**产品完善迭代计划**

　　　　　　　　　　　　　制定日期：2023.7.7

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 2 | 项目名称 | “Testland”--在线代码测评平台 |
| 迭代名称 | 产品完善迭代 | 计划起止日期 | 2023.7.7-2023.9.10 |
| **任务、进度安排和人员分配**  **评测模块**   * 优化评测逻辑 使用docker指令适应不同文件 (7.10) @张奕涵 * 增加编程语言支持(7.11)@张奕涵 * 评测并行化（7.10-7.11）@张奕涵 * 代码安全性检测(7.11-7.14)@张奕涵   **前后端**  **界面优化**   * 管理员界面优化@胡彤 * 排行榜@杨菡雪 * 前端对于还没有返回评测结果的渲染@杨菡雪 * 获取各类别数据统计@全雨乐   **功能拓展**   * 管理员前后端通过压缩包导入题库 @全雨乐 * 前端管理员删题@全雨乐 * 前后端改变题目及测试用例信息@杨菡雪@全雨乐 * 用户上传代码@全雨乐 * 大量数据getList的时候分页传输，减少前后端通信消耗@全雨乐 * 即时搜索，结果高亮@杨菡雪 * 代码版本管理（从历史尝试列表获取某次测评代码）@杨菡雪&@全雨乐 * 题库筛选@杨菡雪 * 回复评论@胡彤 * 用户代码版本管理 @胡彤 * 帖子上传内容图片、内容视频、封面图片@胡彤 * 用户密码安全传输@杨菡雪 * 做题目情况统计从数据库更新 @胡彤 * 雷达图分类从数据库更新@全雨乐@胡彤 * WebSocket @杨菡雪   **前后端协同**  **Bug修复**   * 未登录用户编辑代码提醒登录bug @胡彤 * 前端导航栏高亮bug @胡彤 * markdown @胡彤   **部署**   * 将项目总体部署到服务器@张奕涵   **测试**   * 单元测试@全雨乐 * 大数据量性能测试@全雨乐 * 集成测试@张奕涵 * 系统测试@杨菡雪@张奕涵@全雨乐@胡彤 |  |  |  |
| **预期成果**   * 产品完善迭代计划 * 产品完善迭代评估报告 * 部署到服务器的产品代码 * 单元测试报告 * 性能测试报告 |  |  |  |
| **主要的风险和应对方案**  **技术风险**   1. 技术选型风险：选取的技术可能无法满足项目需求或者存在严重的性能问题。为了缓解这个风险，我们可以在选择技术时进行充分的调研和评估，包括对比不同技术的优缺点和实际应用案例。同时，可以进行一些小规模的原型验证，以确定技术的可行性和适用性。此外，在每一次评审中积极向助教和老师寻求帮助，确认当前技术方案的可行性。 2. 集成风险：由于项目中需要整合多个组件或服务，可能存在集成过程中的兼容性、接口定义不一致等问题。为了降低集成风险，可以在开发过程中尽早进行项目集成，确保各个组件能够正确地协同工作。 3. 未识别的技术风险：部分功能可能实际实现技术较为复杂，但是在制定计划时没有识别到。为了降低这种风险，可以在开发过程中增加交流，一起解决过程中新出现的风险。   **进度风险**   1. 第三方依赖问题：项目中可能涉及使用第三方的组件、服务或库，而这些第三方可能出现不稳定、不可用或存在安全漏洞等问题。为了应对这个风险，可以在选择第三方依赖时进行充分的评估和测试，选择稳定可靠的版本，并及时更新和修复存在的问题。同时，建立灵活的组件替换策略，可以在必要时快速切换到其他可用的解决方案。 2. 分工与协同问题：部分功能的实现可能对其他功能有一定依赖性，必须等待其他功能实现方可添加。部分功能需要对此前已实现的功能进行修改和返工，造成了额外的工作量，因此可能增大进度风险。对于分工与协同问题，我们采用在线文档和立会同步任务进度，采用代码托管仓库同步代码，并且在过程中灵活地分配任务，减少了分工与协同问题。   **质量风险**   1. 缺乏充分的测试覆盖：如果测试不够充分，可能会导致潜在的缺陷未被发现，影响系统的稳定性和可靠性。为了降低质量风险，可以制定详细的测试计划，包括单元测试、集成测试和系统测试等，并建立自动化测试框架，确保代码的质量和功能的完整性。 2. 代码质量和规范性问题：如果开发过程中忽视了代码质量和规范性，可能会导致代码难以维护和扩展，增加后续开发的风险。为了应对这个风险，可以制定代码开发规范和最佳实践，并进行代码审查和静态代码分析等，确保代码的质量和可维护性。   **安全风险**   1. 用户信息安全问题：如果在设计和实现过程中未考虑用户信息安全，可能导致用户数据泄露或被非法使用。为了减轻安全风险，需要采取适当的安全防护措施，例如数据加密、访问控制和安全审计等。 2. 漏洞和攻击风险：由于系统存在漏洞或未经授权的访问，可能会导致系统遭受恶意攻击，例如网络攻击、跨站脚本攻击等。为了应对这个风险，需要进行安全测试和漏洞扫描，并及时修复发现的安全问题。 |  |  |  |