算

无

* 1. 关键问题

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **风险介绍** | **风险类型** | **应对优先级** | **应对措施** | **影响等级** | **可能性等级** | **风险标识** |
| 成员因故请假 | 参与者 | 高 | 提前改变任务的分配，他人顶上 | 高 | 高 | R1 |
| 项目成员不能实现项目 | 技术 | 中 | 制定培训计划 | 低 | 中 | R2 |
| Git远端仓库崩溃 | TBD | 高 | 及时发现，用本地版本去创建新的远端仓库 | 高 | 低 | R3 |
| 与干系人联系邮件发送内容、格式错误 | 任务 | 高 | 提前Deadline发邮件，抄送组员，即使发现错误并修正 | 中 | 中 | R4 |
| 项目文件结构不符合要求 | 任务 | 高 | 配置管理员修改文件结构 | 中 | 低 | R5 |
| 对接下来的计划和任务定义不够充分明确 | 任务 | 高 | 找任务发布者（老师）明确任务，并制定一周的计划，每个组员都要有事可做 | 高 | 显著 | R6 |
| 组内信息回复的实时性 | 参与者 | 中 | 组内QQ群的信息要经常看，也要记得回复 | 中 | 中等 | R7 |
| 教学辅助网站开发经验不足 | 参与者 | 中 | 去找标杆 | 中 | 中等 | R8 |
| 成员空余时间有不确定性 | 参与者 | 高 | 在开会说明接下来一周的行程，提前请假，安排工作表 | 高 | 显著 | R9 |
| 团队成员的能力（包括业务能力和技术能力）和素质，对项目的进展、项目的质量具有很大的影响 | 参与者 | 中 | 在用人之前先选对人、开展有针对性的培训、将合适的人安排到合适的岗位上 | 中 | 中等 | R10 |
| 团队成员是否能齐心协力为项目的共同目标服务 | **参与者** | 低 | 项目在建设之初项目经理就需要将项目目标、工作任务等和项目成员沟通清楚，采用公平、公正、公开的绩效考评制度 | 低 | 中等 | R11 |
| 管理工具、开发工具、测试工具等是否能及时到位、到位的工具版本是否符合项目要求 | **工具** | 低 | 在项目的启动阶段就落实好各项工具的来源或可能的替代工具，在这些工具需要使用之前（一般需要提前一个月左右）跟踪并落实工具的到位事宜 | 低 | 低 | R12 |
| 对方法、工具和技术理解的不够 | **技术** | 高 | 每个人熟悉一种工具（①黄：  project的熟悉与教学；②陈： 熟悉需求管理工具与教学；③吕： 熟悉Axure rp ；④徐： 熟悉UML建模工具与教学  ；⑤陈：  git） | 高 | 显著 | R13 |
| 界面原型不被用户认可 | **参与者** | 高 | 采用快速的手工画图，让用户确认并签字或录音 | 高 | 高 | R14 |
| 组员生病请假或者其他方式离开工作岗位 | **结构** | 中 | 设置替补人员 | 高 | 低 | R15 |
| 电脑硬件不稳定造成文档丢失 | **技术** | 高 | 巧用GITHUB，qq,百度网盘等工具 | 中 | 低 | R16 |
| 组员考评不公平造成内部矛盾 | **参与者** | 中 | 加强共同，完善考评制度，以项目经理为中心 | 低 | 高 | R17 |
| 用户对界面原型有了天马行空的全新的提议 | **参与者** | 高 | 加强与技术人员的同步沟通，确认工作量与可行性 | 高 | 低 | R18 |

* 1. 配置管理计划
     1. 版本命名策略
        1. 版本格式

每一个文档的版本格式为[主版本号.子版本号.修正版本号。

示例：0.1.1

文档的初始版本为0.1.0。

* + - 1. 版本更新

当文件内容有了重大的变化或改进，主版本号加一。

当文档的内容有了模块的增加、补充等，子版本号加一。

当文档的内容有了小修改，如修正了纰漏等，修正版本号加一。

* + 1. Git使用策略
       1. 基础知识

在使用之前，我希望每位组内成员都能明白git的基本用法与术语，在此，我对几个关键术语做出解释，如果不能理解，可以网上搜索资料或者问陈。

* 仓库：可以简单的理解为一个文件夹
* 多版本：如果一个仓库是多版本的，那么我们可以随时把它切换成某个时间段的某个样子，即不同版本。
* 分支：一个文件目前是A状态，甲将这个文件从A状态修改到了B状态，乙将这个文件从A状态修改为了C状态，那么从A这个时间点分叉出了两个不同版本（B、C）,即分支。
* 分支合并：将B、C两个状态相对于A的改动合并到一起。注意，如果B、C对于A都只是增加内容，那他们可以轻易的合并到一起，如果对同一个部分有了修改操作，会造成“冲突”，需要人工合并，应该尽力避免这种情况。
* 远程仓库：即我们放在码市或者github的仓库，对于组员来说是共用的，上面的内容大多数应是可发行的版本（做完的）。
* 本地仓库：就是你自己电脑上从远程仓库克隆下来的文件夹，如果你只是在本地做了修改，是不会影响远程仓库的，其他组员是看不到你做了什么的，除非你push了改动。
* 远程分支：即远程仓库上不同的分支，所拥有的不同版本，对所有组员可用。
* 本地分支：你为自己在本地的仓库建立的分支，你可以选择是否push它，使它成为远程分支。
* push：将本地仓库的改动（包括你建立的本地分支）推送到远程仓库上，使其他组员也能看到你的修改。
* pull：将远程仓库上的内容同步到本地仓库上。
* fetch：可以检测出远程仓库对于你的本地仓库有哪些更新。
* master分支：主分支，上面的所有内容应保证是可用的、可发行的。
  + - 1. 注意点
* push之前请先fetch，看看远程仓库目前是不是最新版本，如果是的话先pull下来，再push，防止冲突。
* 对于push时，备注应该详细，比如对哪些文件的哪些部分做了何种修改，而不要笼统的说修改了某个文件
  + - 1. 使用场景
* 首先创建一个组织，命名为PRD2018。然后建立2个team，一个命名为Admin，一个命名为Member设置Admin的权限为admin，设置Member的权限为write。配置管理员所在team为Admin,然后将组员拉进Member。
* 建立一个仓库，命名为PRD2018-G15。Master分支放入整个项目过程基础的文档，包括受控文档与非受控文档。在master分支的基础上，创建5分支，命名规则为：G15-小组成员名字，作为每个组员的工作区域。
* 每一周组长分配好任务，将模板上传到master分支的非受控文档的公用文件夹里，组员在自己的分支工作区域下，将任务完成，完成，push一下上传到github上。
* 组员将完成的最终版本上传到master分支的非受控文档的自己名称的目录下，并push一下上传到github上，组长pull下每个人的任务，然后进行评审，如果组长不同意，组员需要在进行修改，然后更新下每个人的文档。最终组长同意了，然后组长整理发送Emile给老师,配置管理员上传到受控文档里

**注意事项**：

* 为了方便，组员只需要管理好自己的工作区域，master的上传又配置管理员操作。
* 每一次更改文档、或者上传文档时，需要Fetch origin来同步一下git，保证不出错，防止引起冲突。