# 《软件过程管理》学期论文

## CMMI层次成熟度模型概述

CMMI（Capability Maturity Model Intergration，能力成熟度模型集成）是美国卡内基·梅隆大学软件工程研究所（SEI）提出的用于改进组织软件过程能力的一种模型。CMMI将软件过程划分为五个成熟度等级，每个等级对应着一个组织在软件工程过程中的能力水平与标准化程度。

等级1：初始级（Initial）

过程是随意且混乱的，组织往往不能提供一个稳定的环境来支持过程，这些组织的成果过分依赖于组织内人员的能力和个人英雄主义，而不是使用经过实践证明的过程。一旦关键人员离职，项目质量很容易受到影响。

等级2：已管理级（Managed）

项目确保其过程按照方针得到计划与执行，并开始建立基本的项目管理机制，如需求文档、进度计划、项目周报、配置管理等，过程可控但仍不稳定。

等级3：已定义级（Defined）

过程得到清晰的说明与理解，并以标准、规程、工具与方法的形式进行描述。软件过程被文档化、标准化，项目团队使用组织级的标准流程来开发产品，并对各个项目的流程进行统一管理。

等级4：量化管理级（Quantitatively Managed）

组织与项目建立了质量与过程性能的量化目标并将其用作管理项目的准则。组织对软件过程和产品质量进行度量分析，基于数据进行管理和优化。

等级5：持续优化级（Optimizing）

组织基于对其业务目标与绩效需要的量化理解，通过持续改进机制对流程进行创新性优化，具备自适应能力和风险预测能力。处于成熟度级别5 级时，组织使用从多个项目收集来的数据对整体的组织级绩效进行关注。对数据的分析识别出绩效方面的不足与差距。这些差距用于驱动组织级过程改进，并产生绩效方面的可度量的改进。

## WanderTrack项目的过程成熟度评估

### 2.1 项目概述

WanderTrack 是我在《软件项目管理》这门课中与团队合作开发的一个项目，目标是设计一款面向户外爱好者的轨迹记录与成就管理 APP。开发周期从2025年3月到2025年5月，共约两个月。该项目使用 Android 原生平台进行前端开发，后端采用 Flask + SQLite。团队共 5 人，我担任项目经理与后端开发主责，主要负责项目进度规划、服务器搭建、数据库设计及 API 实现。

### 2.2 过程特征分析

从 CMMI 的五级模型视角出发，对我们这个WanderTrack 项目的过程进行如下评估：

1.项目计划与进度管理：

初期制定了较明确的开发时间表，并通过甘特图进行阶段划分（立项 → 需求分析 → UI 设计 → 开发实现 → 联调测试），进度更新由我负责定期同步团队成员。具备基本的可控性，符合 Level 2 要求。

2.角色分工与协作机制：

团队成员职责清晰：我负责后端开发与项目协调，其他成员分工为前端实现、数据分析与功能测试。但在协作过程中缺少严格的任务跟踪机制，例如没有采用 Issue 工具管理任务状态，靠QQ群内文字沟通为主。

3.文档管理：

需求分析文档、API 设计文档由我主导撰写，并上传至腾讯文档云共享平台。但部分同学提交的设计说明较简略，文档模板不统一，后期查阅效率较低，说明项目过程仍缺乏组织级的标准文档体系。

4.代码管理与版本控制：

我们的项目使用 Git 进行代码管理，并创建多个分支进行模块协作。但未设置代码审查流程，分支合并依赖成员自检，存在提交不规范的情况（如注释缺失、格式不统一等）。

5.测试策略：

项目中负责测试的同学主要采用手动方式进行功能验证，覆盖了基本路径但缺乏系统化测试计划；后端未建立自动化单元测试框架，Bug 修复主要依赖手动发现与反馈。

6.度量指标与过程控制：

我们的项目缺乏对开发效率、缺陷密度、测试覆盖率等数据的系统记录与分析，因此不具备量化过程管理能力。

综合以上的分析，我们这个WanderTrack 项目具备了基本的项目管理流程，过程可重复，但因为我们都比较缺乏完整的团队项目开发经验，项目仍缺乏规范与度量手段。整体过程成熟度评估为 Level 2：已管理级。

## 项目存在的问题与改进方向

经过分析后发现我们这个WanderTrack项目当前存在的主要问题：

1.缺乏统一的开发流程与文档模板

成员各自书写文档的风格和所使用的模版不一致，不利于团队协作与文档积累。

2.协作管理缺少工具化支持

没有使用项目管理平台（如 Trello、GitHub Projects）统一安排与跟踪任务，而是选择了非正式QQ群聊发布任务与跟踪进度，信息容易丢失。

3.缺乏代码规范与审查机制

合并分支未设置代码审查，导致代码的风格不统一，潜在的Bug难以及时发现。

4.测试过程较随意

多数测试仅在功能完成后手动操作验证，没有设计系统的测试用例与自动化测试机制。

5.过程无量化管理能力

在项目开发过程中没有收集关键指标如缺陷率、迭代效率等，后期难以总结与复盘。

未来基于CMMI的改进方向与计划：

|  |  |  |
| --- | --- | --- |
| **改进方向** | **具体措施** | **预期效果** |
| 文档标准化 | 统一撰写格式与模板，规定项目初期需提交的文档列表 | 提高可读性和查阅效率 |
| 协作工具引入 | 使用 GitHub Project 或 Notion 管理任务状态与进度 | 任务更加清晰明了，协作过程更加顺畅 |
| 代码审查机制引入 | 强制 PR 审核制度，引入静态分析工具（如 Checkstyle） | 代码质量提升，减少合并时的冲突 |
| 自动化测试 | 为后端编写单元测试框架，引入 Postman API 测试脚本 | 降低回归成本，提高稳定性 |
| 过程量化与复盘 | 建立简易指标体系（如每周提交次数、测试用例覆盖率） | 用于项目总结，一遍遍经验积累与复用 |

## 总结

通过使用CMMI模型对我们的WanderTrack项目的软件过程进行评估，我认识到虽然我们已经具备了初步的管理意识，但因为缺乏相关的经验，在标准化、协作规范与测试体系等方面仍有较大提升空间。未来在类似的项目中，我会更加重视过程建设，通过工具与制度配合，提高项目效率与产品质量，让过程本身成为成功的保障。