

四川大學

《软件过程与管理》学期论文



题 目 CMMI 层次成熟度模型及其在

改进项目开发过程中的应用

学 院 软件学院

专 业 软件工程

学生姓名 刘悦玲

学 号 2022141480082

年 级 2022

二〇二五 年 六 月 二十 日

目 录

1 CMMI 层次成熟度模型概述	1
2 考拉在线考试系统软件过程成熟度评估	2
2.1 项目概况与过程管理概述	2
2.2 项目 CMMI 成熟度级别认定	2
3 过程改进计划	4
参考文献	5

1 CMMI 层次成熟度模型概述

CMMI (Capability Maturity Model Integration, 能力成熟度模型集成) 是一套旨在规范软件过程管理、提高软件质量的体系化模型。CMMI 模型分为两种表示法: 连续式(Continuous)和阶段式(Staged), 本文主要讨论阶段式成熟度模型。该模型将组织的过程成熟度划分为五个等级, 每个等级都在前一等级的基础上进一步完善:

1、初始级 (Level 1: Initial)

在初始级别, 组织尚未建立稳定的过程管理机制, 项目实施高度依赖个人能力而非系统方法, 典型表现包括需求混乱、进度失控、超预算与质量不稳定等, 导致项目成败难以预测。

2、可管理级 (Level 2: Managed)

可管理级建立了基本的项目管理过程, 关键过程域覆盖需求管理、项目监控与控制、配置管理等, 并严格按已制定流程执行。虽然系统开发仍依赖于团队成员的个人能力, 但通过对成本、进度和功能交付的跟踪与复用成功经验, 项目绩效开始可重复, 在此基础上为更高级的组织级过程改进奠定了基础。

3、已定义级 (Level 3: Defined)

已定义级代表组织从项目级管理向企业级标准化的重要跃升。在此阶段, 组织不仅完善了可管理级的所有流程, 还建立了企业级过程资产库 (EPG), 为所有项目提供统一的过程描述、模板和最佳实践指南。其覆盖范围扩展至需求开发、技术解决方案、产品集成、验证确认等全生命周期活动, 并系统性纳入风险管理、决策分析等高级实践。

4、可量化管理级 (Level 4: Quantitatively Managed)

在可量化管理级中, 组织基于前期积累的数据, 能够对关键过程属性 (如质量、成本、进度) 建立统计度量与控制机制, 形成过程性能基线 (PPB)。通过定量分析, 项目绩效可在可控范围内得到预测与管理。

5、优化级 (Level 5: Optimizing)

优化级是 CMMI 的最高级别, 表明项目已具备持续的过程改进能力。组织持续收集过程绩效反馈, 利用因果分析、根本原因分析等技术识别改进机会, 并引入创新方法和工具, 实现过程的不断优化与创新。此时, 成熟度模型已形成可复制、可持续演进的闭环体系。

2 考拉在线考试系统软件过程成熟度评估

2.1 项目概况与过程管理概述

在学习完 CMMI 层次成熟度模型后，我将选取**考拉在线考试系统项目**进行软件过程成熟度评估。该项目是我在《软件项目管理》课程中的课程设计项目，我在其中担任**组长（产品经理）**的角色，与九位组员共同完成了该系统的开发。考拉在线考试系统旨在通过信息化手段，优化传统考试与作业管理流程，以提升教育效率和准确性。系统核心功能全面，涵盖教师在线出卷、阅卷、作业发布及学生完成情况追踪；学生则可在线答题，并享有错题本功能。

技术架构方面，系统采用前后端分离模式，前端利用 Vue.js 和 Element-UI，后端基于 Spring Boot 和 MyBatis，数据存储采用 MySQL，并以 Redis 进行缓存加速。同时我们团队分工明确，设有需求分析、前后端开发、数据库管理和测试小组。整个项目开发过程规范有序，从最初的可行性分析、明确需求，到系统设计、代码实现，再到最终的全面测试与验收，各阶段都顺利推进。

2.2 项目 CMMI 成熟度级别认定

通过分析“考拉在线考试系统”的项目文档、代码仓库和开发日志等过程资产，并结合结题时小组成员的心得体会，我系统评估了项目在需求管理、项目规划等关键过程域的实施情况。我认为该项目已建立了基本的项目管理过程，其管理实践特征与**可管理级（Level 2）**高度吻合。

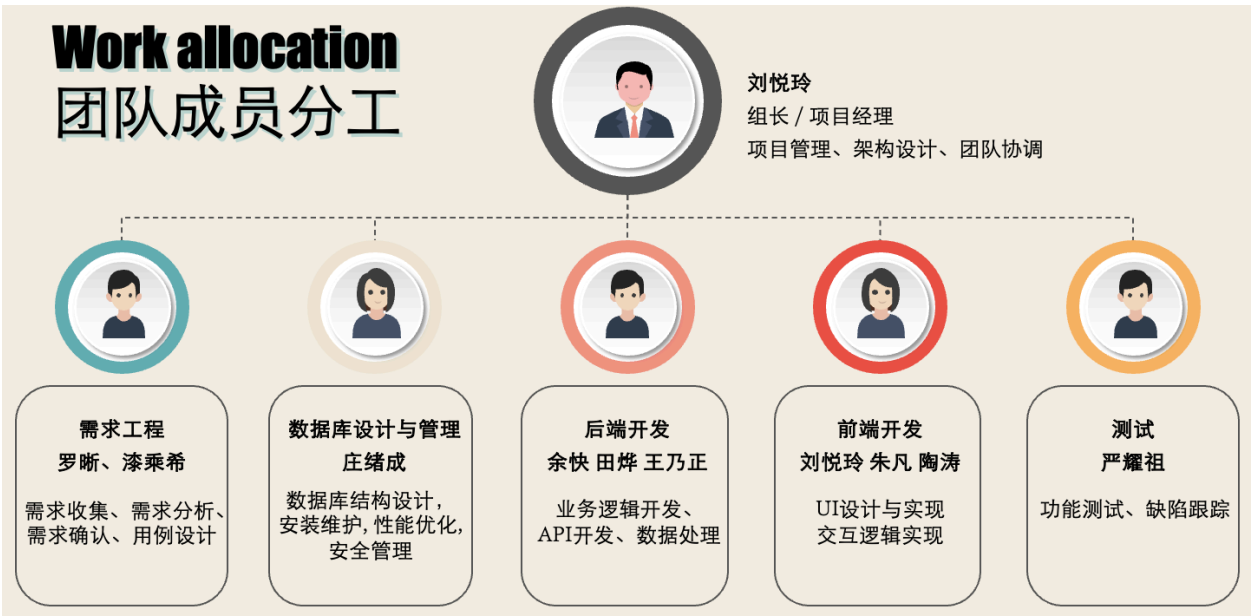


图 2.2.1 项目成员分工图

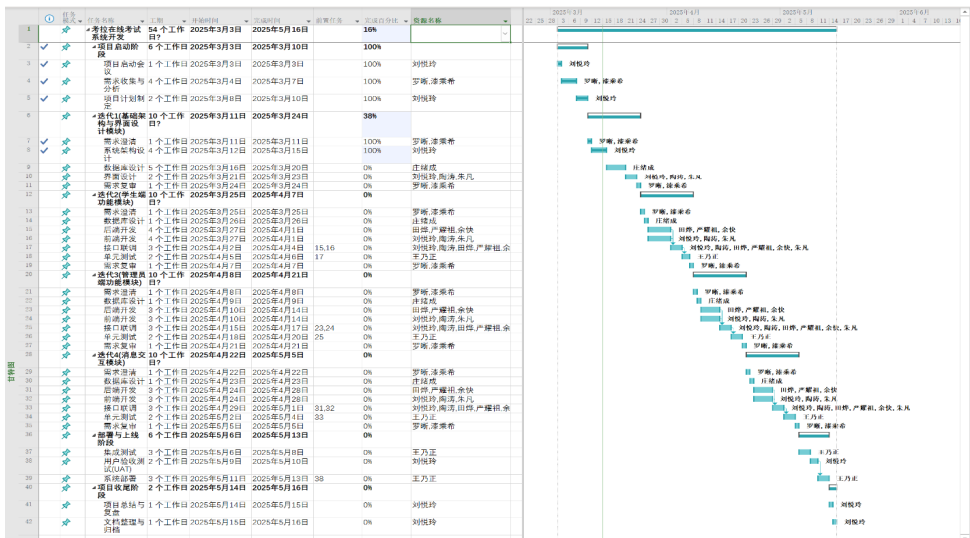
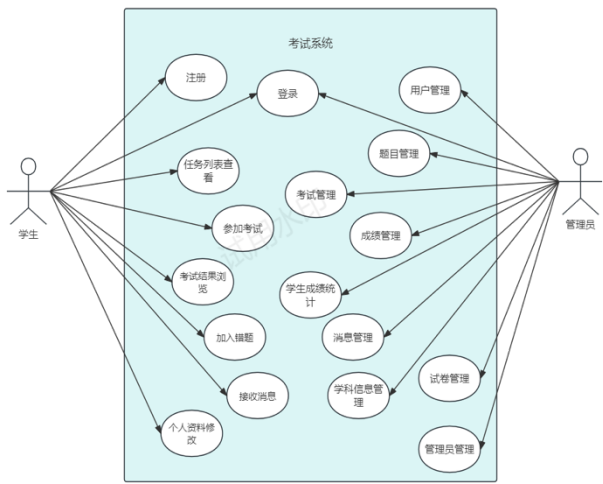
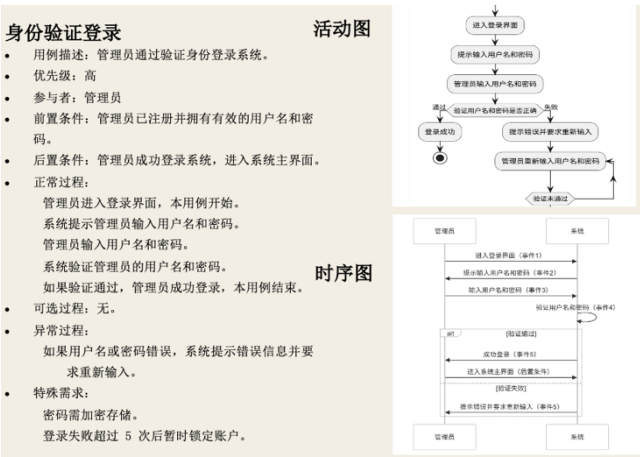


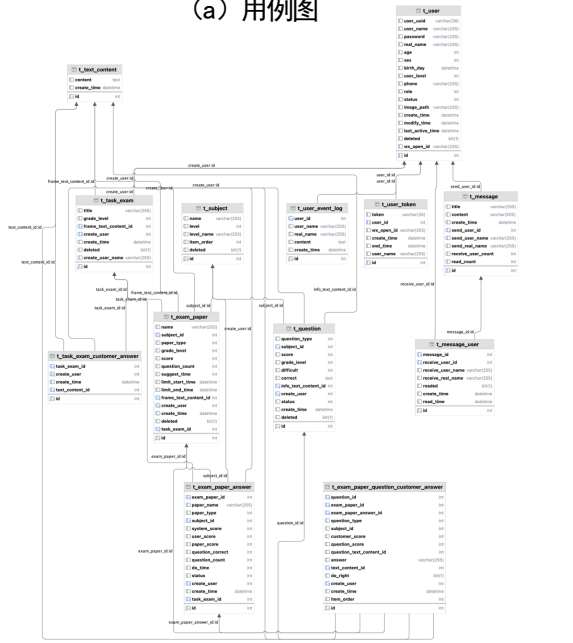
图 2.2.2 项目计划甘特图



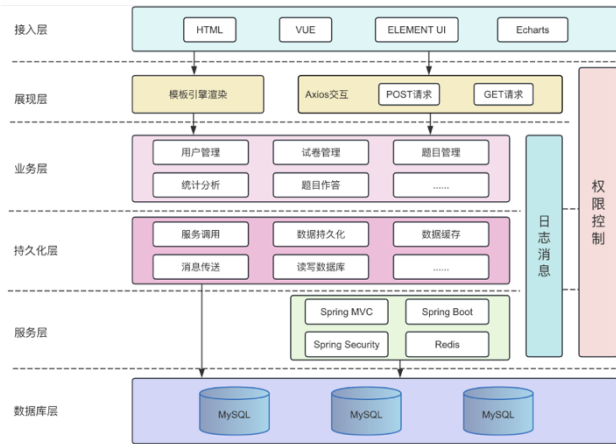
(a) 用例图



(b) 用例设计示例



(c) ER 图



(d) 系统架构图

图 2.2.3 项目开发过程关键图展示

该项目成功交付了一款功能完备的产品，在项目初期我们制定了详细的项目计划，合理任务分配（图 2.2.1）并进行甘特图跟踪（图 2.2.2），整个过程中有效把控项目进度。在需求分析阶段，我们不仅明确了在线组卷、考试、阅卷等核心功能用例（图 2.2.3-a），还通过 E-R 图（图 2.2.3-c）设计了数据实体，并为功能用例编写了用户故事、用例描述，绘制了大量活动图和时序图（图 2.2.3-b），确保了需求的全面理解与文档化。系统设计涵盖概要与详细设计，其中概要设计规划了功能模块、详细的数据库模式图 and 核心数据表、分层架构（图 2.2.3-d）与系统部署方案，并设计了直观的界面布局和原型图；详细设计则通过类图、活动图和时序图精确指导了编码。

错误类型	错误代码	错误描述	系统输出信息	处理能力
数据库连接失败	5001	无法建立数据库连接或连接超时。	“系统繁忙，请稍后重试。”	自动重试 3 次，失败后触发告警通知运维人员，并切换至备份数据库。
用户输入验证错误	4001	用户名/密码格式不符合要求。	“用户名需为 6-20 位字母数字组合。”	前端实时校验，阻止非法提交，并高亮错误字段。
并发访问超限	5031	同时在线用户数超过系统承载能力。	“当前系统繁忙，请稍后再试。”	触发限流策略（令牌桶算法），队列化请求，并建议用户错峰访问。
文件上传失败	4003	上传文件大小超过限制（如 100MB）。	“文件大小不能超过 100MB。”	前端拦截大文件，后端记录日志并返回错误详情。
身份认证失败	4011	Token 过期或无效。	“登录已过期，请重新登录。”	强制跳转至登录页，清理本地缓存，并记录安全日志。
第三方接口超时	5041	调用外部服务（如短信网关）超时。	“服务暂不可用，请稍后重试。”	异步重试机制（最多 3 次），失败后降级处理（如本地缓存短信队列）。
系统内部异常	5000	未捕获的运行时异常（如空指针）。	“系统内部错误，请联系管理员。”	记录详细错误堆栈至日志系统（ELK），触发告警，并展示友好错误页。
磁盘空间不足	5002	服务器存储空间使用率超过 90%。	（后台告警，用户无感知）	自动清理临时文件，通知运维扩容存储，并暂停非核心服务（如日志归档）。

(a) 系统出错处理设计

测试用例	测试步骤	预期结果	实际测试
登录注册模块	1. 登录和注册时输入不符合类型的数据 2. 注册时检查数据库中是否添加了相应的用户数据	1. 做出相应检查并提示 2. 数据库中添加了相应的用户数据	通过
任务中心	1. 任务完成时任务标签是否变为已完成 2. 测试查看详细是否会显示详细的任务信息	1. 任务标签从未完成变为已完成 2. 显示详细的任务信息	通过
考试功能	1. 学生提交试卷后是否正常保存在数据库 2. 限时考试是否必须在规定时间内答题 3. 查看考试详情时是否能显示正确的内容	1. 试卷正常保存在数据库 2. 必须在规定时间内答题 3. 显示正确的考试内容	通过
考试记录浏览功能	1. 测试学生是否可以查看自己的考试记录 2. 测试在考试记录中心界面，学生是否可以查看自己已结束的考试 3. 测试学生是否可以查看一项考试，查看考试的详细信息	1. 学生可以查看自己的考试记录 2. 可以选择查看所有已结束的考试 3. 可以查看考试的详细信息	通过
知识库功能	1. 测试做错题目能否被收入错题库 2. 测试学生是否可以在错题知识库中查找和学习相关错题	1. 测试做错题目能否被收入错题库 2. 学生可以查找和学习错题	通过
错题本功能	1. 测试学生考试答错的题目是否会自动的入错题本内 2. 错题本是否会记录题目的题干，学科，用户作答情况，并给出正确答案和解析	1. 答错题目自动纳入错题本 2. 记录题目信息并给出正确答案和解析	通过
个人资料功能	1. 学生是否可以查看自己的个人信息 2. 学生是否可以点击编辑，进行修改和保存操作，并反馈到数据库中	1. 可以查看个人信息 2. 可以编辑并保存，反馈到数据库	通过
接收消息功能	1. 学生是否可以在消息中心接收来自老师和同学的消息 2. 在消息中心界面，学生是否可以查看所有的消息 3. 学生是否可以查看一项显示未读的消息，查看消息	1. 可以在消息中心接收来自老师和同学的消息 2. 在消息中心界面，学生可以查看所有的消息 3. 学生可以选择一项显示未读的消息，查看消息	通过

(b) 测试设计与结果

图 2.2.4 项目开发过程关键设计展示

系统实现过程中我们严格遵循前述设计文档进行开发。最后，我们在系统测试阶段配置了完整的软硬件环境，并设计了管理员端和学生端的详细测试用例（图 2.2.4），通过功能测试和数据库测试确保了系统质量。这些实践均体现了考拉考试管理系统项目层面过程的规划、执行、监控和控制能力。然而，若要让这种成功经验在更多项目上可复制，就需要将项目中的良好实践正式化、制度化，提升到组织层面。故最终评估结果是，“考拉在线考试系统”项目在项目层面过程管理方面已满足 CMMI 可管理级（Level 2）的要求，但尚未全面达到已定义级（Level 3）。

表 2.2 综合展示了考拉考试系统项目在各关键过程域的实践情况、已达到管理级（Level 2）的证据以及与已定义级（Level 3）存在的差距，以此作为本次成熟度级别认定的直观依据：

表 2.2 考拉在线考试系统项目 CMMI 成熟度具体评估

CMMI 过程域	项目过程成熟度达到 Level 2 的证据	与 Level 3 存在的差距
需求管理 (REQM)	项目为 在线组卷、错题本 等核心功能编写了详细的用户故事和用例描述，并绘制了 48 个活动图和时序图 直观展现系统流程和对象交互；明确定义了 系统响应时间、并发用户数 等性能，以及遵循 OWASP 标准的安全性需求。通过需求小组和项目经理举行需求确认会议确保理解一致。	缺乏正式的需求变更管理流程和双向可追溯机制。
项目规划 (PP)	项目经理 基于软件开发生命周期 制定了详细的项目进度表，分解任务并分配给 前后端开发、测试 等小组；利用甘特图 持续跟踪项目各阶段进度 ；识别了 技术选型风险、开发周期风险 等。规划明确且具可控性。	未明确利用 组织过程资产 进行规划，也未贡献于组织资产。
项目监控与控制 (PMC)	项目经理 每周定期召开例会 跟进 前后端开发、测试 小组的进度；期间对接口定义不明确、数据格式不匹配等风险进行及时沟通和干预，有效避免了返工。监控与控制有效。	侧重被动纠正而非主动预防；缺乏 系统性根本原因分析机制 。
配置管理 (CM)	当前主要依靠人工管理和不完全的 Git 仓库进行 版本控制 ，；团队成员为解决协作困难 自行编写详细 README 文档 。	缺乏统一的 配置管理制度 ，人工效率低且易出错；团队成员自行编写 README 缺乏规范性，难以复用。
过程与产品质量保证 (PPQA)	项目进行了 全面细致的功能测试和数据库测试 ，为管理员端（如用户管理、试卷管理）和学生端（如在线答题、错题本）设计了详细测试用例；项目经理执行制定验收标准。	项目经理 一人承担大部分审查 ，存在视角局限，缺乏正式、独立、多角色参与的评审流程。
测量与分析 (MA)	项目定义了 系统响应时间、并发用户数、吞吐量 等 非功能性性能需求 ；管理端主页动态仪表板实时展示试卷总数、用户活跃度趋势等 运营指标 。	缺乏对 开发过程本身 （如缺陷密度、返工率）的定量测量和分析。
供应商协议管理 (SAM)	项目主要基于 内部团队开发	内部协作未建立规范化标准 （如存在代码风格差异问题）。

3 过程改进计划

本改进计划的核心目标是把“考拉在线考试系统”的开发流程从 CMMI 可管理级（Level 2）提升到已定义级（Level 3），这意味着我们需要项目层面的成功实践和过程中的经验正式化为组织范围内的标准过程，并建立明确的裁剪和管理这些过程的指导方针。

为实现这一目标，我计划采取以下改进策略：标准化现有的实践经验为组织级过程资产；制度化这些已定义过程在组织内的执行；形式化过程相关的方案、角色、职责和工具；并通过系统培训组员提升，确保所有团队成员理解并遵循新的过程管理模式。具体来说，“考拉在线考试系统”软件过程成熟度的改进措施如下：

1、将项目实践固化为组织级流程。把“考拉”项目里最有价值的过程成果都整理成统一的文档和模板库，例如需求分析模板（用例和用户故事）、活动图与时序图、数据库模式图、前后端分层架构设计、以及组件化开发经验（如可复用的倒计时和自动保存逻辑）等。同时，为了杜绝目前“手动管理 + 零散 Git 仓库”带来的版本混乱问题，我计划上线一套文档管理平台（例如配合 CI/CD 的 GitLab/GitHub Enterprise），让所有图表、设计稿、测试用例都能自动归档、打标签、精确回溯。这样，新项目只需从这个“知识仓库”里拿模板、填内容，就能快速启动并保持高一致性。

2、引入多维度评审与风险决策。除了项目经理把关外，项目要成立一个小型评审小组，成员由架构师、测试负责人、业务代表等组成，定期对关键模块（在线组卷、评分逻辑、核心用例实现）进行交叉审查，并把审查结果和改进建议记录在案。同时，为了应对高并发、数据隐私和接口兼容等风险，我计划制定简单明了的风险登记表，给每个风险打上“可能性”和“影响度”标签，并附上缓解和应急措施。对于重大技术抉择（如框架升级或安全加固），要以标准化的决策模板汇总备选方案、利弊分析和最终结论，确保全体成员都能够看得懂、查得清。

3、用数据说话，持续优化。建立过程管理仪表盘，实时展示以下几个关键指标：需求与设计阶段的缺陷数、代码评审发现的问题、主要功能的返工工时、测试用例通过率，以及进度偏差率。在每个里程碑完成后，团队会开个短会（30 分钟即可），对照仪表盘数据，做“5 个为什么”根因分析，找出阻碍效率的瓶颈并制定下一步改进措施。改进方案也要同步写入我们的模板库，让好的做法和经验归档保存下来，以便日后学习查看。我相信通过这样“度量—复盘—改进—再度量”的闭环，该项目过程管理会越来越成熟，为未来达到可量化管理（Level 4）作准备。

参考文献

- [1] CMMI Institute. CMMI V2.0 Overview [S]. Pittsburgh, PA: CMMI Institute, 2018.
- [2] CMMI (Capability Maturity Model Integration, 能力成熟度模型集成) 概述.
https://blog.csdn.net/weixin_43510208/article/details/148558636, 2022-04-15.