软件项目管理学期论文

学号：2022141461072 姓名：李芃润

目录

[一、 简述CMMI的层次成熟度模型简述 1](#_Toc31228)

[1.1 初始级（Initial） 1](#_Toc4826)

[1.2 已管理级（Managed） 1](#_Toc16449)

[1.3 已定义级（Defined） 2](#_Toc2428)

[1.4 量化管理级（Quantitatively Managed） 2](#_Toc19752)

[1.5 优化级（Optimizing） 2](#_Toc10494)

[二、 软件过程成熟度评估 2](#_Toc21019)

[2.1 评估目标与维度 2](#_Toc19796)

[2.2 分析项目的CMMI成熟度等级 2](#_Toc12972)

[三、 过程改进计划 3](#_Toc3466)

[3.1 改进目标​ 3](#_Toc27757)

[3.2 现状分析与改进点 3](#_Toc4286)

[3.3 改进策略与措施​ 3](#_Toc29330)

# 简述CMMI的层次成熟度模型简述

CMMI（能力成熟度模型集成）的层次成熟度模型将组织的过程能力划分为5个递进等级，每个等级代表不同的管理成熟度和过程规范性。以下是各等级的核心特征及关键提升点。

1.1 初始级（Initial）

**特征：**

项目开发依赖个人能力，缺乏标准化流程，处于“救火式”状态。

成功具有偶然性，人员变动可能导致项目失败。

无明确的项目管理策略，时间和成本常超支

**提升点：**

建立基本的项目管理框架，明确角色职责。

引入需求跟踪和版本控制工具。

**1.2 已管理级（Managed）**

**特征：**

项目执行有明确计划和监控，可重复历史成功经验。

需求变更受控，过程文档逐步完善。

资源分配和风险应对机制初步形成

**提升点：**

制定标准化项目管理流程（如甘特图、里程碑评审）。

实施基线管理，确保配置项可追溯。

**1.3 已定义级（Defined）**​​

**特征：**

组织级标准过程文档化，跨项目统一应用。

技术活动（如设计、编码）与管理工作（如测试、评审）整合。

过程改进纳入组织文化，通过培训确保全员理解

**提升点：**

建立组织级过程资产库（如模板、案例库）。

推动跨部门协作，消除流程孤岛。

**1.4 量化管理级（Quantitatively Managed）**​​

**特征：**

关键过程指标（如缺陷率、进度偏差）量化监控。

使用统计工具（如控制图）预测和优化过程性能。

质量目标与业务指标直接挂钩

**提升点：**

部署SPC（统计过程控制）工具。

建立数据驱动的决策机制。

**1.5 优化级（Optimizing）**​​

**特征：**

持续改进成为组织核心能力，基于数据分析识别优化点。

主动预防缺陷，流程具备自适应能力。

通过技术创新（如AI辅助开发）提升效率

**提升点：**

实施CMMI高成熟度实践（如过程性能基线）。

构建自动化反馈闭环（如CI/CD流水线）。

# **软件过程成熟度评估**

### **2.1 评估目标与维度**

目标：在线考勤系统

维度：项目管理，需求管理，数据管理

### **2.2 分析项目的CMMI成熟度等级**

**项目成熟度等级：**已管理级（managed）

**分析：**

该项目具备标准化开发流程，有基础的项目管理框架，有明确的角色职责，并且引入了git作为版本控制工具。

该项目有明确的计划和监控，可重复历史成功经验，需求变更也是受控，存在过程文档。有基础的资源分配和风险受控机制。

虽然该项目有标准过程文档，但技术活动和管理工作整合程度较低，开发成员之间时长存在误解。

**总结：**

该项目不存在关键过程指标量化监控，尚未使用统计工具预测和优化过程性能，也没有持续改进或主动预防缺陷，流程比较僵硬，不具备自适应能力。

总结：该项目具备初始级和已管理级的特征，具备部分已定义级的部分，但实际做的并不好，所以最终该项目被定义为已管理级。

# **过程改进计划**

3.1 改进目标​

目标：提升项目过程成熟度至已定义级（Defined），实现以下改进。

建立量化监控体系，覆盖关键过程指标（如缺陷密度、需求变更率）；

优化技术与管理活动整合度，减少团队沟通误解；

引入统计工具预测和优化过程性能，形成持续改进机制。

3.2 现状分析与改进点​

基于CMMI已管理级（Level 2）评估结果，需解决以下问题：

**3.2.1量化监控缺失**​

问题：未定义关键过程指标（KPI），无法量化评估流程有效性。

改进方向：引入敏捷度量指标（如燃尽图、周期时间）和缺陷跟踪看板

​

**3.2.2技术与管理整合不足​**

问题：开发流程与技术活动脱节，需求变更时缺乏自动化工具支持。

改进方向：建立需求跟踪矩阵（RTM），集成CI/CD流水线实现自动化部署

3.3 改进策略与措施​

**3.3.1量化监控体系建设**​

措施：

定义KPI：缺陷密度（每千行代码缺陷数）、需求变更率、测试覆盖率。

工具部署：使用SonarQube监控代码质量，Jenkins生成构建成功率报表

**3.3.2技术与管理流程整合**​

措施：

需求跟踪：在Confluence中建立需求网页，关联GitLab分支和测试用例。

自动化测试：引入Selenium实现核心功能自动化测试，减少人工干预