**MarkII实验项目工程管理与过程控制文档**

《软件工程管理与经济》课程项目文档

二〇二三年六月

虚拟仿真实验软件规模度量MARKII方法

**2053171白钰**

**2053412刘怡然**

**2051840梁厚**

**2053512孙博闻**

同济大学

Tongji University

软件学院 软件工程

**引言**

**项目背景**

随着互联网的快速发展和教育技术的进步，网上实验教学平台逐渐成为教育领域的重要组成部分。传统的实验教学方式存在诸多限制，如实验设备受限、时间和空间限制等，无法满足学生的需求。

同济大学作为一所知名高等教育机构，致力于提供高质量的实验教学。然而，传统的实验教学方式存在诸多挑战，包括资源有限、时间安排不灵活以及学生参与度不高等问题。为了解决这些问题，同济大学决定开发一个网上实验教学平台，用以帮助软件工程管理与经济课程的教学，以提供更灵活、便捷和互动性强的实验教学体验。

**项目概述**

我们的项目是一个网上实验教学平台，专注于为学生提供实验报告填写和实验指导书下载的功能。该平台的诞生旨在解决传统实验教学中存在的种种不便和限制。通过该平台，学生可以方便地填写和上传实验报告，并从平台上获取实验指导书，从而实现在线化、数字化的实验教学。

我们的虚拟仿真实验平台提供了一个用户友好的界面，使学生能够轻松浏览和下载实验指导书。实验指导书详细介绍了每个实验的目的、步骤和相关要求，帮助学生全面了解实验内容和要点。学生可以根据指导书进行实验操作，并在实验完成后通过平台填写和上传实验报告。

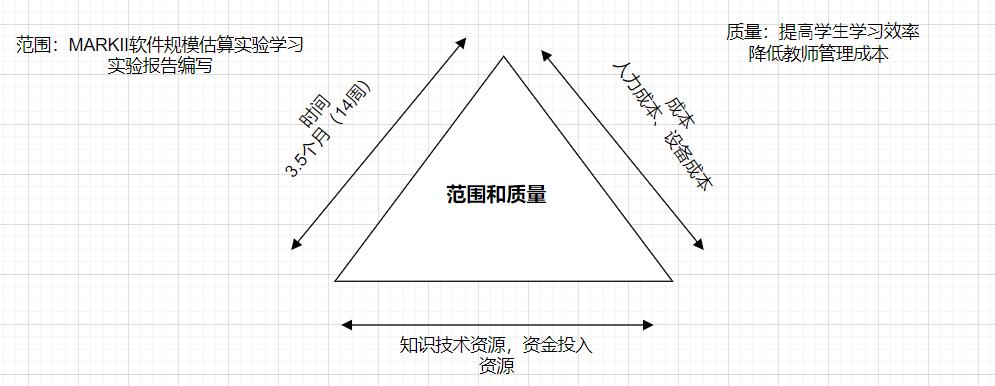
该平台的必要性在于提供了一种更灵活、便捷的实验教学方式。学生可以在任何时间、任何地点访问平台，下载实验指导书，了解实验要求，并通过平台填写和上传实验报告。这消除了传统实验教学中时间和空间的限制，提供了更多自主学习的机会。同时，教师可以通过平台及时查看学生的实验报告，进行评估和反馈，促进学生的学习和成长。

我们希望通过这个项目，为学生提供一个方便、高效的实验教学平台，激发他们的学习兴趣和实践能力。同时，该平台也为教师提供了一个管理实验教学的工具，简化了实验报告的收集和评估流程。我们相信，这个网上实验教学平台将为教育领域带来更多的创新和发展。

1. **项目范围**

1.1 **项目范围三角形**

为在宏观层⾯上对本项⽬进⾏较粗粒度的事前估计，了解本项⽬在范围、质量、成本、时间、资源、⻛险等⽅⾯的约束条件，在项⽬范围界定阶段，本项⽬组在与客⼾进⾏初期讨论后绘制了如下的基本处于平衡状态的项⽬范围三⻆形图，期望在项⽬开展后的各个⾥程碑阶段对本项⽬的各约束条件进⾏持续性检查，保证本项⽬在整个⽣命周期内处于基本平衡状态：



1.2 **项目POS（Project Overview Statement）**

下表是本项⽬组在项⽬范围确定阶段向客⼾提交的POS，该表格包含了本项⽬的开展⽬的、实施⽬标、项⽬进⾏过程中可能遇到的问题和机遇等项⽬宏观层⾯上的描述，帮助⽤⼾快速了解本项⽬组的⼯作内容，项⽬的具体范围，以及预期达到的结果，增强本项⽬组与客⼾的沟通。

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Overview Statement** | **项目名称** | **项目标识** |  |
| 虚拟仿真实验平台-基于MARKII的软件规模度量实验 | 42-MARKII |  |
| **风险：**   1. 技术难题：该实验模块的开发可能遇到技术上的挑战，例如算法设计、数据处理、平台兼容性等。这些技术难题可能导致开发延迟、功能缺陷或性能问题。 2. 用户接受度：用户（同济大学教师和学生）对该实验模块的接受度是项目成功的关键因素。如果用户对新平台和实验模块的接受度较低，可能导致使用率不高，影响实验教学的效果。 3. 项目进度延误：由于各种原因（如技术难题、资源限制或沟通问题），项目可能面临进度延误的风险。延误可能导致项目交付延迟、成本增加和用户满意度下降。 4. 资源限制：项目所需的人力、物力和财力资源可能受到限制，这可能影响项目的开发和实施计划。 5. 变更管理：项目需求、范围或技术可能发生变化，而变更管理不当可能导致项目目标的偏离、质量问题和冲突。   **机会：**   1. 技术创新：项目实施过程中可能出现技术上的创新和突破，例如新的算法、工具或方法，这有助于提升实验模块的性能和功能。 2. 用户参与度：通过积极引入用户（同济大学教师和学生）参与项目开发和测试，可以增加用户对平台的认同感和满意度，提高用户接受度。 3. 合作机会：与其他团队合作开发该实验教学平台的项目，可能带来合作机会和资源共享，加快项目进展并提升项目成果的质量。 |  |  |  |
| **目标：**   1. 提供便捷的实验报告填写和实验指导书下载功能，使学生能够在任何时间、任何地点进行实验教学。 2. 提供规范的MARKII实验流程指导，自动计算软件规模，提高学生的实践能力和自主学习能力，通过在线实验教学促进他们的学习兴趣和成果。 3. 简化教师管理实验教学的流程，减轻教师的工作负担，使教学过程更高效。 |  |  |  |
| **实现功能：**   1. 实验报告填写：  * 学生应能够在平台上填写实验报告，包括实验的目的、步骤、结果和分析等内容。 * 系统应支持实验报告的保存、编辑和提交功能，以便学生能够随时修改和提交报告。  1. 实验指导书管理：  * 教师应能够上传和发布实验指导书，包括实验的背景介绍、操作步骤和相关要求等内容。 * 学生应能够浏览和下载实验指导书，以便全面了解实验内容和要求。 |  |  |  |
| **成功标准：**  实现全部功能点  上线后稳定运行，满足师生教学需要   1. 学生满意度达标：通过平台调查和反馈，衡量学生对该实验平台的满意程度和使用体验。 2. 学生实践能力提升：通过评估学生在实验教学中的实践能力提升情况，包括实验报告的质量和内容。 3. 教师工作效率提升：通过统计教师使用平台管理实验教学所需的时间和工作量的减少程度。 4. 学生学习成果：通过比较使用平台进行实验教学和传统实验教学的学生学习成果，评估平台对学生学业成绩的影响。 5. 教学成本降低：通过对实验教学过程中所需资源和成本的评估，比较使用平台和传统实验教学方式的成本效益。 |  |  |  |
| **交付物：**  **代码：**   1. 基于vue3.0框架的前端页面。 2. js语言编写的实验报告填写模块、实验结果计算模块和实验原理与实验步骤模块的代码：包括前端代码、表单设计、数据验证和提交操作等的实现代码。   **文档：**   1. 项目宪章 2. 项目中期答辩PPT 3. 项目技术文档（包括需求分析、概要设计和详细设计等内容） 4. 项目过程管理文档（本文档） 5. 项目经济学文档 |  |  |  |
| **关键假设：**   1. 学生具备一定的计算机和网络技能，能够使用虚拟仿真实验平台进行实验报告的填写和上传。 2. 学生对在线实验教学具有积极的接受态度，并愿意主动使用虚拟仿真实验平台进行学习和实践。 3. 虚拟仿真实验平台的安全性和稳定性能够得到有效保障，以保护学生和教师的个人信息和数据。   **约束：**   1. 平台的开发和维护需要足够的技术支持和资源投入，包括人力、硬件设备和软件开发成本。 2. 虚拟仿真实验平台的功能和性能应该与学校的实验教学需求相匹配，以满足学生和教师的实际需求。 3. 学生和教师需要进行一定的培训和指导，以熟悉和正确使用虚拟仿真实验平台，确保平台的有效使用和实验教学的质量。 |  |  |  |
| **2053171白钰**  **2053412刘怡然**  **2051840梁厚**  **2053512孙博闻** | **日期**  **2023.3.25** | **审批人** | **日期** |

1.3 **开发模式**

经过讨论，我们采用敏捷开发Scrum框架。

通过我们的项目章程，我们可以知道项目的大体需求已经确定。但是项目的利益相关者对项目的最终实现程度不够明确，需要对项目进行密切关注。此外。整个项目小组的成员在地域上高度集中，能够支持高频的敏捷开发会议的进行。最终，我们选择以敏捷开发中的Scrum进行开发。通过Scrum的回顾会议，邀请利益相关者参加使用当前的Sprint的结果，并提出更为修改和制定下一阶段的需求。Scrum的优势在于：组织团队与客户紧密协作，通过高度迭代式、增量式的软件开发过程响应变化，并在每次迭代结束时交付经过编码与测试的有价值的软件。

需要注意的是，为了适应实际情况，我们对Scrum的会议做了一定的修改，使它更加可行。

* 我们将每日Scrum会议修改成了每周Scrum会议，因为我们并不是每天都会分配时间给本项目
* 为了提高效率，我们将每一次冲刺规划会议、冲刺评审会议和冲刺回顾会议合并，这样，在一个冲刺阶段结束时首先进行冲刺评审，演示冲刺期间的软件增量（增量评审在组外会议进行），之后回顾冲刺过程，优化开发实践。在会议的最后，我们会选择新的冲刺待定项、确定交付物及交付时间、角色和分工。
* 产品负责人黄杰老师等利益相关者和我们的时间安排不容易协调，因此我们在冲次会议之外额外安排时间与黄杰老师交流，也就是说冲刺规划会议、冲刺评审会议和冲刺回顾会议分成组内和组外两个时间线分别进行

2. **项目计划**

项⽬计划阶段的⼯作是根据上⼀阶段确定的项⽬范围与⽤⼾需求，对项⽬⼯作进⾏进⼀步分解，获得可执⾏的项⽬活动，并依据项⽬活动结合本项⽬的特点对项⽬的持续时间进⾏估计，最后合理安排项⽬活动的起⽌时间，以便更好的跟踪与管理项⽬进度。

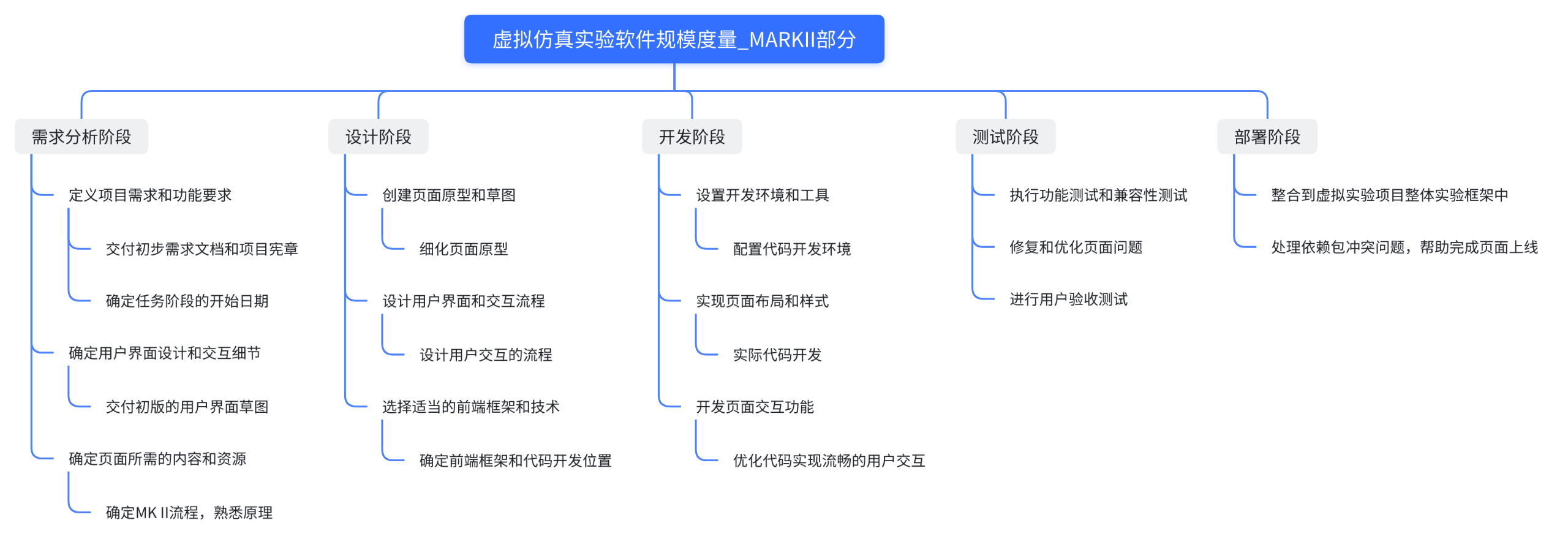
良好的项⽬计划不仅能够减少项⽬开展过程中的不确定性，增进项⽬组成员之间、项⽬组与客⼾之间的理解，还能有效提⾼项⽬开展的效率。在这⼀阶段，本项⽬组⾸先对上⼀阶段产出的RBS进⾏进⼀步分解与细化，获得能够直接落实到具体项⽬⼯作活动中的WBS，然后根据WBS，综合采⽤了德尔菲技术与三点技术对本项⽬的持续时间进⾏了估计。最后，本项⽬组基于项⽬时间估计与WBS的分解绘制了本项⽬的⽹络拓扑图。

在完成对项⽬⼯作进⾏进⼀步分解，我们制定了详细的工程管理与过程控制计划。

2.1 **工作分解结构**

工作分解结构（WBS）以[可交付成果](https://baike.baidu.com/item/%E5%8F%AF%E4%BA%A4%E4%BB%98%E6%88%90%E6%9E%9C/6780498)为导向，对项目要素进行的分组，它归纳和定义了项目的整个工作范围，每下降一层代表对项目工作的更详细定义。WBS总是处于计划过程的中心，也是制定进度计划、资源需求、成本预算、风险管理计划和采购计划等的重要基础。

WBS可以把整个工作分解为更小、易于管理的部分。这些部分可以进行合理评估，并分配给团队。工作分解结构可以识别活动，我们将整个项目分解为以下部分，并以此创建了WBS。



WBS是围绕整个管理体系组织起来的，将项目分解为需求分析，设计，开发，测试和部署等五个个Sprint的任务。根据WBS并利用所有可用资源，我们可以创建工作计划。并以甘特图和网络图的形式表现。

为保证WBS得到准确且完整的定义，本项⽬组利⽤如下的六个WBS完整性指标对其进⾏检验：

• 活动状态和完成情况可测量；

• 活动有边界；

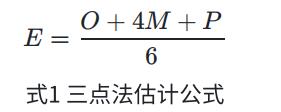
• 活动有可交付成果；

• 活动的费⽤与时间容易估算；• 活动⼯期在可接受期限内；

经检验，WBS中的每项活动都得到了正确的分解，适合的WBS保证了项⽬的顺利进⾏。

2.2 **项目持续时间估计**

在项⽬持续时间估计中，本项⽬组基于WBS，综合使⽤德尔菲技术和三点技术，对最可能情况、 最悲观情况，最乐观情况都进⾏了三轮预测估计，最终取三种情况的第三轮估计值带⼊如下的公式中进⾏计算，得到了项⽬预测持续时间为98天，即14周。





2.3 **项目网络拓扑图**

以下为网络图，红色展示了最优路径

**[该类型的内容暂不支持下载]**

2.4 **项目管理计划概述**

在确定了项目范围，并对在项目计划阶段对项目工作进一步分解后，我们制定了完整的项目管理计划。

我们的计划具体包括以下管理模块：

|  |  |  |
| --- | --- | --- |
| **序号** | **管理模块** | **内容** |
| 1 | **团队管理** | 包括人员招募、组织结构、运作规则和管理范围 |
| 2 | **项目相关方管理** | 识别相关方、确定相关方参与方式和沟通需求、利益管理策略 |
| 3 | **沟通管理** | 确定了沟通频率、沟通方式和沟通内容 |
| 4 | **时间管理** | 3月20日制定初版时间管理计划表，4月6日会议后扩充为40+任务列表 |
| 5 | **资源管理** | 包括资源识别、资源的分配与调度 |
| 6 | **过程监控与控制** | 项⽬的监督与控制从项⽬启动后便开始，直到项⽬结束才结束，这⼀活动属于整个项⽬的庇护性活动。  在项⽬的监督与控制中，本项⽬组⾸先绘制了本项⽬的过程质量矩阵与区域图，然后本项⽬组采⽤持续过程改进模型并配合⾥程碑趋势图、 ⽢特图、挣值分析图、⻥⻣图等多种图形化监控⼯具对项⽬进⾏监控。 |
| 7 | **质量管理** | 3月25日制定了项目的质量目标（goal）和标准（criteria）管理原则和活动  4月6日制定了代码审查、UI评估、软件测试的原则、流程和要求等具体标准  进入开发阶段后，文档中展示了我们对软件质量的具体管理措施 |
| 8 | **变更管理** | 3月25日制定初版变更管理，4月6日会议后再版  文档中以变更实例展示了我们变更管理的具体流程 |
| 9 | **风险管理** | 制定⻛险管理计划；  制定了适⽤于本项⽬的风险识别模板；•  编写⻛险评估表；  对风险进行监控 |
| 10 | **验收管理** | 在文档的最后，我们展示了具体的验收计划和实际交付过程+验收结果 |

3. **项目团队管理**

3.1 **人员招募**

在收到项目要求后，我们在软件工程管理与经济班级群中发消息组队，建立了一个四人组织，成员全部来自于同济大学软件学院。本次小组成员最终确定为：白钰（2053171），刘怡然（2053412），梁厚（2051840），孙博闻（2053512）

3.2 **组织结构**

**项⽬经理：**统筹安排协调项目全局，采⽤轮值制，顺序如下：

|  |  |  |
| --- | --- | --- |
| 姓名 | 时间 | 联系⽅式 |
| 刘怡然 | 1-5周 | 907021907@qq.com |
| 梁厚 | 6-9周 | 1306715855@qq.com |
| ⽩钰 | 10-13周 | 1845201056@qq.com |
| 孙博闻 | 14-17周 | 1137248567@qq.com |

**团队成员**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 角色 | 职责 |
| 2053171 | 白钰 | 质量管理组长 | 度量软件质量，监督项⽬过程中的软件质量，提⾼软件质量，增进质量规程。 |
| 2053412 | 刘怡然 | 需求收集副组长 | 收集⽤户需求，不断迭代，作为⽤户和开发团队的桥梁 |
| 2051840 | 梁厚 | 需求收集组长 | 收集⽤户需求，不断迭代，作为⽤户和开发团队的桥梁 |
| 2053512 | 孙博闻 | 风险管理组长 | 识别使项⽬处于⻛险中的因素，排除⻛险，量化⻛险对项⽬的影响等。 |

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 角色 | 职责 |
| 2053171 | 白钰 | 研发工程师 | 参与MARKII软件规模估算实验的前端研发 |
| 2053412 | 刘怡然 | 研发工程师 | 参与MARKII软件规模估算实验的前端研发 |
| 2051840 | 梁厚 | 测试工程师 | 统筹安排协调项目测试进程，对软件进⾏单元测试、系统测试、交付测试等。 |
| 2053512 | 孙博闻 | 运维工程师 | 负责保证项目正常运行，维护项目运行环境 |

3.3 **团队运作规则**

教材上将团队运作规则⽅分为了六⼤块：

解决问题、制定策略、解决冲突、达成⼀致、头脑⻛暴、团队会议。

根据本项目的特点，我们将在本文档中着重阐述以下三方面的规则：做出决策、解决冲突、团队会议。

3.3.1 **做出决策**

1. 明确决策过程：

* 确定决策的标准和目标，明确团队在决策过程中的优先权和约束条件。
* 确定决策的流程和步骤，包括信息收集、分析、讨论和最终决策的确认。

1. 确定决策责任人：

* 在每个决策中指定一位责任人（当前轮值项目经理），负责组织和推动决策的过程，并最终做出决策。
* 责任人应具备决策能力、沟通能力和公正性，能够全面考虑各方利益和意见。

1. 充分信息收集：

* 在做出决策之前，收集和整理与决策相关的充分信息，包括需求、约束、资源等。
* 确保团队成员有足够的时间和机会了解和分析相关信息。

1. 推动讨论和辩论：

* 我们项目团队为“民主型”，鼓励团队成员进行积极的讨论和辩论，就不同的选择和方案进行深入的思考和交流。
* 提供一个安全的环境，让团队成员可以自由表达自己的意见和担忧。

1. 考虑共识和多数意见：

* 在决策过程中，努力达成共识，尊重多数意见，避免强行推行个人意见。
* 如果无法达成共识，责任人有权做出最终决策，但需要确保对少数意见的合理考虑和解释。

1. 及时沟通和执行决策：

* 在决策做出后，及时将决策结果和理由沟通给团队成员，并确保决策得到有效执行。
* 提供适当的支持和资源，帮助团队成员理解和执行决策。

1. 定期评估和反馈：

* 定期评估已做出的决策的效果和结果，收集团队成员的反馈和意见。
* 根据反馈和经验教训，对决策过程进行改进和调整，以提高决策质量和效率。

3.3.2 **解决冲突**

1. 提倡开放和尊重：

* 鼓励团队成员之间开放的沟通和讨论，尊重每个人的意见和观点。
* 建立互相理解和支持的氛围，使每个人都感到被尊重和听取。

1. 设立有效的沟通渠道：

* 创建一个团队共享的QQ群，用于及时交流和讨论。
* 定期组织团队会议，让每个人有机会表达意见、分享问题和解决方案。

1. 强调合作和团队精神：

* 强调团队的整体目标和利益，鼓励团队成员相互支持、合作和互助。
* 避免个人主义和竞争心态，促进团队的协作和凝聚力。

1. 指定冲突解决负责人：

* 每位项目经理为冲突解决负责人，负责协调和促进冲突解决过程。
* 这位负责人应公正地处理各方的利益和关注点。

1. 及时解决冲突：

* 鼓励团队成员及时识别和解决冲突，避免冲突积累和扩大化。
* 提供适当的支持和资源，帮助团队成员解决冲突并达成共识。

3.3.3 **团队会议**

本团队的会议形式为线上会议，前期频率一周一次，在最后交付阶段一周3~5次讨论。

会议前准备会议内容概要文档，会议后有专人整理会议纪要。

3.4 **团队管理工作范围**

3.4.1 **团队建设**

* 团队角色和职责：在第一次团队会议中，明确团队成员的角色和职责，确保每个成员了解其在项目中的具体职责和任务。
* 团队沟通：建立良好的团队沟通机制，包括定期组织腾讯会议、建立QQ群和飞书群，促进团队成员之间的有效沟通和协作。
* 团队协作：鼓励团队成员之间的合作和知识共享，建立积极的团队氛围，提高工作效率和质量。

3.4.2 **培训**

* 技术培训：根据团队成员的需求和项目要求，组织相关的技术培训，提升团队成员的技术能力和专业知识。这些技术培训包括MARKII方法的学习、过程管理与控制知识的学习和软件工程经济学学科内容。
* 过程培训：提供项目管理方法和工具的培训，确保团队成员了解项目的工作流程和规范，减少工作中的错误和偏差。在项目初期，我们便制定了一系列项目管理计划，来确保我们的团队的流程规范和工作高效。

3.4.3 **绩效管理**

* 目标设定：与团队成员一起制定明确的个人和团队目标，根据项目目标和要求进行绩效评估。在每次会议上，我们都会为每一名成员设定本周工作目标。
* 绩效评估：定期评估团队成员的绩效，包括但不限于在团队会议中讲述自己的工作内容、工作交付物（如文档、代码）评估等等。在QQ群中，成员会提供反馈和指导，帮助他们改进和发展。

4. **项目相关方管理**

4.1 **相关方识别**

1. 相关方分析：进行相关方分析，识别可能对项目产生影响的各个利益相关方。
2. 本项目的相关方主要为黄杰老师及共同开发实验平台的其他小组

4.2 **相关方参与方式**

1. 会议和沟通：与黄杰老师组织定期会议或沟通，邀请相关方参与讨论项目进展、需求变更等事项。
2. 反馈收集：定期收集相关方的反馈和意见，明确其需求、期望和关注点，以便及时调整项目计划和决策。
3. 信息发布：通过各种途径向相关方发布项目进展、重要决策和变更等信息，确保他们能够及时了解项目情况。我们会在会议上向黄杰老师报告进展，与其他小组成员通过社交软件保持联系。

4.3 **沟通需求**

1. 信息内容：确定相关方对于项目进展、风险、决策等方面的信息需求，确保沟通内容与其关注点一致。每次沟通前，由项目经理记录沟通目标和具体事项，并交由全体项目团队成员确认。
2. 沟通频率：根据相关方的需求和项目进展情况，确定不同相关方之间的沟通频率，避免信息滞后或过度沟通。
3. 沟通方式：了解相关方的偏好和可用沟通渠道，选择适当的沟通方式，例如会议、邮件、电话、在线平台等。

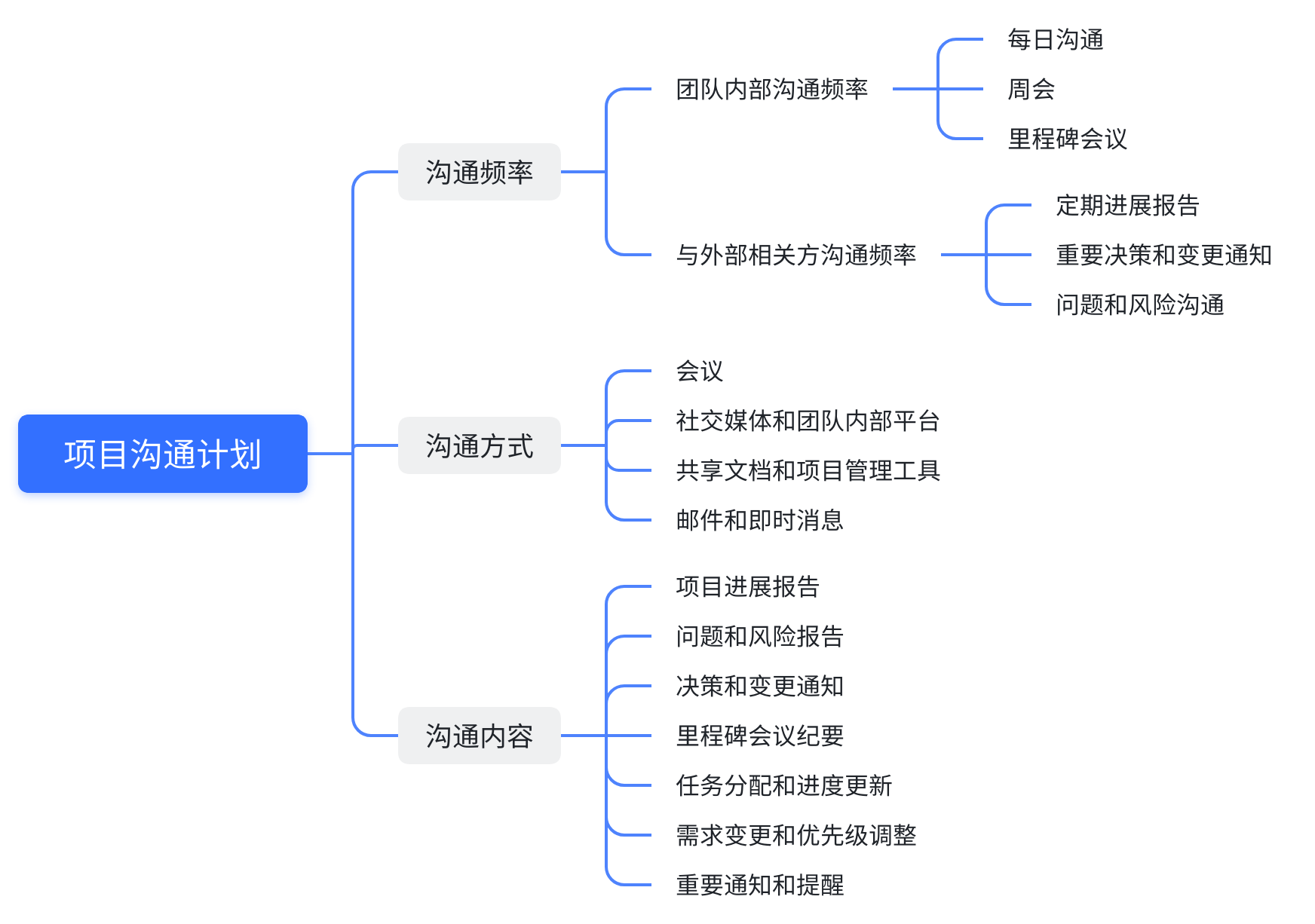
我们主要以线上线下会议和电子邮件的方式与黄杰老师沟通，并向其定期提交正式报告。在QQ群与其他小组沟通，满足不同的需求。

4.4 **利益管理策略**

1. 利益相关方参与：积极邀请相关方参与项目决策和活动，确保他们的利益得到充分考虑和体现。
2. 期望管理：了解相关方的期望和需求，通过沟通和协商，寻求利益相关方之间的共识和平衡。
3. 信息透明度：确保及时、准确地向相关方提供项目信息，增加信息的透明度和可获取性，避免信息不对称和猜测。
4. 冲突管理：及时发现和解决与相关方之间的冲突和分歧，维护项目的合作关系和团队合作氛围。
5. 期望管理：与相关方进行有效的期望管理，明确项目的可行性和限制，并及时沟通变更和调整。

5. **项目沟通管理**

在建立团队、确定项目相关方后，我们制定了项目沟通计划



5.1 **沟通频率**

1. 团队内部沟通频率：

* 每日沟通：团队有专门的QQ群和飞书交流群，团队成员每天进行短暂的日常沟通，分享进展、交流问题和协调工作。
* 周会：每周召开团队会议，回顾上周工作，讨论本周计划，解决团队内部问题，并确保团队成员对项目整体进展有清晰的了解。
* 里程碑会议：在关键的里程碑节点，召开会议进行全员沟通，分享重要进展、评估风险和问题，并制定下一阶段的计划。

1. 与外部相关方沟通频率：

* 定期进展报告：根据约定的时间表，向相关方提供项目的定期进展报告，包括项目宪章、页面布局、项目过程管理情况等信息，以保持透明度和相互了解。
* 重要决策和变更通知：在项目发生重要决策或变更时，及时向相关方发出通知，并提供相应的解释和理由。
* 问题和风险沟通：以会议和邮件的形式定期向相关方报告项目中的问题和风险，并提供相应的解决方案和风险应对策略。
* 我们初步决定与外部相关方召开三次会议，并在项目过程中提交项目宪章并作中期汇报，以保证透明度和相互了解。

5.2 **沟通方式**

1. 会议：包括线上会议和线下会议，用于讨论重要议题、解决问题、进行决策和策划工作。线下会议在黄杰老师的办公室济事楼514，线上会议用腾讯会议软件召开，便于跨地域团队的沟通和合作。每次会议都有专门成员记录会议纪要并事后在内部QQ群分享。
2. 邮件和即时消息：用于书面沟通和快速交流，可以传达信息、提出问题和进行反馈。
3. 共享文档和项目管理工具：使用飞书和github帮助工作，团队成员可以共享文档、追踪任务进度、交流评论等，以便更好地协同工作。
4. 社交媒体和团队内部平台：利用社交媒体和内部平台，可以在团队内部和外部进行信息发布、知识共享和互动交流。我们组建了专门的QQ群。

5.3 **沟通内容**

1. 项目进展报告：包括项目里程碑的完成情况、任务进度、资源使用、预算状况等。
2. 问题和风险报告：识别和描述项目中的问题和风险，并提供解决方案和风险应对策略。
3. 决策和变更通知：向相关方通知重要决策和变更，并提供相关的背景信息和理由。
4. 里程碑会议纪要：记录里程碑会议的讨论要点、决策结果和下一步行动计划。
5. 任务分配和进度更新：沟通团队成员的任务分配情况和任务的进度更新。
6. 需求变更和优先级调整：沟通需求的变更和调整，以及相关的优先级和影响评估。
7. 重要通知和提醒：传达项目中的重要通知、提醒和截止日期，包括每次作业截止日期和答辩日期，确保相关方能及时知悉。

6. **项目时间管理**

6.1 **时间管理计划**

我们最初的时间管理计划如图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 计划 |  | 具体内容 | 时间 |
| 开始日期 |  | 开始开发 | 2023.3 |
| 截止日期 |  | 交付第一版可用软件 | 2023.6 |
| 项目启动和需求分析阶段 (2023.3-2023.4) | 里程碑 | 项目启动会议和团队组建 | 2023.3.12 |
| 完成项目范围和需求分析文档 | 2023.3.19 |
|
|
| 关键交付物 | 项目宪章 | 2023.3.26 |
| 项目范围和需求分析文档 | 2023.3.31 |
| 设计和开发阶段  (2023.4-2023.5) | 里程碑 | 用户界面设计和交互设计 | 2023.4.2 |
| 实验报告填写功能开发完成 | 2023.4.9 |
| 实验指导书浏览和下载功能开发完成 | 2023.4.16 |
|
|
| 关键交付物 | 实验指导书浏览和下载功能实现代码 | 2023.5.28 |
|  | 实验报告填写功能实现代码 | 2023.5.14 |
|  | 用户界面设计文档 | 2023.5.14 |
| 测试和调试阶段  (2023.5-2023.6) | 里程碑 | 前端功能测试和调试 | 2023.5 |
| 用户验收测试和修复 | 2023.5-2023.6 |
| 最终产品演示和交付 | 2023.6 |
| 关键交付物 | 测试报告 | 2023.6 |
| 修复和改进后的前端代码 | 2023.6 |
| 交付和部署阶段 | 里程碑 | 代码上线 | 2023.6.10 |
| 期末答辩 | 2023.6.16 |
| 关键交付物 | 最后代码 | 2023.6.13 |
| 需求与设计文档 | 2023.6.13 |
| 工程管理与过程控制文档 | 2023.6.13 |
| 项目经济评价文档 | 2023.6.13 |
| 答辩PPT | 2023.6.16 |

6.2 **活动计划细化**

在实际工作过程中，我们感到有必要对任务进行拆分，以便大家更有针对性地分工完成不同工作内容。

下面是针对项目时间计划的任务细化。



**点击图片可查看完整电子表格**

7. **项目资源管理**

7.1 **资源识别**

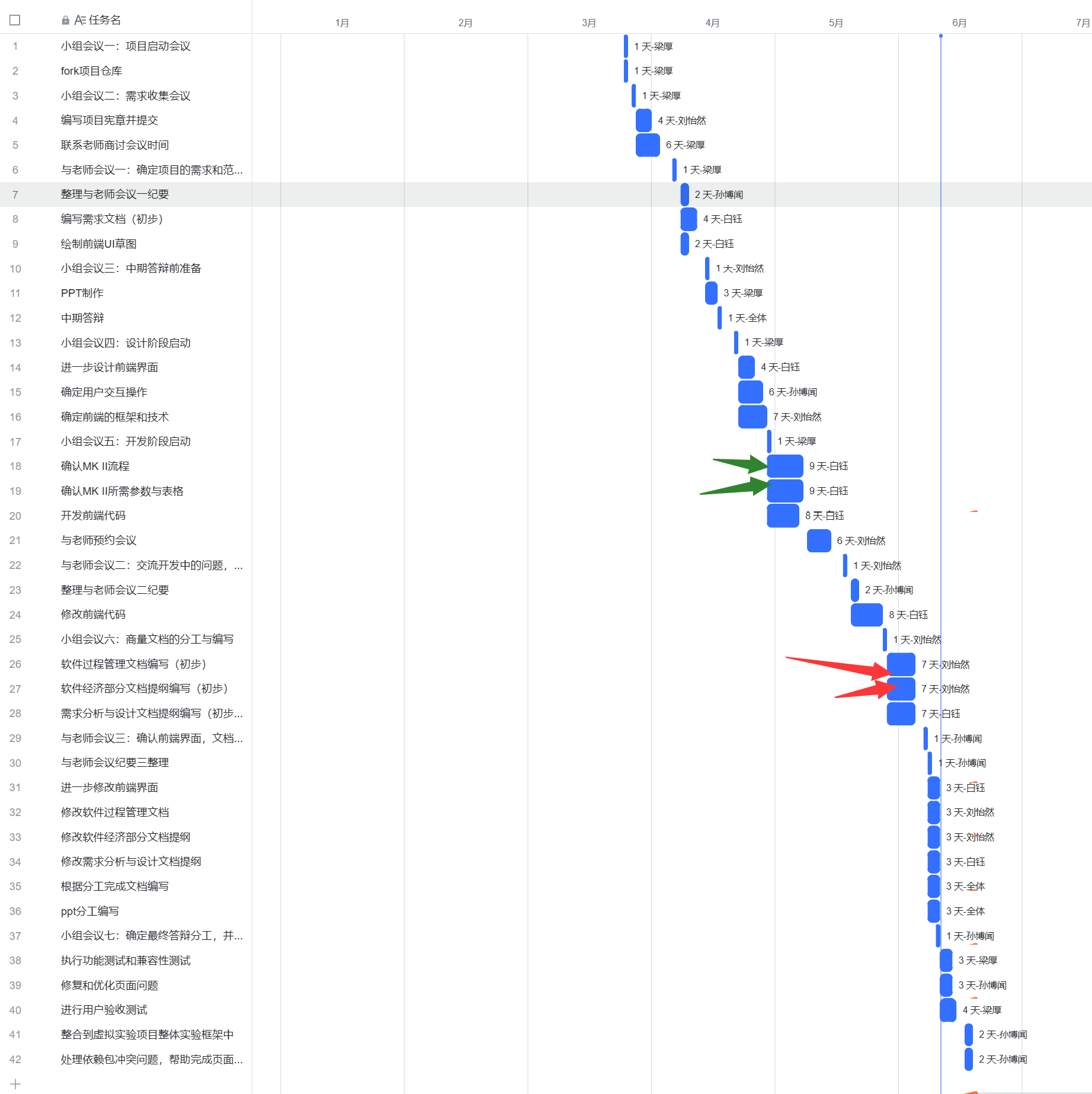
资源识别是指在项目管理中识别和确定所需的各种资源，以支持项目的实施和顺利完成。资源可以包括人力资源、物质资源、设备和工具、技术和知识等。资源识别是项目规划阶段的重要任务之一，旨在确定项目所需资源的类型、数量、质量和时间等方面的要求。

MARK II项目资源需求清单如下：

|  |  |
| --- | --- |
| **资源类别** | **资源详述** |
| 劳动力资源需求 | 前端开发人员：4人 |
|  | UI/UX设计师：1人（可选，根据设计需求决定） |
|  | 项目经理：1人（负责项目管理和协调，由开发团队成员轮流担任） |
| 设备和软件需求 | 个人电脑或笔记本电脑：每位团队成员需要一台可靠的计算机设备。 |
|  | 开发工具和集成开发环境（IDE）：例如Visual Studio Code、Sublime Text等。 |
|  | 版本控制软件：例如Git，用于代码管理和协作。 |
|  | 浏览器：用于测试和调试前端页面的兼容性。 |
|  | 图形设计工具：例如Adobe Photoshop、Sketch等（如果需要进行UI/UX设计）。 |
| 会议和协作需求 | 会议室或虚拟会议工具：用于团队会议、进度更新和问题讨论。可选选项有：   * 同济大学嘉定校区图书馆 * 教学楼 * 济事楼 * 线上会议室 |
|  | 团队协作工具：例如Slack、Microsoft Teams、Trello、飞书、QQ等，用于团队沟通和协作。 |
| 场地需求 | 办公空间：为团队成员提供舒适的办公环境，包括工作桌、椅子和其他办公设施。可选选项：   * 寝室 * 同济大学嘉定校区图书馆 * 自习教室 * 济事楼 |
| 时间和进度要求 | 确定每位团队成员的可用工作时间和工作日程。 |
|  | 制定项目进度计划，包括里程碑和任务分配。 |
| 知识与技术要求 | 实验教学平台相关开源项⽬、与本项目框架与技术选型相似的开源项目（github项目等） |
| 官方文档（软件规模估算标准、vue3.0官方文档、js官方文档等） |
| 相关书籍（电子版图书、图书馆借阅），主要与软件项目的工程管理与过程控制相关 |
| ⽹络资源（其他网络资源，包括开发过程中的具体问题解决方案等） |

7.2 **资源分配与调度**

由上表的资源需求，可以得到如下分配策略：



上图中红色箭头处于绿色箭头处表示两处资源冲突，在任务细化之后，考虑到以上两处包含的并行任务较为重要且花费时间，分配相同的人员资源并不合理，所以进行人员资源的调整，调整之后如下：

调整后资源分配如下：



8. **项目监控与控制**

本项目旨在开发一个用于虚拟仿真实验软件规模度量MARK II方法前端实验报告生成页面，用于给用户提供一个关于MARK II软件度量方法的实验指导和实验报告在线生成的平台。

敏捷开发方法（Agile Model）中的Kanban开发方法的核心思想是通过可视化、限制工作项和持续改进来提高团队的工作效率和项目交付能力。它强调团队协作、任务流程的可见性和适应性，使团队能够更好地应对需求变化和优化工作流程。

我们小组采用了**Kanban** 来管理项目进度。项目初期，我们将其划分为大致几个重要阶段：需求分析阶段、设计阶段、开发阶段、测试阶段、部署和交付阶段。在项目开展过程中，根据所处的阶段制订相应的任务，同时分配对应的人员完成该任务。基于对需求变更、人员时间分配问题的一些现实考量，我们并不采用线性迭代的瀑布模型，而是选用敏捷开发的方法，所以在特定的某一阶段也可能覆盖上一阶段的内容（例如，在开发阶段，出现需求变更，我们会重复开展需求分析阶段的一些任务）。

8.1 **项目监控与控制计划**

8.1.1 **监控项目进展**

* 进度监控：定期跟踪项目的进度，比较实际进展与计划进度，识别任何延误或偏差，并采取相应的措施进行调整。
* 成本监控：监控项目的成本支出，与预算进行对比，识别任何超支或节约，并采取相应的控制措施。
* 质量监控：进行质量检查和评估，确保项目交付物符合质量标准和要求，及时纠正和预防质量问题。
* 风险监控：识别和评估项目的风险，跟踪风险的发展和影响，制定风险应对措施，确保项目的稳定和可控性。

8.1.2 **评估项目绩效**

* 里程碑评估：根据项目里程碑的完成情况，评估项目的整体进展和达成情况，确定项目是否按计划推进。
* 成果物评估：对项目交付物进行评估，与预期的质量标准进行对比，确认交付物的完整性、准确性和满足度。
* 绩效指标评估：根据预先定义的绩效指标，对团队成员和项目进展进行评估，以衡量团队和项目的表现和效果。

8.1.3 **采取纠正措施**

* 问题识别和分析：识别项目中出现的问题和偏差，分析其根本原因，以确定需要采取的纠正措施。
* 纠正措施制定：制定具体的纠正措施和行动计划，包括调整进度计划、资源分配、风险应对等，以解决项目中的问题和偏差。
* 实施和监控：执行纠正措施并监控其效果，及时调整和调整计划，确保问题得到解决并项目重新回到正确的轨道上。

8.2 **项目监控与控制计划细化**

**项目阶段和里程碑 （详细任务列表见MARK II Kanban）**

1. 需求分析阶段

* 定义项目需求和功能要求
* 确定用户界面设计和交互细节
* 确定页面所需的内容和资源
* 里程碑：
* 项目启动和团队组建
* 和用户（教师）开会确定项目需求
* 生成第一次会议纪要&基本的需求文档
* 关键交付物
* 项目宪章
* 第一次会议纪要&基本的需求文档
* 对应任务（No.）：01，02，03，04，05，06，07，08

1. 设计阶段

* 创建页面原型和草图
* 设计用户界面和交互流程
* 选择适当的前端框架和技术
* 里程碑
* 用户界面和交互设计
* 关键交付物
* 用户交互界面UI图
* 对应任务（No.）：09，10，13，14，15

1. 开发阶段

* 设置开发环境和工具
* 实现页面布局和样式
* 开发页面交互功能
* 里程碑：
* 实验报告填写功能开发完成
* 关键交付物
* 实验报告填写功能代码
* 对应任务（No.）：16，17，18，19，20，21，24，25，26，27，28，30，31，32，34，35，36

1. 测试阶段

* 执行功能测试和兼容性测试
* 修复和优化页面问题
* 进行用户验收测试
* 里程碑：
* 前端功能测试和调试
* 用户验收测试和修复
* 关键交付物
* 测试报告
* 对应任务（No.）：22，23，29，37，38，39，42

1. 部署和交付阶段

* 整合到虚拟实验项目整体实验框架中
* 处理依赖包冲突问题，帮助完成页面上线
* 里程碑：最终产品演示和交付
* 对应任务（No.）：12，40，41

8.3 **开销管理计划**

8.3.1 **成本分类**

1. **员工成本**：包括员工的工资、奖金等。员工的工资按照时薪计费，要求不低于上海市最低时薪标准，具体依照岗位和工作内容变化。在周末和法定节假日应该支付三倍时薪。
2. **系统成本**：系统的开发需要使用设备成本，包括计算机、键盘鼠标等外设以及系统部署的服务器使用、网络费用、域名费用等。
3. **日常支出：**主要是组织管理承担的费用，例如场地租金、办公耗材等。

具体的成本计算会在“成本分析”部分给出。

8.3.2 **成本控制**

软件开发项目中的成本控制是确保项目在预算范围内进行并有效管理项目成本的过程。我们小组拟采用的成本控制方法有：

1. 制定预算：  
   在项目启动阶段，制定详细的项目预算，包括开发成本、人员费用、硬件和软件成本、测试费用、培训费用、项目管理费用等。预算应该基于项目需求和可行性研究，同时也要考虑到风险和不确定性。
2. 成本估算和跟踪：  
   在项目计划阶段，进行详细的成本估算，并制定成本控制的基准。跟踪实际成本与预算之间的差异，及时发现和解决成本超支的问题。使用成本跟踪工具和技术，例如成本绩效指数（CPI）和成本差异（CV）等来评估项目的成本绩效。
3. 管理资源：  
   确保合理分配和管理项目所需的资源，包括人员、设备、工具和软件许可等。优化资源利用率，避免过度或不足的资源配置，以减少成本浪费和效率低下的情况。
4. 风险管理：  
   识别和评估项目风险，特别是与成本相关的风险。制定相应的应对策略，如风险转移、降低风险或保险等，以减少风险对项目成本的影响。
5. 控制范围：  
   严格管理项目范围，并防止范围蔓延和变更引起的成本增加。确保变更请求经过评估和审批，并明确变更对成本的影响，避免未经授权的变更引起的成本溢出。
6. 项目沟通：  
   保持与项目相关各方的良好沟通，包括团队成员、利益相关者和管理层。及时共享项目成本情况和预测，讨论成本控制策略，并与利益相关者共同解决成本问题。项目沟通在本项目中体现为与老师线上/线下进行的几次会议。
7. 经验教训的应用：  
   在项目执行过程中，不断总结和应用经验教训，寻找成本控制的最佳实践和方法。通过知识管理和经验积累，提高未来项目的成本控制能力。

通过有效的成本控制，软件开发项目可以更好地管理和优化成本，确保项目在可接受的预算范围内交付，并提高项目的成功率和客户满意度。

8.4 **进度管理**

进度管理可视化Kanban在线文档地址：[✅VSE-MarkII项目任务管理](https://ulqqdencc9.feishu.cn/base/VAn3byxqwaCe5QsoAnZc9LQXn6a?table=tblU6UIhc2yleLra&view=vewXxBNTOK)

8.4.1 **kanban法工作流介绍**

**Kanban工作流**

1. 创建Kanban面板：

* 使用在线Kanban工具（如Trello、Asana、Jira等）创建一个新项目面板。
* 列出以下基本列：待办、进行中、已完成。
* 每个任务将在面板上作为卡片表示。

1. 初始任务规划：

* 将项目中的主要任务（如需求分析、设计、开发、测试、部署）添加到待办列中。
* 将每个任务拆分为具体的子任务，以更好地管理进度。

1. 分配任务：

* 在项目团队中讨论并分配每个任务给适当的成员。
* 将每个分配给成员的任务移动到进行中列。

1. 设定优先级：

* 根据任务的重要性和紧急性，确定任务的优先级。
* 在卡片上使用标签或其他指示符表示任务的优先级。

1. 迭代规划：

* 将较大的任务进一步细分为小的迭代或里程碑。
* 将每个迭代的子任务添加到待办列中。

1. 迭代进行：

* 团队成员根据优先级和可用时间选择并处理待办列中的任务。
* 将选定的任务移动到进行中列，并在开始工作时更新卡片的状态。

1. 工作跟踪：

* 每个团队成员根据任务进度更新卡片的状态和进展情况。
* 使用评论、附件等功能进行卡片的实时讨论和信息共享。

1. 进度更新：

* 团队成员应定期更新任务状态，以反映实际进展。
* 尽量遵循每日例会，以了解每个成员的工作情况和任何问题。

1. 迭代评审：

* 在每个迭代结束时，进行团队会议，回顾已完成的任务并讨论下一步的计划。
* 移动已完成的任务到已完成列，并记录下任何有用的反馈或教训。

1. 项目交付：

* 当所有任务完成并通过测试后，将任务移动到已完成列。
* 进行最终的验收测试，并准备项目交付和部署。

8.4.2 **任务执行表**



8.4.3 **任务统计看板**







8.4.4 **挣值分析**

挣值分析（Earned Value Analysis）是一种项目管理工具，用于评估项目的**实际进度和成本**与**计划进度和成本**之间的差异，并提供关于项目绩效的可视化和定量的指标。由于该项目为学校课程的课程作业，并不涉及具体的金额，所以我们用人力成本单位（man/months）代替实际的金额作为分析的指标。

计划进度和成本：

* 预计项目开始日期：2023年3月25日（以提交项目宪章 Project Charter为准）
* 预计项目结束日期：2023年6月12日（以最终项目其中答辩为准）
* 预算成本：373.88ph（根据工作量估算得出的数据，取中间值）
* 说明1：一般的挣值分析是以具体的金额为单位进行控制，但由于我们的项目的特殊性，这里用人月代替具体的金额进行分析和控制；
* 说明2：在工作量估算中，我们把人时转换成人月，是基于每月21.75个工作日和每天8小时的工作时间进行的折算。但是在我们的项目开发中，由于实际情况限制，我们每月在该项目上的投入远远低于21.75个工作日和每天8小时。所以，在此处的挣值分析中，假定每月一人的工作时长为10个工作日每天3小时。
* 根据假定，预算成本折算成人月为：人月，取近似值为12。

挣值分析：

1. 定义指标：

* PV（计划价值）：项目在特定时间点上的计划成本。
* EV（挣值）：在特定时间点上已完成的实际工作的预期成本。
* AC（实际成本）：在特定时间点上已实际支出的成本。
* SV（进度偏差）：计划价值与挣值之间的差异。
* CV（成本偏差）：挣值与实际成本之间的差异。

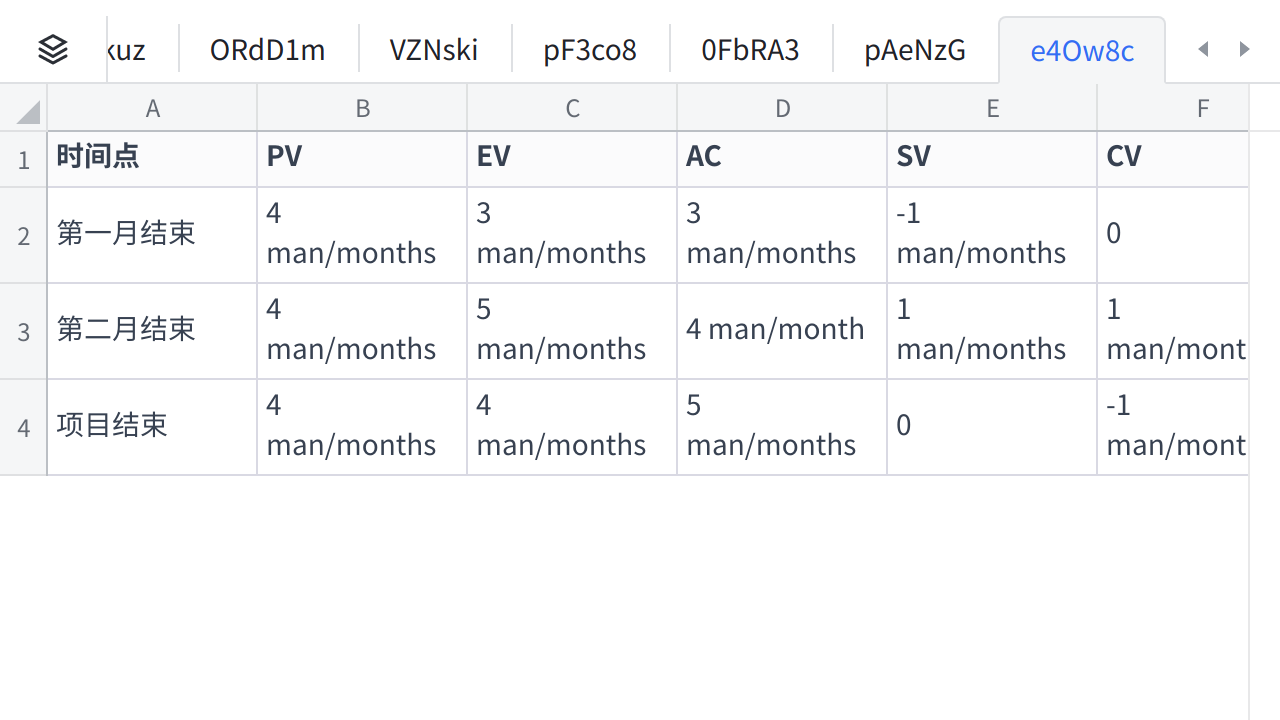
1. 确定测量时间点：

* 第一月结束（2023年4月25日）
* 第二月结束（2023年5月25日）
* 项目结束（2023年6月25日）

1. 计算指标：

* PV：根据每周工作量和工时估计计算每个测量时间点的计划价值。
* EV：根据每个测量时间点团队实际完成的工作量和相应的预算成本计算挣值。
* AC：每个测量时间点的实际成本。
* SV：EV - PV，评估项目进度。
* CV：EV - AC，评估项目成本。

1. 净值分析表：



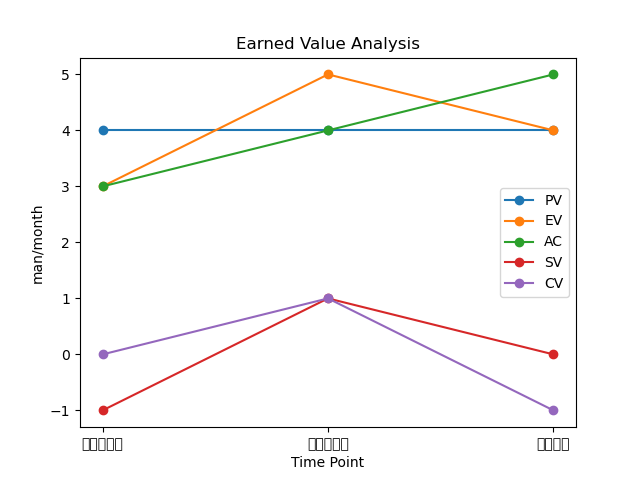
**点击图片可查看完整电子表格**

解释说明：

* 在第一月结束时，团队完成了较少的工作，导致进度偏差（SV）为负值。
* 在第二月结束时，团队加快了开发速度，进度偏差和成本偏差有很大改善。
* 在项目结束时，团队完成了预期的工作量，但仍然有一定的进度偏差和成本偏差。

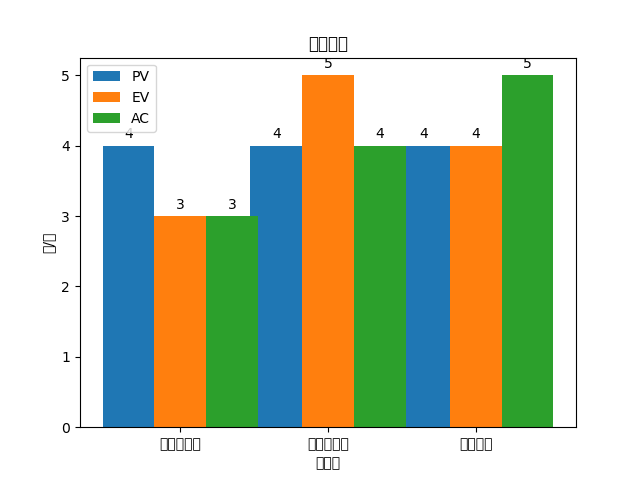
结论：  
根据净值分析结果，我们可以看到项目的进度和成本偏差情况。团队在第一周遇到了一些挑战，但在后续的周数中逐渐改善。然而，项目结束时仍然存在一定的进度偏差和成本偏差。为了控制成本和提高绩效，团队应进一步优化工作流程，加强沟通和协作，及时应对问题和风险。

以上结果可以通过图表的方式呈现：



净值分析折线图

可以看出，项目总体进度落后于规划，主要的落后部分出现在项目前期，前期由于项目刚刚开展，启动比较困难。在适应了相应的情况以及完成一些基本任务之后，可以看到5月份项目组开始采取了一些应对措施，并开始逐渐追赶进度。项目的成本偏差和进度偏差也在追赶之下日渐接近预期。



净值分析柱状图

由净值分析可以得出，项目的进度慢于预期。其原因可以归结如下：

* 前期需求不明确，项目启动动力不足。
* 项目开展过程中由于前期需求未完全明确，出现需求变更的情况，需要更多的工作量进行迭代。

对于上述情况，项目组采用鱼骨图进行分析，并在追赶期进行了补救，详情在“问题处理”和“过程优化”部分介绍。

8.5 **问题处理**

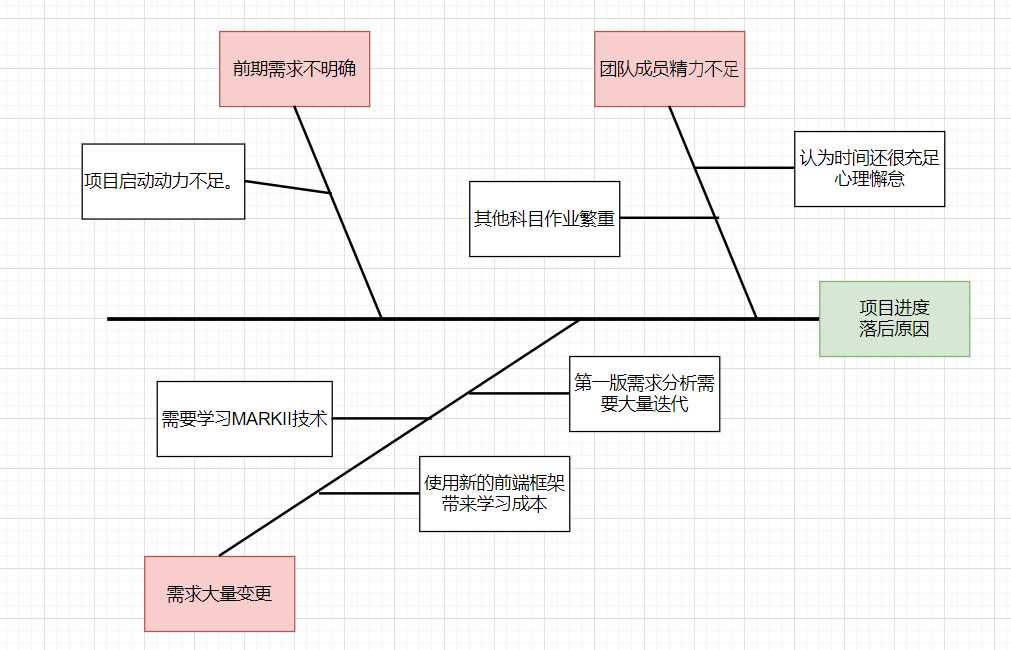
针对项⽬开展中出现的问题，本项⽬组采⽤⻥⻣图进⾏原因分析与问题溯源，此处截取前期时绘制的⻥⻣图进⾏相应的说明，针对项⽬前期项⽬进度落后于预期这⼀问题，本项⽬组发现问题原因如下所⽰：

* 团队成员精⼒不⾜
* 前期需求不明确，项目启动动力不足。
* 项目开展过程中由于前期需求未完全明确，出现需求变更的情况，需要更多的工作量进行迭代。

对于上述原因，本项⽬组采⽤以下措施及时补救：

* 加班：我们需要更多工作量进行迭代，因此需要增加每天⽤于项⽬开发的时间；
* 增加线上会议沟通：尽快明确需求和分工
* 增加QQ群内沟通，分享技术，交流经验
* 积极寻求项⽬组外⼒量的帮助：针对增加的新需求与前期尚未预料的⻛险，本项⽬成员积极向⾝边有过相关经历与经验的⽼师和同学求助，在他们的帮助下快速把握问题关键，于较短时间内解决相应的问题。

通过上述措施，本项⽬团队得以在项⽬中后期不断缩⼩进度与计划之间的差距，最终如期完成项⽬。



项目前中期鱼骨图分析

8.6 **过程优化**

8.6.1 **常规过程管理**

常规过程管理是指在软件开发项目中，对项目过程进行有效管理和控制，以确保项目按时、高质量地交付。

1. 过程执行和监控：

* 在每个阶段执行项目管理计划，进行需求分析、设计、开发、测试和交付等活动。
* 监控项目进度、资源使用和质量，及时识别和解决问题。
* 进行定期的状态报告和项目评审，确保项目按计划进行。
* 项⽬计划由项⽬负责⼈制定，组内4位成员轮流担任负责人。由项⽬各⼩组组⻓、项⽬成员⼀起进⾏评审。



8.6.2 **常规过程优化**

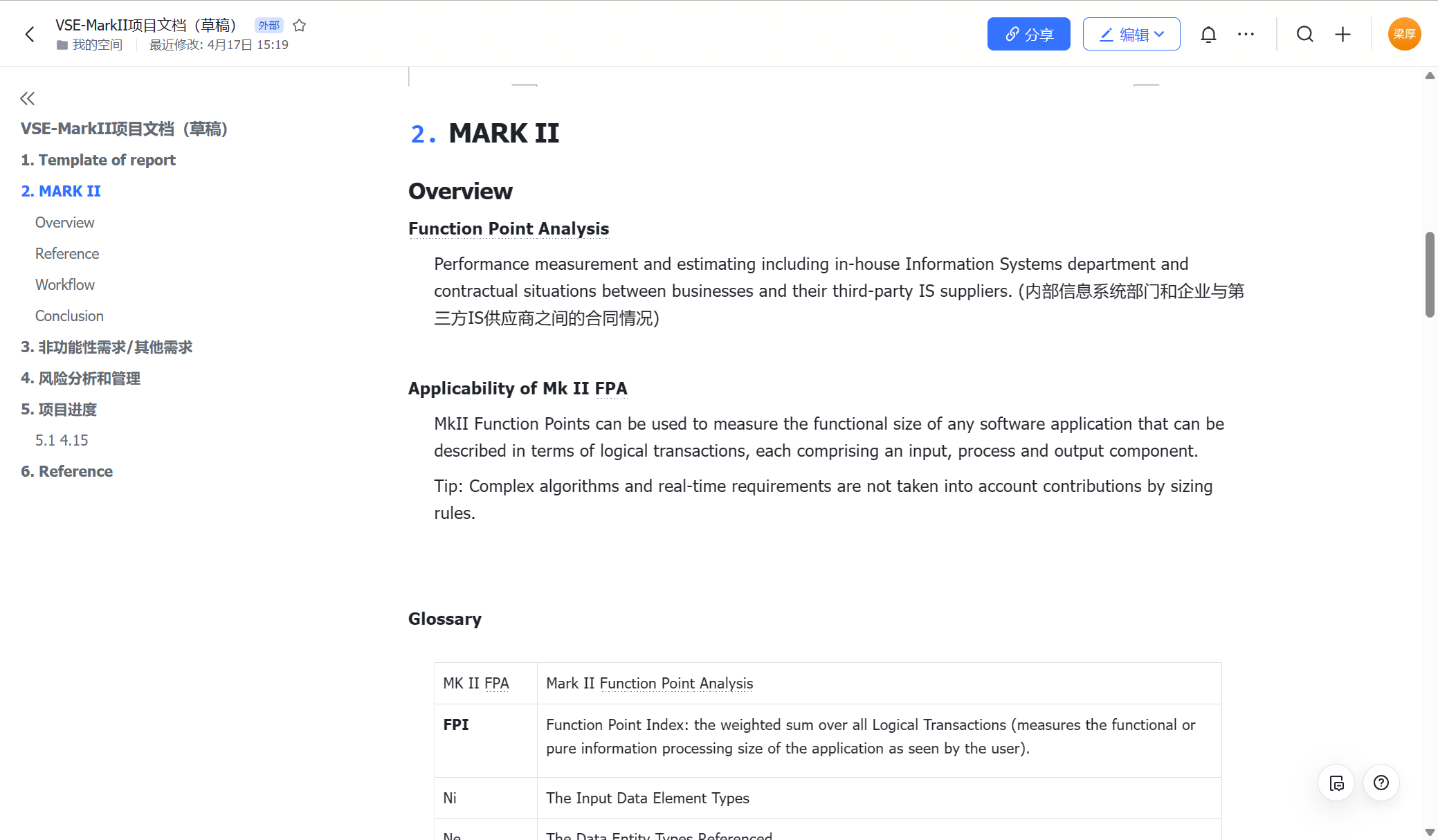
1. 过程改进：

* 定期评估项目过程的效果和改进机会。
* 收集反馈和经验教训，针对性地调整和改进项目过程。
* 推动团队成员的能力提升和知识分享，促进过程的持续改进。

1. 过程文档和知识管理：

* 建立项目文档和知识库，包括项目计划、需求文档、设计文档、测试报告等。
* 确保文档和知识的有效管理和共享，便于团队成员的参考和使用。

[VSE-MarkII项目文档（草稿）](https://ulqqdencc9.feishu.cn/docx/Dxbpd98Z7opQSHxzj2TcGDgdnKb)



1. 沟通和协作：

* 保持团队成员之间的良好沟通和协作，及时交流项目进展、问题和风险。
* 确保项目相关方（教师）的参与和支持，及时解决利益冲突和问题。

3次与教师的讨论会（2023.04.05/2023.06.02/2023.06.07）

9. **项目质量管理**

9.1 **质量管理计划**

9.1.1 **质量目标**

本项目的质量目标是确保开发的虚拟仿真实验平台能够稳定、可靠地运行，并提供优质的用户体验和教学效果。具体质量目标包括：

1. 保证前端页面功能的正确性和完整性。
2. 提供良好的用户体验和界面设计。
3. 遵循最佳的编码实践和代码规范。

9.1.2 **质量标准和指标：**

1. 页面功能测试通过率：目标达到100%。
2. 用户界面设计评分：目标达到8分（满分10分）。
3. 代码规范符合程度：目标达到90%以上。
4. 代码覆盖率：目标达到80%以上。
5. 响应时间：页面加载时间不超过2秒。

9.1.3 **质量管理原则**

1. 真实性:质量保证和检测工作的结果必须是真实的。真实、准确、完整地记录，包括质量检查中发现的问题和测试中发现的软件缺陷。
2. 客观性:质量保证和测试是客观的，基于质量保证规范和标准、需求文档和测试用例。质量保证以质量规范为基础，不受项目经理意志的转移。
3. 独立性:质量保证和测试人员独立从事质量工作，不受项目管理人员的制约

9.1.4 **质量管理活动**

质量管理应在项目的关键阶段进行。项目的重要里程碑是:设计完成、代码完成、集成测试通过、用户验收测试通过和在线完成。主要包括以下活动:

1. 评审工作输出，即输出是否符合质量要求。
2. 需求文档是否满足规格要求。
3. 设计文件是否完整、清晰。
4. 代码编写是否符合代码规范。
5. 过程符合性评审，即检查项目管理和工程技术过程是否符合过程规范。
6. 需求文件是否经过需求部门的正式确认。
7. 设计是否经过技术评审和评审，记录被保留。
8. 代码是否经过技术评审和评审，记录被保留。
9. 收集数据并执行度量分析，主要是系统的质量属性数据，包括:需求覆盖率，缺陷统计分析，以及测试用例执行结果。
10. 实施和改进质量保证体系，总结最佳工程实践，改进软件开发过程。

9.2 **测试计划**

9.2.1 **软件测试周期**

软件测试包括以下主要阶段:

1. 冒烟测试:测试系统的基本功能，包括进入系统各模块及主要功能路径。
2. 功能测试:对系统各模块的功能进行详细的测试。它应该包括正向测试、反向测试和边界值测试。
3. 系统集成测试:有两种类型的集成测试，首先应该包括系统内模块之间的集成测试，还应该包括系统与系统其余部分之间的集成测试。
4. 用户验收测试:根据用户的业务操作场景。从最终用户的角度进行测试

9.2.2 **软件测试操作规程**

1. 在项目开始时，QA需要制定一个测试计划。它可以是项目计划的一部分，也可以作为项目计划的一部分
2. 独立的计划。
3. 应该审查设计好的测试用例，并且通常是与开发团队一起审查。理想情况下，还可以与最终用户一起审查它。
4. 在测试中发现的软件缺陷应该在系统中注册，并遵循缺陷生命周期的需求。

9.2.3 **软件缺陷的级别**

根据软件缺陷对系统的影响程度，可以将其严重程度分为以下几个级别:

1. 严重:系统崩溃、数据丢失、内存泄漏、安全问题或核心特性不可用。
2. 主要:导致重大业务操作错误、结果不正确或缺少重要功能。
3. 一般:导致一般功能问题或不正确的结果。
4. 次要:次要的功能问题，或替代解决方案。
5. 轻微:轻微的功能缺陷或边界，例如UI元素的轻微缺陷。

9.2.4 **系统实施的质量要求**

1. 该系统应由用户测试。
2. 系统不应有严重、重大或一般级别的缺陷。

9.3 **质量管理实现**

9.3.1 **代码审查**

9.3.1.1 **代码审查要求**

我们现在对代码的审查则是作为开发流程的一个必选项，每次开发新功能或者修复Bug，开一个新的分支，分支要合并到master有两个必要条件：

* 所有的自动化测试通过（使用Selenium）
* 有至少一个人Code Review通过

9.3.1.2 **代码审查规范**

在做Code Review时，需要针对审查出有问题的代码行添加评论，如果只是评论，有时候对于被审查者比较难甄别评论所代表的含义，是不是必须要修改。

对Review的评论进行分级，不同级别的结果可以打上不同的Tag：

* [blocker]: 在评论前面加上一个[blocker]标记，表示这个代码行的问题必须要修改
* [optional]：在评论前面加上一个[optional]标记，表示这个代码行的问题可改可不改
* [question]：在评论前面加上一个[question]标记，表示对这个代码行不理解，有问题需要问，被审查者需要针对问题进行回复澄清

类似这样的分级可以帮助被审查者直观了解Review结果，提高Review效率。

9.3.1.3 **代码审查记录**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 任务序号 | 任务内容 | 开发者 | 审查员 | 单元自动化测试 | 审查意见 | 审查结果 | 代码状态 |
| 14 | 开发前端代码 | 白钰 | 梁厚 | 已通过 | [optional] 建议将实验步骤中的输入框整合入一个表单 | 通过 | 已提交 |
| 24 | 修改前端代码 | 刘怡然 | 梁厚 | 已通过 | [question] test: '21111'是什么意思？ | 通过 | 已提交 |
| 31 | 进一步修改前端界面 | 白钰 | 孙博闻 | 已通过 | [blocker]成本计算逻辑出错 | 修改后通过 | 已提交 |
| 39 | 修复和优化页面问题 | 刘怡然 | 孙博闻 | 已通过 | [optional]页面布局用了太多空格，建议修改得更简洁一些 | 通过 | 已提交 |

9.3.2 **UI/UX评估**

邀请用户对界面进行评估（邀请教师参加讨论会提出改进意见），收集反馈并进行相应的改进。

我们的UI界面质量管理过程如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 迭代次数 | 评估意见 | 评估日期 | UI改进任务 | 完成日期 | 改进评价 |
| 第一次会议 | 1. 实验原理修改为实验步骤 2. 实验情况所需只有ADDED部分，其他默认为0 3. 需要添加Calculate Productivity | 4.06 | 标题修改  表格默认值改为0  添加计算工作量的部分 | 4.8 | 通过 |
| 中期答辩评估 | 1. 需要展示出MARKII算法的计算过程 2. 添加中间计算结果的展示 3. 输入参数以表格的形式展示 | 4.17 | 1. 实验步骤细化 2. 中间结果展示 3. 将参数填空改为表格 | 4.20 | 通过 |
| 第二次会议 | 1. 完善后续软件成本实验步骤：功能点所需人时 2. 不需要展示全部功能点的结果 | 6.2 | 添加人时  功能点结果展示表格修改 | 6.3 | 通过 |
| 第三次会议 | 1. 第四步拆分成3步，每步出一个中间结果 2. 调整字号，小标题字号>表格标题 | 6.7 | 小标题修改  字号修改 | 6.8 | 通过 |

9.3.3 **软件测试**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务序号 | 测试等级 | 任务内容 | 测试用例审查 | 测试结果 | 是否达到要求 |
| 14 | 单元测试 | 前端v1.0自动化测试 | 已通过 | 通过 | 是 |
| 24 | 单元测试 | 前端v2.0自动化测试 | 已通过 | 通过 | 是 |
| 31 | 单元测试 | 前端v3.0自动化测试 | 已通过 | 通过 | 是 |
| 39 | 单元测试 | 前端v4.0自动化测试 | 已通过 | 通过 | 是 |
| 38-1 | 集成测试 | 功能测试 | 已通过 | 通过 | 是 |
| 38-2 | 系统测试 | 系统测试 | 已通过 | 通过 | 是 |
| 40 | 验收测试 | 验收测试 | 已通过 | 通过 | 是 |



10. **项目变更管理**

10.1 **变更管理计划**

在指定项目计划时，我们决定采用以下的流程和决策机制管理变更

10.1.1 **变更管理流程**

1. 变更识别：识别和记录项目中的变更请求，包括提出变更的原因、影响范围和期望结果。
2. 变更评估：对变更请求进行评估，分析其对项目范围、进度、成本、质量和风险等方面的影响。
3. 变更决策：基于变更评估结果，进行变更决策，确定是否接受、拒绝或推迟变更，并对变更进行优先级排序。
4. 变更实施：对接受的变更进行实施，包括更新项目计划、调整资源分配、修改项目文件等。
5. 变更控制：监控和控制已实施的变更，确保其对项目目标的实现产生预期的影响，并追踪变更引起的新问题和风险。
6. 变更记录：记录所有变更请求、评估结果、决策和实施情况，以便追溯和审计。

10.1.2 **决策机制**

1. 变更委员会：设立由项目相关方组成的变更委员会，负责评估和决策变更请求。委员会成员包括团队全体成员。
2. 变更控制板：设立变更控制板，负责监督变更管理流程的执行，确保变更决策的一致性和合规性。我们的每一次变更都会体现在任务管理看板上，方便及时追踪
3. 变更影响评估：对于重大变更，可以进行详细的变更影响评估，涉及多个方面的影响，并由决策机构进行综合评估和决策，必要时与外部相关方（黄杰老师）沟通交流。

10.2 **变更管理计划细化**

10.2.1 **变更流程**

1. 编写变更请求表（Change Request Form）
2. 提出变更请求：任何团队成员可以提出变更请求，包括功能增加、修改或删除、界面调整等。在本项目中变更不仅来自开发成员的意见，还来源于与客户（教师）交流的几次线下讨论会。
3. 变更评估：项目经理和团队成员评估变更请求的可行性、影响范围、工作量、成本和时间等因素。
4. 变更讨论和决策：团队成员和项目经理进行变更讨论，并做出最终的变更决策（开发团队组内会议）。
5. 变更实施：如果变更被批准，团队将根据决策结果进行变更实施（由变更生成的任务将被加入Kanban）。
6. 成员完成变更任务
7. 变更验证和测试：团队进行变更后的验证和测试，确保变更符合预期并不会引入新的问题。
8. 变更文档记录：所有变更请求、评估、讨论和决策、实施以及验证和测试结果都应记录在变更文档中，以便日后查阅和追溯（本项目中，变更文档记录主要来源于与用户开展几次讨论会的会议纪要）。

10.2.2 **变更管理角色**

|  |  |
| --- | --- |
| 项目经理 | 负责变更管理流程的监督和决策，确保变更管理的有效执行（在本项目中由轮值组长担任） |
| 团队成员 | 负责提出变更请求、评估变更请求、参与变更讨论和实施变更。 |
| 项目所有者/客户代表 | 参与变更评估和决策，提供对项目目标和需求的意见和反馈（教师）。 |

10.2.3 **变更管理注意事项**

1. 提前规划：在项目计划中考虑变更管理的时间和资源，并明确变更管理的责任和流程。
2. 及时响应：对于变更请求要及时响应，并在合理的时间范围内进行评估和决策。
3. 变更影响分析：评估变更对项目目标、进度、资源和成本的影响，确保变更对项目整体有利。
4. 沟通和协作：团队成员之间要密切合作，确保变更请求的评估和讨论充分透明和充分讨论。
5. 文档记录：所有变更相关的信息都应记录在变更管理文档中，以便追溯和参考。

10.3 **变更实例1**

10.3.1 **变更请求表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **变更请求表** |  |  |  |  |  |
| 项目名称：虚拟仿真实验平台-基于MARKII的软件规模度量实验 |  |  |  |  |  |
| 请求提交时间：6.2 |  |  |  |  |  |
| 变更请求名称：实验报告布局变更 |  |  |  |  |  |
| 变更序号：03 |  |  |  |  |  |
| 提交人（姓名+联系方式）：白钰（1845201056@qq.com ） |  |  |  |  |  |
| 变更种类 | * 范围 | * 计划 | * 花销 | * ~~技术~~ | * 其他 |
| 变更请求描述：前端页面整体修改，实验步骤重新细化 |  |  |  |  |  |
| 引起变更请求的事件：与项目负责人的会议中，黄杰老师对项目提出了修改意见 |  |  |  |  |  |
| 变更的理由/为什么需要/希望继续/完成项目:  必要性：  项目负责人指出我们目前的前端页面不够清晰，其中：  功能点计算部分使用下划线填空，建议改成表格，更加整齐  第四步包含3个中间结果生成，应该分成3步  最后加一步，计算直接人力成本  可行性：  该变更工作量不大，一名前端开发人员短期即可完成，不需要额外投入成本 |  |  |  |  |  |
| 该变化带来的影响：   * 范围：实验前端页面 * 时间安排: * 花费：无 * 人员: 前端开发人员（白钰） 、测试人员（梁厚） * 风险：无 * 其他：无 |  |  |  |  |  |
| 如果变更通过，应该执行的任务:   * 修改前端页面 * 重新进行自动化单元测试 * 代码审查 * 代码提交 |  |  |  |  |  |
| 所需审批:孙博闻 |  |  |  |  |  |
| 姓名/职位 |  |  | 日期 | 通过/拒绝 |  |
| 2053171白钰 | 前端开发工程师 |  | 6.2 | 通过 |  |
| 2053412刘怡然 | 前端开发工程师 |  | 6.2 | 通过 |  |
| 2051840梁厚 | 测试工程师 |  | 6.2 | 通过 |  |
| 2053512孙博闻 | 项目经理 |  | 6.2 | 通过 |  |

10.3.2 **会议纪要**



10.3.3 **变更实施**

原任务：



该任务已经完成，小组评审后认为代码已经不再需要修改。变更通过后新建任务



10.4 **变更实例2**

在项目的部署与交付阶段，团队成员刘怡然发起PPT分工计划变更申请，目的使分工更为合理，提高工作效率。

10.4.1 **变更请求表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **变更请求表** |  |  |  |  |  |
| 项目名称：虚拟仿真实验平台-基于MARKII的软件规模度量实验 |  |  |  |  |  |
| 请求提交时间：2023/6/10 |  |  |  |  |  |
| 变更请求名称：答辩PPT计划修改 |  |  |  |  |  |
| 变更序号：04 |  |  |  |  |  |
| 提交人（姓名+联系方式）：刘怡然（907021907@qq.com ） |  |  |  |  |  |
| 变更种类 | * 范围 | * ~~计划~~ | * 花销 | * 技术 | * 其他 |
| 变更请求描述：  将答辩PPT分工修改如下：  需求（白钰）  过程（前一半刘怡然，后一半梁厚）  经济（成本：梁厚，财务+风险：孙博闻）  演示（白钰） |  |  |  |  |  |
| 引起变更请求的事件：项目工程管理与过程控制文档结构大幅度修改，经济评价文档重新分工，原有的PPT分工不再适用 |  |  |  |  |  |
| 变更的理由/为什么需要/希望继续/完成项目:  必要性：   1. 需求部分时间比较短，白钰同学既是需求文档的主要作者之一，又有过页面演示经验 2. 作为软管答辩的主体部分，我们原先只为工程管理与过程控制分配了一名同学，工作量不平衡。 3. 梳理经济学文档后，原来负责经济评价答辩的刘怡然发现主要需要讲述的财务分析、成本分析、风险分析部分都不是自己做的 4. 软管文档刘怡然刚刚改过，在这部分工作更多，对内容更加熟悉 5. 梁厚讲完过程的后半段可以接着讲自己写的成本分析模块，每个人讲自己熟悉的部分，顺序更加顺畅   可行性：  PPT分工编写任务由于工作负载过重，几乎全部停滞，因此此时更改计划不会影响已有工作 |  |  |  |  |  |
| 该变化带来的影响：   * 范围：PPT 编写、答辩分工 * 时间安排:每个人的工作内容出现变化，交付时间延后2天 * 花费：无 * 人员: 全体成员 * 风险：无 * 其他：无 |  |  |  |  |  |
| 如果变更通过，应该执行的任务:   1. 修改任务kanban 2. 交换PPT分工 3. 交换答辩顺序 |  |  |  |  |  |
| 所需审批:项目经理 孙博闻 |  |  |  |  |  |
| 姓名/职位 |  |  | 日期 | 通过/拒绝 |  |
| 2053171白钰 | 前端开发工程师 |  | 2023/6/10 | 通过 |  |
| 2053412刘怡然 | 前端开发工程师 |  | 2023/6/10 | 通过 |  |
| 2051840梁厚 | 测试工程师 |  | 2023/6/10 | 通过 |  |
| 2053512孙博闻 | 项目经理 |  | 2023/6/10 | 通过 |  |

10.4.2 **会议纪要**

由于变更提出时团队成员均比较忙碌，且大部分在图书馆等不方便会议交流的位置，所以此次以线上交流群的形式提交变更申请



10.4.3 **变更实施**

**原任务**



**现任务**



11. **项目风险管理**

11.1 **风险管理计划**

11.1.1 **风险评估**

1. 技术风险和资源风险：

* 高并发场景下的稳定性问题：当平台面临大量用户同时访问和使用时，系统可能面临性能瓶颈和稳定性问题，导致平台效率显著下降。
* 知识水平要求高：本实验模块承担对学生实验的引导和教学功能，需要项目团队成员对MARKII实验内容有透彻的了解，否则，可能会造成实验内容出错，影响用户使用。

1. 业务风险：

* 平台效果低于预期：如果平台的功能、用户体验或教学效果不如预期，可能无法吸引足够的用户，影响项目的成功和可持续发展。
* 缺乏足够的用户：平台的成功与否直接关系到用户的数量和活跃程度。如果无法吸引足够的学生和教师使用该平台，可能会影响平台的影响力和经济效益。

1. 管理风险：

* 项目管理不善：不合理的项目规划、进度延误、资源分配不当等可能导致项目无法按计划完成，影响项目的质量和交付时间。
* 资源不足：项目所需的人力、物力和财力资源可能不足，导致项目开发和运营困难，影响项目的推进和可行性。

1. 竞争风险：

* 市场竞争激烈：在虚拟实验教学领域，可能存在其他类似的平台或解决方案，竞争激烈。如果无法提供独特的价值和优势，可能难以脱颖而出并吸引用户。

1. 法律和合规风险：

* 隐私和数据保护：处理用户个人信息和教学数据时，需要遵守相关的隐私和数据保护法律法规，否则可能面临法律风险和用户信任问题。

11.1.2 **风险应对策略**

* 避免风险
* 预防风险
* 转移风险
* 自留风险
* 制定应急措施

针对项目中的各项风险，我们将采取以下策略来应对和减轻其影响：

1. 技术风险和资源风险：

高并发场景下的稳定性问题：我们将进行系统性能测试和优化，确保平台在高并发负载下的稳定性。我们将使用负载测试工具模拟大量用户访问，定期进行性能测试和调整系统配置。

知识水平要求高：团队成员查找资料，一同学习MARKII实验，并共同制定实验页面的全部内容。所有前端页面均需经过同行评审，并与项目相关方交流，取得老师的认可。

1. 业务风险：

平台效果低于预期：我们将与教师和学生保持密切合作，收集他们的反馈和建议，并及时对平台进行改进和优化。我们将进行用户满意度调查和教学效果评估，以确保平台的功能和效果达到预期水平。

缺乏足够的用户：我们将积极推广平台，与学校和教育机构合作，提供培训和支持，以吸引更多的学生和教师使用。我们将开展市场营销活动，并提供具有吸引力的功能和优惠政策来增加用户数量和活跃度。

1. 管理风险：

项目管理不善：我们将制定详细的项目规划和进度安排，并建立有效的沟通和协调机制。我们将定期监测项目进展，并及时采取措施解决潜在的问题和延误。同时，我们将优化资源分配，确保项目所需的人力、物力和财力资源充足。

资源不足：我们将进行充分的资源调查和评估，确保项目所需的人力、物力和财力资源得到合理配置。我们将与相关部门和机构合作，争取额外的资源支持，并进行资源管理和优化，以提高项目的可行性和推进效率。

1. 竞争风险：

市场竞争激烈：我们将不断提升平台的独特价值和优势，与其他竞争对手区分开来。我们将进行市场调研和竞争分析，了解市场需求和趋势，并及时调整我们的产品和服务策略。我们将注重创新，不断推出新的实验模块和功能，以吸引用户并保持竞争力。

1. 法律和合规风险：

隐私和数据保护：我们将遵守相关的隐私和数据保护法律法规，制定合规的数据处理和保护措施。我们将实施严格的访问控制和数据加密，确保用户个人信息和教学数据的安全性和保密性。我们将定期进行安全审计和风险评估，及时修复和改进系统的安全漏洞。

11.1.3 **风险监控措施**

为了及时识别和应对项目中的风险，我们将采取以下监控措施：

* 建立风险监控团队：我们将组建专门的团队负责风险监控和管理工作。该团队将定期审查风险评估和应对策略，并制定相应的监控计划。
* 定期风险评估：我们将定期评估项目风险的发生概率和影响程度，更新风险登记册，并进行优先级排序，以确定风险的处理优先级。
* 监控关键指标：我们将建立关键绩效指标（KPIs），监测项目进展、用户满意度、市场份额等关键指标，以及潜在风险的预警指标。
* 定期沟通和报告：我们将定期召开风险管理会议，与相关各方分享风险情况和应对措施。我们将及时向项目干系人报告风险的变化和处理进展，以保持透明度和沟通畅通。

11.2 **风险识别**

通过进行风险管理规划会议，绘制工作分解结构（工作计划）等工作，识别出以下风险：

|  |  |
| --- | --- |
| 风险序号 | 风险 |
| R1 | 某些人员需要更多的时间适应新的且还不熟悉的软件工具和环境 |
| R2 | 与其他模块间有较强的依赖性，需要频繁修改 |
| R3 | 产品规模比估计的要大，需要的时间比原本的要多 |
| R4 | 需求已经成为项目基准，但需求还在不断更迭与变化 |
| R5 | 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； |
| R6 | 客户对于最后交付的产品不满意，要求重新设计和重做； |
| R7 | 正式上线后的系统有些高性能的要求不能满足 |
| R8 | 功能未明确导致的开发后功能错误 |
| R9 | 缺乏对项目成功标准的定义 |
| R10 | 系统缺乏真实用户的参与 |

11.3 **风险监控**

针对风险值最高的10个风险,对这些风险进行全程监督和控制。为每个风险制订详细的风险应对计划，不断跟踪风险应对的执行情况。同时追踪风险变化的趋势,随时调整每个风险的风险值,并更新十大风险清单。

**项目风险评估与处置表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 风险类别 | 序号 | 风险识别 | 影响结果 | 危险程度 | 解决措施 |
| 财务风险 | 1 | 募集资金不足 | 项目启动时间推迟，错过市场窗口 | 中 | 加大宣传，吸引投资 |
|  | 2 | 后续资金不足 | 项目更新放缓，收入减少 | 中 | 贷款或吸引新投资 |
| 开发风险 | 3 | 代码和库质量低下 | 后续的新实验开发和与原系统 | 中 | 提升员工开发技能 |
|  | 4 | 人员迭代时，新的人员对系统不熟悉 | 工作效率低下，进度延后 | 中 | 对开发人员提前进行培训及提前熟悉软禁工具和环境 |
|  | 5 | 实际实验项目规模大于预估 | 项目进度延期 | 中 | 对过去项目的记录和分析；方法标准化 |
|  | 6 | 模块间依赖性过强 | 频繁修改，进度延后 | 低 | 变更控制规程；增量开发， |
| 市场风险（需求风险） | 7 | 学校对于最后交付的产品不满意 | 重新设计实验ui和过程 | 高 | 原型设计；敏捷开发；用户参与 |
|  | 8 | 学校需求变化 | 项目收益减少，修改和维护成本增加 | 中 | 增量开发，满足市场新需求 |
|  | 9 | 市场上出现其他类似的仿真实验功能产品 | 项目收益减少 | 低 | 增量开发，定位细分市场，加大项目宣传 |
|  | 10 | 正式上线后要求更高的系统性能 | 运营成本增加 | 低 | 原型；提前测试 |
| 其他风险 | 11 | 意外情况导致部分人员无法进行工作 | 项目进度延期 | 中 | 临时招聘人员或其他人员加班 |

12. **项目验收和交付管理**

在项目的不同阶段，针对每个里程碑和交付物，我们可以设定相应的验收标准和交付管理计划，以确保项目按时交付并满足预期要求。

12.1 **交付管理计划**

12.1.1 **项目启动和需求分析阶段**

* 项目启动会议和团队组建 (2023.3)
* 验收标准：成功召开项目启动会议，明确项目目标、角色和责任，完成项目团队的组建，建立相关QQ群和飞书群。
* 交付物：项目启动会议纪要，团队组建文档。
* 完成项目范围和需求分析文档 (2023.4)
* 验收标准：项目范围和需求分析文档完整、准确，并得到相关方（黄杰老师）的确认和批准。
* 交付物：项目范围和需求分析文档，中期答辩PPT。

12.1.2 **设计和开发阶段**

* 用户界面设计和交互设计 (2023.4)
* 验收标准：用户界面设计和交互设计满足用户体验和界面设计要求，得到相关方的认可。
* 交付物：用户界面设计文档。
* 实验报告填写功能开发完成 (2023.4-2023.5)
* 验收标准：实验流程准确无误，实验报告填写功能实现，并经过测试，能够正常使用、保存、编辑和提交实验报告。
* 交付物：实验报告填写功能实现代码。

12.1.3 **测试和调试阶段**

* 前端功能测试和调试 (2023.5)
* 验收标准：前端功能经过测试和调试，能够正常运行，没有明显的错误或异常。
* 交付物：测试报告。
* 用户验收测试和修复 (2023.5-2023.6)
* 验收标准：项目交付物经过相关方验收测试，用户能够使用并确认符合预期要求，修复任何问题和缺陷。
* 交付物：修复和改进的前端代码。

12.1.4 **最终产品演示和交付**

* 最终产品演示和交付
* 验收标准：最终产品（前端页面、文档）经过演示，符合项目需求和规范，得到相关方的认可，可以进行正式交付。
* 交付物：最终产品，包括代码、技术文档、过程管理文档、经济学文档和答辩PPT

12.2 **交付记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 阶段 |  | 交付物 | 交付时间 | 交付方式 | 验收结果 |
| 项目启动和需求分析阶段 | 项目启动会议和团队组建 | 项目启动会议纪要 | 2023.3.25 | 发至团队QQ群审阅 | 通过 |
| 团队组建文档 | 2023.3.25 | 发至团队QQ群审阅 | 通过 |
| 完成项目范围和需求分析文档 | 项目范围和需求分析文档 | 2023.4.15 | 发至团队QQ群审阅 | 通过 |
| 2023.6.12 | 作为期末作业一部分提交canvas | 未知 |
| 中期答辩PPT | 2023.4.17 | 发送到助教邮箱 | 通过 |
| 设计和开发阶段 | 用户界面设计和交互设计 | 用户界面设计文档 | 2023.4.11 | 发至团队QQ群审阅 | 通过 |
| 实验报告填写功能开发完成 | 实验报告填写功能实现代码 | 5.13 | github上传 | 通过 |
| 测试和调试阶段 | 前端功能测试和调试 | 测试报告 | 5.25 | 发至团队QQ群审阅 | 通过 |
| 用户验收测试和修复 | 修复和改进的前端代码 | 5.30 | github上传 | 通过 |
| 最终产品演示和交付 | 最终产品演示和交付 | 代码 | 2023.6.10 | github上传 | 未知 |
| 技术文档 | 2023.6.12 | canvas提交 | 未知 |
| 过程管理文档 | 2023.6.12 | canvas提交 | 未知 |
| 经济学文档 | 2023.6.12 | canvas提交 | 未知 |
| 答辩PPT | 2023.6.16 | 发送到助教邮箱 | 未知 |