**软件过程成熟度评估与改进计划报告**

**——基于CMMI模型的个人项目分析**

**一、CMMI层次成熟度模型简述**

CMMI（Capability Maturity Model Integration，能力成熟度模型集成）是由美国国防部资助、卡内基梅隆大学软件工程研究所（SEI）提出的一套用于改进软件过程能力的评估与改进模型。它为组织在软件开发和项目管理中提供了一套逐步演进、结构化的过程改进框架。

CMMI的成熟度模型分为**五个层级**，从最低的初始级到最高的优化级，分别如下：

1. **初始级（Level 1）——无序/不可预测：**  
   软件过程混乱不堪，缺乏文档记录，开发依赖个人能力，项目成功不可复制。
2. **可管理级（Level 2）——基本项目管理：**  
   已建立基本的项目管理机制，有计划、有监督，过程可追踪，文档完整，能在类似项目中复制经验。
3. **已定义级（Level 3）——标准化过程：**  
   软件过程标准化、制度化，团队遵循组织级标准流程，有明确的角色、规范与文档模板。
4. **量化管理级（Level 4）——可度量：**  
   对过程与产品质量进行度量与统计分析，通过数据预测项目风险、控制质量。
5. **优化级（Level 5）——持续改进：**  
   持续基于度量数据进行过程优化，采用创新技术、改进实践以提升组织整体能力。

该模型不仅适用于大型企业，也适合指导高校学生或团队在课程设计、大创项目等非正式开发中的实践与改进。

**二、个人项目的软件过程成熟度评估**

在过去三年中，我参与了多个开发项目，包括如下两个典型案例：

* \*\*软件工程课程项目：\*\*独立开发一个图书管理系统，包括用户模块、借阅模块、管理员模块，实现基本的增删查改功能；
* \*\*全国大学生创新创业训练项目（大创）：\*\*参与一个基于深度学习的图像识别Web平台开发，担任后端与模型训练负责人。

通过对这两个项目的软件过程回顾，结合CMMI模型的五个层级，我评估我所在项目团队当前的过程成熟度大致处于**Level 2：可管理级**，表现如下：

**（1）过程管理基本建立，但仍不稳定**

在项目初期，我们制定了项目计划与分工表，使用 Trello 管理任务分发，采用 Git 进行版本管理，使用文档记录需求与会议纪要。但过程中仍存在以下问题：

* 计划随意调整，变更不及时同步；
* 缺乏正式的代码审查机制；
* 部分成员对流程理解不一致，执行差异较大。

**（2）文档与追踪能力有但不完善**

我们在大创项目中建立了开发手册与测试记录文档，但格式非标准化，内容较为粗糙。对于需求变更、缺陷修复过程无详细记录。

**（3）标准流程缺失，过程依赖经验**

虽然我们尝试使用敏捷开发，但缺乏Scrum的具体实践，如每日站会、燃尽图等未系统使用，很多决策仍依赖核心成员的直觉判断。

综上，我的开发实践虽然已迈过“初始级”的混乱阶段，具备一定的管理基础，但还未建立统一标准的流程，且缺乏量化分析与持续优化意识，因此属于CMMI的**Level 2**阶段。

**三、基于当前成熟度的过程改进方向**

为了向更高的成熟度水平（Level 3 甚至 Level 4）迈进，我将从以下三个方面进行过程改进：

**1. 标准化开发流程（向 Level 3 过渡）**

* 建立统一的项目开发模板，如需求文档、设计文档、测试计划书；
* 明确版本控制规范与分支策略，如 Git Flow；
* 全员培训 Scrum 的关键流程（冲刺计划、站会、回顾会议），推动敏捷实践落地。

**2. 加强质量保证机制**

* 引入 Code Review 流程，使用 GitHub Pull Request + Review；
* 增加单元测试与自动化测试覆盖率，确保功能可验证；
* 引入 CI 工具（如 GitHub Actions）自动运行测试与部署。

**3. 初步引入度量体系（迈向 Level 4）**

* 对开发任务进行工作量估算与工时记录；
* 使用 Bug 跟踪工具（如 Jira）统计缺陷率；
* 跟踪 Sprint 完成率、测试覆盖率等关键指标，形成项目数据分析基础。