CMMI 简述

CMMI 是一种用于评价和改进组织软件过程能力的模型框架，其广泛应用于软件开发与项目管理领域。CMMI 将过程成熟度划分为五个等级，从低到高可分为：

第一级：初始级（Initial）
过程缺乏规范与纪律，往往依赖个人英雄主义，项目实施过程难以预测。项目目标、进度、成本等往往因人员经验差异而波动较大。管理方式随意性较强，缺少正式文档、标准与度量。

第二级：可管理级（Managed）
表现为已经建立了项目级别的基本管理过程，并对项目需求、计划、跟踪和质量保证予以管控。能够在项目范围内对需求和资源进行规划，建立基本的度量指标，能够按计划交付版本。关键过程域包括了需求管理、项目规划、项目监控与控制、供应商合约管理、度量与分析、过程与产品质量保证以及配置管理等。

第三级：已定义级（Defined）
在组织层面制定并推广标准化的过程体系，所有项目都必须在此基础上进行定制。通过采用组织级别的过程资产库，不同项目可以复用经验，且过程步奏与产品质量更加稳定。关键过程域包括组织过程聚焦、组织过程定义、培训、集成项目管理、风险管理、决策分析与解决、组织环境对 SM 的支持等。

第四级：量化管理级（Quantitatively Managed）
除了组织级的标准过程以外，还在各项目层面进行定量化管理，通过统计与度量数据来控制项目。关键过程域引入统计过程控制，依靠度量数据识别过程变异，并通过分析降低波动，确保稳定性。关键过程域主要包括组织绩效管理和量化项目管理，要求组织收集历史数据，建立过程性能基线，并对项目进行预测与调优。

第五级：持续优化级（Optimizing）
其在已有的量化管理基础上，组织持续进行过程改进，通过因果分析和技术创新来不断提高流程性能和产品质量。过程绩效方面关注识别过程瓶颈与潜在风险，运用缓解策略与创新技术持续优化；对改进活动进行度量与评估，并将结果反馈到过程资产库。关键过程域包括因果分析与解决以及组织创新与部署

这五级模型体现了组织从“无序混乱”到“标准可控”，再到“量化管理”与“持续改进”的演进过程。每个成熟度级别均对应一组关键过程域，帮助组织建立、巩固和提升软件过程能力。

个人项目评估:数据库大作业--学生住宿管理系统

过程域（PA）	所属级别	现状摘要	得分
需求管理（REQM）	Level 2	编写简单需求文档，包含三种角色用例，绘制相关 ER 图，并定期手动同步文档。	2
项目规划（PP）	Level 2	制定阶段性计划，将前端后端分阶段处理，但无细化任务工时与验收标准。	1
项目监控与控制（PMC）	Level 2	会定期整理进度，与组员交流	1
配置管理（CM）	Level 2	使用 Git 管理项目代码，但未强制 Pull Request 审查；数据库脚本也无版本管理。	2
过程与产品质量保证（PPQA）	Level 2	组内 Code Review 检查功能；无自动化单元测试或静态分析，测试以手动为主。	1
度量与分析（MA）	Level 2	仅记录开发进度和 Bug 数，无代码覆盖率、数据库性能等度量，无法趋势分析。	0
风险管理（RSKM）	Level 3	项目初期口头讨论潜在风险，未形成书面风险登记表，也无持续跟踪与应急计划。	0
组织过程定义（OPD）/ 组织过程聚焦（OPF）	Level 3	无成文的过程指导手册、模板或知识库；项目总结仅是一份实验报告，缺乏跨项目复用。	0
集成项目管理（IPM）	Level 3	仅在本地环境运行，无多用户并发测试；缺少完整部署脚本与环境说明。	0
决策分析与解决（DAR）	Level 3	技术选型（Swing + JDBC + MySQL）基于经验，无多方案对比与决策记录。	0
量化项目管理（QPM）/ 组织绩效管理（OPM）	Level 4	无度量基线与预测模型；不具备定量分析进度或质量的能力。	0
因果分析与解决（CAR）	Level 5	未做根因分析或形成改进报告，仅有个人心得。	0
组织创新与部署（OID）	Level 5	未引入新技术或新工具，也无创新评估机制。	0

总体来看此项目仅有限落实 Level 2 的关键过程（需求管理、配置管理）的部分要求，但在项目规划、监控、质量保证等方面深度不足；同时并未触及 Level 3 及以上（量化、持续改进）过程域，所以自评软件整体成熟度位于 CMMI Level 2

改进思路与计划

1. 完善 Level 2 关键过程域
需求管理优化：将需求文档放入 Git 仓库，使用分支与提交记录跟踪变更；简单引入 RTM，将需求 ID 与用例、设计模块、测试用例关联。

项目规划优化调整：细化任务到“周/日”，使用 GitHub Project 看板管理 Issue，实现“待办→进行中→已完成”的完整可视化流程；

配置管理方面，强制 Pull Request 审查，至少一人 Review 并通过 CI 检查后才能合并；将数据库建表和初始化脚本纳入 Git，必要时使用 Flyway 或 Liquibase 管理表结构演进。

2. 初步探索 Level 3 及以上过程
组织过程定义（OPD）与组织过程聚焦（OPF）：撰写《Java Swing + JDBC 项目开发指导手册》，包括需求模板、设计模板、测试模板、编码规范等；将该手册与模板存放于团队公共 Git ，让新项目快速复用和更新迭代。

尝试集成项目管理（IPM）：搭建一台测试服务器或使用云主机，模拟多用户并发操作；
编写部署文档（数据库初始化脚本、JDK/Tomcat 版本以及环境变量），并进行端到端集成测试。

组织创新与部署（OID）：小范围尝试新的技术栈（如 Spring Boot + Vue 前端、JavaFX 前端），并评估其开发效率与运行效果；形成《技术创新评估报告》，为组织后续重大技术选型提供决策依据。