一、CMMI的层次成熟度模型：

CMMI（Capability Maturity Model Integration）的层次成熟度模型将组织流程能力划分为 5 个渐进式层级，每个层级代表不同的管理水平和流程成熟度。以下是对各层级的详细阐述：

**1. 初始级（Level 1：Initial）**

初始级组织的流程处于无序状态，项目执行依赖个人经验而非系统化方法。工作方式具有高度临时性，常因突发问题（如需求变更、资源不足）导致计划频繁调整。项目成功主要取决于个人能力和运气，而非组织能力。例如，团队可能在未充分规划的情况下启动项目，缺乏标准化的开发流程和质量控制机制，导致进度延误或产品缺陷率高。此阶段的组织难以重复成功经验，过程改进缺乏基础。

**2. 已管理级（Level 2：Managed）**

已管理级组织建立了基本的项目管理流程，能够对单个项目的成本、进度和质量进行跟踪和控制。核心实践包括需求管理、项目计划制定、配置管理和质量保证等。例如，团队会明确项目目标和范围，制定详细的进度计划，并定期监控实际进展与计划的偏差。每个项目独立实施流程，虽能保证基本可控性，但流程可能因项目而异，缺乏组织层面的统一标准。此阶段为后续的流程标准化奠定基础。

**3. 已定义级（Level 3：Defined）**

已定义级组织将项目管理流程标准化和文档化，形成组织级的标准流程体系。所有项目基于此标准流程进行裁剪和实施，确保方法一致性。例如，组织会建立统一的软件开发流程、测试规范和培训体系，要求团队严格遵循。跨部门协作得到加强，过程文档（如需求规格说明书、设计文档）完整且可追溯。此阶段通过标准化提升效率，减少重复工作，并为组织积累知识资产。流程改进由个别项目扩展到组织层面。

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

量化管理级组织引入数据驱动的管理方法，通过量化指标（如缺陷密度、生产率）对流程进行精确控制和预测。组织定义关键绩效指标（KPI），持续收集过程数据，并使用统计方法分析流程稳定性。例如，通过控制图监控缺陷率波动，预测潜在质量风险；基于历史数据优化资源分配。此阶段强调预防而非纠正问题，流程性能可被精确测量和预测。数据驱动决策使组织能够识别微小偏差并及时调整，提升过程可靠性。

**5. 优化管理级（Level 5：Optimizing）**

优化管理级组织以持续改进为核心目标，主动寻求流程创新和技术突破。通过收集过程数据识别系统性弱点，开展试点项目验证改进方案，并将成功经验推广到全组织。例如，引入自动化测试工具提升效率，或采用敏捷开发方法缩短交付周期。缺陷预防机制成为常态，组织从 “解决问题” 转向 “预防问题”。此阶段注重培养创新文化，鼓励员工提出改进建议，并通过技术创新实现流程跨越式提升。组织能力持续演进，以适应市场变化和业务需求。

二、项目过往软件过程成熟度评估

以《软件项目管理》课程项目《川大贴吧》为例，对其软件过程成熟度进行评估。

需求管理：需求获取阶段，团队通过访谈、问卷等方式收集川大师生对贴吧功能的期望，形成《初始需求列表》（如帖子发布、分类检索、用户认证等）。需求分析与定义阶段，对需求进行初步分析，明确功能边界，编写《需求规格说明书》，经导师或客户（如川大信息化处）评审确认。需求变更控制部分，制定需求变更流程，当提出新需求（如增加 “校友圈” 功能）时，需填写《需求变更申请表》，经团队讨论和听取导师意见后纳入项目范围。项目需求相对明确。

项目计划制定：基于需求规格说明书，明确项目功能范围（如用户管理、帖子管理、评论系统）和非功能需求（如响应时间≤2 秒）。进度计划：使用甘特图制定详细进度计划，明确各阶段里程碑（如需求完成、原型交付、测试完成），并分配资源和时间节点（例如：需求分析 2 周、设计 1 周、开发 7 周、测试 3 周）。并且每周召开项目例会，对比实际进度与计划偏差，必要时调整计划（如因技术难点延长开发周期 1 周）。

配置管理：在《川大贴吧》项目中，我们采用了一套完整的工具链来支撑软件过程管理，确保项目达到 CMMI 已管理级标准。具体而言，通过 Maven 进行统一的依赖库管理，在pom.xml中精确定义所有依赖项的版本号（如 Spring Boot 2.7.5、MyBatis 3.5.7），并利用依赖传递机制避免版本冲突；使用 Git 进行版本控制，创建主分支（master）用于生产发布，开发分支（develop）集成每日迭代，按功能模块（如 user-service、post-module）创建特性分支；引入 PingCode 进行测试管理，测试团队在 PingCode 中创建测试计划、设计测试用例，执行过程中实时记录测试结果与缺陷信息（如缺陷编号 PC-123 对应 “用户注册验证码失效”），并自动生成测试覆盖率报告；借助华为云 CodeArts 平台进行项目需求管理，将《需求规格说明书》导入平台生成需求条目，关联用户故事与任务，通过看板视图跟踪需求实现状态，支持需求变更的审批与影响分析。这套工具链的整合应用，使项目团队能够在需求、开发、测试等环节实现无缝协作，确保过程数据可追溯、进度可控，为项目的成功交付提供了坚实保障。

**综上所述，该项目已达到已管理级（Level 2：Managed）**

三、修改计划

基于《川大贴吧》项目当前已达到 CMMI 已管理级（Level 2）的评估结果，为进一步提升至已定义级（Level 3），需对现有过程进行标准化、制度化改进。以下是具体的改进计划：

1. 组织级标准流程体系建设

目标：

建立覆盖需求、设计、开发、测试、维护全生命周期的标准化流程

改进措施：

流程文档化：

制定《软件开发流程规范》，统一采用敏捷开发框架（如 Scrum），明确各阶段输入 / 输出（如需求阶段需产出《用户故事地图》《验收标准》）。开发《模板库》，包含需求规格说明书、设计文档、测试用例模板、项目总结报告等标准模板

跨项目流程统一：

成立过程改进小组（EPG），负责流程制定与推广，确保《川大贴吧》项目的最佳实践（如 Git Flow、Maven 依赖管理）应用于所有后续项目。建立《流程裁剪指南》，允许项目根据规模和复杂度在标准流程基础上进行合理裁剪

2. 需求管理深化

目标：

实现需求全生命周期跟踪与双向追溯

改进措施：

需求建模与分析：

引入用例图、用户故事地图进行需求可视化，使用工具（如 JIRA 或华为云 CodeArts）建立需求条目间的父子 / 依赖关系增加需求优先级评估（如 MoSCoW 方法），确保资源优先投入核心功能需求变更影响分析：在《需求变更申请表》中强制要求填写变更影响范围（如涉及的模块、测试用例、文档更新）

3. 配置管理强化

目标：

实现更精细的版本控制与基线管理

改进措施：

构建自动化：配置 Jenkins 流水线，实现代码提交后自动编译、单元测试、代码质量检查（如 SonarQube 扫描）建立《构建规范》，确保所有环境（开发、测试、生产）依赖一致发布管理标准化：制定《发布流程指南》，明确版本号规则（如 SemVer）、发布前检查清单（如回归测试覆盖率）