

CMMI 成熟度模型与个人软件过程评估报告

姓名：罗皓天

学号：2021141450210

一、CMMI 成熟度模型简介

CMMI (Capability Maturity Model Integration, 能力成熟度模型集成) 是一种评估与提升组织软件过程能力的模型。其“成熟度等级”采用分阶段结构, 将组织的过程能力划分为以下六个级别:

| 成熟度等级 | 名称 | 特点概述 |
|-------|-------------------------------|--------------------------------|
| 0 | Incomplete (不完整) | 过程没有被落实或无序, 缺乏稳定性与可预测性。 |
| 1 | Initial (初始) | 依赖个人能力完成工作, 流程随意, 风险高、质量不可控。 |
| 2 | Managed (已管理) | 具备项目级流程规划、执行与监控, 关键过程受到控制。 |
| 3 | Defined (已定义) | 建立组织层面标准流程并推广使用, 强调流程文档化和可复用性。 |
| 4 | Quantitatively Managed (量化管理) | 以数据和量化指标驱动流程控制, 提升可预测性。 |
| 5 | Optimizing (优化) | 持续改进, 鼓励创新与敏捷响应变化。 |

二、个人软件开发过程的成熟度评估

评估对象：本人在大模型科研项目中的开发实践

根据我最近一年的科研经历, 我对自己的科研项目“开发”过程进行了反思和总结, 并进行了成熟度评估。

1. 前期调研与选题确定

- 在研究方向确定前, 仔细检索和阅读arXiv及主流人工智能学术会议 (如 NeurIPS、ICML、ACL 等) 上的论文。
- 完成对现有研究的调研, 总结现有 baselines、研究现状和研究空缺。
- 明确研究方向, 确定研究目标和研究意义, 评估研究对现有技术和研究方向的潜在价值。

2. 可行性验证

- 在确定研究方向后，构建极小化、简化版的实验原型。
- 用简单代码验证核心技术或思想的可行性，确定研究方向的可行性和潜力。
- 完成第一次概念验证，为后续大规模开发提供参考依据。

3. 项目初始化与版本控制

- 创建专用 GitHub 私有仓库，确定代码结构和主要模块。
- 完成配置管理和版本控制，确定代码提交和协同管理规则。
- 配置代码检查和文档自动化构建等辅助工具，建立标准化研发环境。

4. 实验开发与方法迭代

- 完成研究代码结构和核心模块的设计及构建。
- 在确定研究方向后，进入迭代开发阶段，持续完善代码结构和参数配置。
- 完成对比实验设计，撰写脚本自动化管理训练及验证过程。
- 定期提交代码和实验配置至 GitHub，维护代码和实验记录，确保研究过程可追溯。

5. 论文撰写

- 在确定研究结果后，整理实验结果及参考文献，撰写论文。
- 完成论文结构设计、实验结果展示、附录撰写及论文排版。
- 完成论文提交至预印本平台（如 arXiv），为研究成果提供更大曝光和交流空间。

6. 开源与文档编写（如有必要）

- 完成代码整理及开源，撰写完善的 README、使用指导和示例代码。
- 整理和共享研究中的训练数据、配置和脚本，确保研究可复现。
- 在论文附录或额外文档中补充研究过程中发现的关键细节和设计理念，完善研究参考资料。

CMMI 成熟度评估结论：Level 2 – Managed（已管理级）

| 评估维度 | 体现情况 |
|---------|------------------------------|
| 流程可重复性 | 调研 → 验证 → 实验 → 总结的闭环流程具备一致性。 |
| 项目管理机制 | 使用 GitHub 进行版本控制与项目归档。 |
| 开发可控性 | 实验设计目标明确，开发过程规范有序。 |
| 缺乏组织级流程 | 没有统一模板、标准测试规范、质量保障流程。 |
| 缺乏流程制度化 | 项目间方法传承依赖记忆，尚未形成制度资产。 |

三、过程改进与提升计划（迈向 CMMI 3）

为进一步提升软件开发成熟度，我可以采取以下措施，让流程的成熟度接近 CMMI Level 3（已定义级）：

改进方向一：标准化项目结构

- 制定统一的项目模板，涵盖以下内容：
 - `README.md`：项目说明
 - `requirements.txt`：依赖说明
 - `experiments.log`：实验记录
 - `src/`、`data/`、`results/` 等标准目录结构（目前尚不具备）

改进方向二：文档与日志制度化

- 每个版本提交附带变更说明（changelog），commit的时候要写清楚内容，不能追求效率就忽略这一部分。
- 每次，每组实验需记录是实验的设计、结果和评估

改进方向三：核心代码测试机制引入

- 为核心数据处理与评估函数添加单元测试（pytest 等工具）
- 保证代码稳定性与可移植性，避免在切换服务器或者环境的时候出现兼容性的问题。

改进方向四：个人流程资产沉淀

- 形成“最佳实践清单”与“常用脚本库（目前已有一个不规范的初级形式）”
- 记录常见问题与解决方案，提高开发效率
- 准备conda、docker环境模板，bii

总结

目前我的软件开发实践已达到 **CMMI Level 2**，具备可控性与可重复性。下一阶段的目标是建立组织级流程标准与开发制度化机制，以此推进向 **CMMI Level 3** 的跃升。这有助于提升科研项目效率，也将强化过程复用能力和跨项目协同能力，避免重复造轮子。