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。