**一、CMMI层次成熟度模型简述**

CMM是一种评估组织软件开发过程成熟度的框架，由美国卡内基梅隆大学软件工程研究所提出的，目的是通过标准化流程提升软件质量与项目管理效率。这个模型将成熟度分为五个层级，每个层级代表过程可控性和可预测性的提升。

**1级-初始级**

过程无序且依赖极度个人能力能力，项目经常失控。次层级的成功往往源于“个人英雄主义”，难以复现。存在需求变更频繁、无文档化规范等问题。

**2级-可重复级**

拥有建立基础项目管理的能力，如需求跟踪、进度监控和风险应对等能力。项目基于历史经验制定计划，能复现过往成功项目。典型实践包括项目计划文档化、代码版本控制等。

**3级-已定义级**

组织级标准化流程形成，工程活动与管理活动集成。设立SEPG（软件工程过程组）制定统一规范，项目可裁剪使用标准流程。例如，设计评审制度化、测试用例库共享。

**4级-量化管理级**

通过数据指标预测过程与产品质量。采用统计工具实现异常控制。目标例如千行代码缺陷率≤0.5等指标进行测评。

**5级优化级**

项目可以持续改进与技术创新。通过根因分析及预防缺陷，通过引入自动化工具来提升效率。

**GDP数据可视化项目的成熟度评估**

以​“GDP数据可视化”项目为例（技术栈：vue3 + ECharts + d3，数据源：国家统计局）：

**1. ​成熟度定位：Level 2**

**符合特征**​：

需求管理：明确用户需求，编写明确的项目需求文档，并将需求拆分为任务

配置管理：代码通过Git版本控制，数据清洗脚本复用历史项目。

**未达标特征**​：

项目计划：未使用甘特图分配开发、测试时间，仅仅进行了大体粗略的项目计划设计和时间管理。

缺乏组织级标准​：未定义统一的数据处理流程，导致不同成员清洗逻辑不一致。

无量化管理​：未统计代码缺陷率或模块开发效率，延期风险靠经验判断

**2. ​关键问题分析​**

|  |  |  |
| --- | --- | --- |
| ​CMMI过程域​ | ​项目表现​ | ​成熟度差距​ |
| 需求管理 | 用户需求文档化，进行了需求收集及分析并将其文档化产生需求文档 | Level 2 达标 |
| 项目计划 | 制定了粗略的时间计划但未量化资源投入，未进行规范化时间流程管理 | Level 2 部分达标 |
| 过程与产品质量 | 只进行了简单的人工测试，无自动化覆盖率统计吗，没有进行规范化的单元测试 | Level 3 未达标 |
| 组织过程定义 | 无标准化数据清洗流程，只是每位成员对于自己所需要的数据单独进行清洗，并且部分数据是在程序运行过程中才进行清洗的 | Level 3 核心缺失 |

**过程改进计划**

**改进措施**​：

**从项目计划方面改进**

选择专业项目管理工具，通过甘特图实现。并且将任务拆解与依赖关系​将开发、测试任务细化为子任务进行“前端模块开发→单元测试→集成测试”的流程，明确前后依赖，测试必须在开发完成后启动。为每项任务设定精确起止时间，标注关键路径。

**从组织过程方面改进**

应当制定《数据清洗流程标准》，包含缺失值处理、单位统一等步骤，所有清洗操作需记录四要素日志（操作人、时间戳、原始值、修正值）实现可追溯性

**从过程与产品质量方面改进**

应建立规范化的单元测试体系，遵循AIR原则和BCDE设计准则，确保核心模块的语句覆盖率和分支覆盖率均达到100%。测试环境需独立部署，并整合工具链。重点覆盖接口参数校验、局部数据结构和异常处理逻辑，每个函数至少包含1个正常用例和1个异常用例，同时检查参数匹配、全局变量一致性和边界条件等关键项。

其次，构建自动化覆盖率统计体系，从代码级、接口级到功能级多维度监控。通过工具链生成可视化报告，标注未覆盖代码行，并集成到CI流水线，设置覆盖率阈值（如行覆盖率≥80%、分支覆盖率≥70%）以阻塞低质量代码提交。