# 基于CMMI模型的软件过程成熟度评估与改进——以“老人智慧管理系统”开发

## 一、CMMI层次成熟度模型简述

CMMI（Capability Maturity Model Integration）即能力成熟度模型集成，是衡量软件开发组织过程改进和能力成熟度的国际标准框架。其核心的阶段式成熟度模型分为五个等级，每个等级代表过程管理和质量控制能力的显著提升：

1. 初始级（Level 1 - Initial）：

过程不可预测、缺乏控制，依赖个人英雄主义。成功偶然，项目常超期超预算。

特点：无序、被动响应、无稳定环境。

1. 可重复级（Level 2 - Managed）：

基于过往经验，能对基本项目管理过程（如需求、计划、跟踪、配置管理、质量保证）进行纪律化，在类似项目中可重复成功。

特点：项目级管理、过程制度化、注重基线控制。

1. 已定义级（Level 3 - Defined）：

建立组织级标准过程（OSSP），项目可据此裁剪定制。过程被清晰定义、理解并文档化。

特点：组织级标准、过程资产库、主动预防缺陷。

1. 量化管理级（Level 4 - Quantitatively Managed）：

运用统计技术对过程和质量进行量化管理，设定量化目标并监控偏差。

特点：数据驱动决策、过程性能模型、预测能力增强。

1. 优化级（Level 5 - Optimizing）：

持续基于量化反馈，采用创新技术系统性优化过程，消除根本性缺陷。

特点：持续改进、技术创新、缺陷预防常态化。

## 二．过往项目过程成熟度评估：计算机设计大赛项目“老人智慧管理系统”

在开发“老人智慧管理系统”（子女端/Ionic-Android、老人端/Ionic-Android、管理端/Electron-Windows）的过程中，我评估团队的过程成熟度大致处于CMMI 2级（可重复级）向3级（已定义级）过渡的阶段，具体表现如下：

达到CMMI 2级（可重复级）的方面：

基本项目管理到位：制定了项目计划（功能清单、粗略时间线）、进行了任务分解（前端/后端/测试）、使用了版本控制（Git），定期沟通（每日站会）。

需求管理初步实现：记录了核心功能需求（如老人定位、子女告警接收、后台数据管理），并进行了评审。需求变更虽存在但基本有记录。

配置管理基本建立：使用Git进行代码版本控制，有主分支/开发分支概念，主要代码资产得到管理。

质量保证意识萌芽：进行了测试（主要是手动功能测试），修复了发现的缺陷。有部署上线流程。

尚未完全达到CMMI 3级（已定义级）的方面/不足：

缺乏组织级标准过程（OSSP）：项目过程主要依赖团队成员（尤其是我）的个人经验和临场决策，而非组织裁剪的标准流程。例如：代码审查、设计规范、测试策略均未形成团队共识模板。

过程定义与文档化不足：开发流程（如迭代周期定义）、设计规范（UI/API）、详细的测试用例、部署手册等关键过程资产未系统化文档化，知识传递效率低。

依赖核心成员（个人能力）：项目的成功很大程度上依赖核心开发者（我）的技术能力和推动力，过程稳定性和可传承性弱。

度量与分析缺失：未定义和收集关键过程数据（如代码缺陷密度、构建成功率、测试覆盖率、任务完成速率），决策多基于主观判断而非客观数据。

同行评审未制度化：代码审查和设计评审未成为强制且规范的环节，主要依赖个人自觉。

风险管理形式化不足：对技术风险（如Ionic/Electron兼容性）、进度风险识别不够系统，应对计划简单。

亮点（体现向3级迈进）：

采用了相对现代的技术栈（Ionic跨平台、Electron桌面），架构设计合理（前后端分离）。

实现了多端协同开发并最终集成上线，体现了基本的协调能力。

有基本的迭代意识（分模块开发测试）。

总体评估结论：项目能“重复”成功（完成并上线），依赖于基本的管理实践和核心成员能力，但过程尚未“定义”清晰、制度化、文档化，缺乏组织级资产和量化管理，处于CMMI 2级偏上，正向3级努力的阶段。

## 三、基于现有成熟度的过程改进计划

针对上述评估发现的不足，结合CMMI 3级（已定义级）的核心要求，制定以下过程改进计划，目标是在未来1-2个学期内（或下一个类似规模项目），使个人/小型团队的过程成熟度稳定达到CMMI 3级水平。

改进目标：建立清晰定义、文档化且团队共识的软件开发过程，形成可复用的过程资产库，减少对个人英雄主义的依赖。

改进核心领域与具体措施：

定义组织级标准过程（OSSP）与裁剪指南：

行动：参考CMMI 3级实践和业界最佳实践（如Scrum/敏捷精髓），结合学生团队特点（周期短、人员少），撰写一份《小型软件项目开发过程指南（V1.0）》。

内容：明确项目启动、需求分析、设计（概要/详细）、编码规范、代码审查、测试（单元/集成/系统）、构建与部署、结项等阶段的核心活动、入口出口准则、推荐工具（如Jira, GitLab CI, Jest）。提供针对不同项目规模（课程作业/大创/比赛）的裁剪说明。