8月20号 星期二

1.今天做了什么

今天我学习了产品经理的工作内容和技能要求，了解了产品经理的基础知识

2.今天最有收获的点

对产品经理所需的多样化技能有了清晰的认识，并掌握了如何通过有效的需求分析推动产品开发。

3.今天解决了什么问题，解决方案

问题：在了解产品经理的技能要求过程中，对如何进行有效的需求分析存在困惑。

解决方案：通过查阅相关资料和案例，深入了解了需求分析的步骤，包括需求收集、需求整理、需求优先级排序和需求文档编写。具体步骤包括：

需求收集：通过用户调研、市场分析和竞品分析获取需求信息。

需求整理：将收集到的需求进行分类和整理，确定每项需求的背景和业务价值。

需求优先级排序：根据需求的重要性和紧急性进行优先级排序，使用工具如MoSCoW方法来确定哪些需求是必须的、应该的、可以的和不会做的。

需求文档编写：编写清晰详细的需求文档，包括需求背景、功能描述、用户故事和验收标准，确保开发团队理解并能准确实施需求。

4.今天还有什么问题未解决，接下来大致计划

仍需深入了解如何有效与开发团队沟通和协作，以及如何处理需求变更和冲突。接下来计划通过参加相关培训和阅读案例研究，提升在这些领域的能力。

8月21日 星期三

1. 今天做了什么

今天我了解了互联网项目管理的基本要素、常用工具以及流行的开发模式。通过学习，我掌握了 GitHub 的基本使用方法，特别是在合作开发中的应用。重点学习了如何创建和管理仓库、分支的操作以及拉取请求的处理。

2. 今天最有收获的点

掌握了 GitHub 的基本操作流程，尤其是分支管理和拉取请求的创建，这对未来的团队合作开发至关重要。

3. 今天解决了什么问题，解决方案

问题：在使用 GitHub 进行分支管理时遇到了一些困惑，特别是如何有效地管理多个分支以及处理分支合并时的冲突。

解决方案：

为了提高对 GitHub 分支管理的理解，我通过以下步骤解决了相关问题：

创建和管理分支：

创建分支：使用命令 git checkout -b <分支名> 创建新的分支，并立即切换到该分支。这样可以在不影响主分支的情况下进行开发。

切换分支：使用 git checkout <分支名> 切换到已有的分支进行工作。

删除分支：在不再需要的分支上使用 git branch -d <分支名> 删除本地分支，用 git push origin --delete <分支名> 删除远程分支。

处理分支合并：

合并分支：在主分支上使用 git merge <分支名> 合并指定的分支。合并时可能会遇到冲突，需要手动解决冲突并提交。

解决冲突：Git 会标记冲突文件，并在文件中显示冲突标记。需要手动编辑文件，删除冲突标记，保存更改，然后使用 git add <文件名> 和 git commit 提交解决后的文件。

创建拉取请求：

创建：在 GitHub 页面上选择要合并的分支，点击 “Pull request” 按钮，填写标题和描述，选择目标分支，然后提交拉取请求。

审查和合并：团队成员可以在拉取请求中进行代码审查，讨论更改，最终由有权限的人员合并拉取请求。

通过这些操作，我有效掌握了如何在团队开发中使用分支进行并行开发，并解决了分支合并过程中的冲突问题。

4. 今天还有什么问题未解决，接下来大致计划

今天主要集中在学习 GitHub 的基本操作和解决分支管理的问题。接下来，我计划进一步了解 GitHub Actions 的自动化功能，以提高开发流程的效率。同时，探索如何更好地管理大型项目中的任务和问题，特别是如何使用 GitHub Issues 跟踪和管理开发任务。

8月22日 星期四

1.今天做了什么

今天深入学习了高质量软件开发的两大方面：软件工程方法和个人代码质量提升。具体包括研究了软件工程中的质量保障流程，以及编写和审查高质量代码的最佳实践。

2.今天最有收获的点

理解到高质量软件开发不仅依赖于工程方法，也需要持续改进个人的编码能力和习惯。

3.今天解决了什么问题，解决方案

问题：在软件开发过程中，发现部分代码质量不高，导致维护困难。

解决方案：实现了一套代码审查流程以提升代码质量，包括建立代码规范、引入自动化代码检查工具和定期进行代码审查会议。通过代码规范化，自动化工具（如静态分析器）可以提前发现潜在问题，而代码审查会议则促进了团队的知识共享和最佳实践的推广。

详细说明：

代码规范：制定了一套详细的编码标准，涵盖命名规则、注释规范、函数和类的设计原则等。

自动化工具：集成了静态代码分析工具（如 SonarQube），在每次提交代码时自动进行检查，及时反馈代码问题。

代码审查：每周组织一次团队代码审查会议，讨论高质量代码的最佳实践和当前项目中的改进点。

4.今天还有什么问题未解决，接下来大致计划

目前尚未完全解决的主要问题是如何将新引入的流程有效融入现有的开发周期中。接下来的计划是调整团队工作流程，逐步实施新的审查机制，并收集反馈进行优化。

8月23日 星期五

1. 今天做了什么：

今天，我深入了解了不同测试岗位的工作职责，包括功能测试、性能测试和自动化测试等。同时，我掌握了测试用例设计的基本方法，学习了如何编写有效的测试用例以确保软件的质量。

2. 今天最有收获的点：

我学会了如何设计详细且高效的测试用例，这对提高测试覆盖率和发现潜在问题非常重要。

3. 今天解决了什么问题，解决方案：

今天，我面临的主要问题是如何将测试用例设计理论应用到实际的测试工作中。经过调研和实践，我采用以下方法来解决这个问题：

确定测试目标： 首先明确测试的目标是什么，确保测试用例能够覆盖所有重要的功能点和需求。

设计测试用例： 根据功能需求文档，设计详细的测试用例，涵盖正向测试和负向测试场景。测试用例包括输入数据、预期结果以及测试步骤。

测试用例优化： 通过将相似的测试用例合并，避免冗余，并确保测试用例的可维护性。

编写测试用例文档： 将测试用例文档化，确保团队成员能够清晰理解每个测试用例的目的和操作步骤。

4. 今天还有什么问题未解决，接下来大致计划：

虽然我已经掌握了基本的测试用例设计方法，但在实际应用中还需要进一步提升效率。接下来的计划包括：

进行更多的实际操作： 在实际项目中应用测试用例设计方法，积累更多经验。

学习自动化测试工具： 探索和学习自动化测试工具，提升测试效率。

参与团队讨论： 向经验丰富的同事请教，获取关于测试用例设计和实施的宝贵建议。