

# 学期论文

## 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) 是一个用于改进组织软件开发流程的国际通用模型。它由卡内基梅隆大学软件工程研究所 (SEI) 开发, 用于评估和改进软件开发组织的能力水平。

CMMI 将软件过程划分为五个成熟度等级, 形成一个逐步演进的管理和控制机制:

1. 初始级 (Level 1 - Initial):
  - 过程是不可预测的、反应性的。
  - 成功主要依赖个人能力。
  - 项目常因缺乏管理控制而失败。
2. 可管理级 (Level 2 - Managed):
  - 有基本的项目管理流程, 能跟踪成本、进度和功能。
  - 有组织的过程执行, 但主要关注项目层面。
3. 已定义级 (Level 3 - Defined):
  - 过程文档化、标准化, 形成组织级的标准软件过程。
  - 所有项目遵循组织标准过程定制的版本。
4. 量化管理级 (Level 4 - Quantitatively Managed):
  - 对过程和产品质量有量化的理解和控制。
  - 能预测性能, 并对变化做出快速反应。
5. 优化级 (Level 5 - Optimizing):
  - 组织不断改进过程以适应外部变化和内部优化。
  - 创新技术和过程改进是驱动因素。

## 二、我在个人项目中的开发成熟度评估

以我主导开发的智能云图库系统 (基于 SSM + Redis + COS + WebSocket) 为例, 我对个人开发的全流程进行了评估:

当前水平: 接近 CMMI Level 3 (已定义级)

1. 项目计划:
  - 有初步的开发计划 (功能拆解、阶段性目标), 但缺少书面形式 → Level 2
  - 常通过 TODO 标注功能点推进, 存在任务遗漏 → 需要改进
2. 技术选型与架构:
  - 使用主流框架组合 (Spring Boot + MyBatis-Plus + Redis), 合理规划 Controller-Service-DAO 层次结构 → Level 3
  - 架构未书面文档化, 后期查阅困难 → 尚未系统沉淀
3. 代码管理:
  - 使用 GitHub 进行版本管理, 具备主分支与开发分支结构 → Level 3
  - 未写提交日志规范, commit 信息较杂 → 有改进空间
4. 模块设计:
  - 基本遵循单一职责, 封装工具类, 复用性高 → Level 3
  - 服务之间耦合度略高, 尚未拆分为独立服务 → 有待提升

5. 测试与调试：
  - 编写少量 JUnit 单元测试，主要依赖手动 Postman 测试 → 处于 Level 2
  - 无测试覆盖率数据，异常路径测试不足 → 不具备 Level 4 特征
6. 日志与异常处理：
  - 使用 SLF4J + logback 日志体系，记录接口访问、异常信息 → Level 3
  - 日志级别控制良好，但缺乏链路追踪 → 待量化提升
7. 部署与上线：
  - 成功部署到腾讯云，使用 Nginx 和 COS，能进行生产级打包和部署 → Level 3
  - 缺少 CI/CD 自动化流程，部署步骤人工操作繁琐 → 未达 Level 4

### 三、过程改进目标与计划（向 Level 4 演进）

**【目标】：**提升个人开发的自动化能力、度量能力和规范意识，过渡至“量化管理级”。

#### 1. 过程度量改进

改进内容	当前情况	改进方式
缺陷统计	Bug 修复靠手记	每次测试记录缺陷来源、修复方式、耗时
代码覆盖率	无系统记录	引入 Jacoco 工具，结合 Maven 插件统计覆盖率
模块耗时统计	无数据	每完成一个模块记录开发耗时 + 测试耗时

#### 2. 测试能力改进

改进内容	当前情况	改进方式
单元测试覆盖	仅覆盖主要方法	对服务类与工具类补全测试用例
接口测试	主要靠 Postman	使用 MockMvc + 自动化测试脚本
异常测试	未覆盖	对所有边界值和错误输入补充测试场景

#### 3. 自动化与持续集成改进

改进内容	当前情况	改进方式
构建流程	手动打包、上传	使用 GitHub Actions 实现打包构建
测试流程	手动测试	构建前执行自动单元测试
结果通知	无反馈机制	构建成功或失败通过钉钉或邮件通知

#### 4. 文档与规范改进

改进内容	当前情况	改进方式
需求文档	无	使用 Typora 编写 Markdown 格式需求说明书
接口文档	使用 Swagger, 但未导出	定期导出 Swagger HTML 并同步到文档平台
提交规范	commit 信息不规范	引入 conventional commit 模式, 如 feat: fix:

## 5. 流程管理改进

- 每次开发前撰写《任务计划表》(功能点、预估时间、依赖项)
- 每周进行“开发日记”记录: 总结本周完成任务、问题、反思
- 设定阶段里程碑检查点: 阶段性评估功能进度 + Bug 累计情况