

《软件过程与CMMI》课程报告

——基于CMMI的失败案例分析

学生姓名： 周麟

学 号： 20171001561

班 号： 111181

指导教师： 周顺平

成 绩：

中国地质大学信息工程学院

2021.5

要求：选取世界范围内失败的软件项目，并基于CMMI2.0版第2级过程域的实践要求，分析项目失败的真正原因。

目录

[1 项目概况 2](#_Toc5694)

[1.1 项目基本信息 2](#_Toc2998)

[1.2 项目实施过程简介 2](#_Toc24561)

[1.2.1组织架构 2](#_Toc29190)

[1.2.2项目主要成员及组织架构 2](#_Toc18791)

[1.2.3项目开发生命周期 3](#_Toc11414)

[1.2.4项目实施过程 3](#_Toc20607)

[2 基于CMMI 2.0 L2的失败原因分析 4](#_Toc8212)

[2.1 实践活动强弱项分析 4](#_Toc21588)

[2.2 项目失败原因分析总结 12](#_Toc4040)

[2.2.1需求变更频繁 12](#_Toc20801)

[2.2.2设计、评审不足 13](#_Toc31756)

[2.2.3 验证任务分配错误 13](#_Toc3574)

[2.2.4 配置管理缺失 13](#_Toc16776)

[3 参考文献 13](#_Toc19024)

# 项目概况

## 项目基本信息

项目名称：集资讯、理财、服务为一体的服务网上交易客户的辅助工具软件系统

项目承担者：YD公司的移动互联网事业部（即M事业部）

项目实施周期：成立于2009年-至今

## 项目实施过程简介

1.2.1组织架构

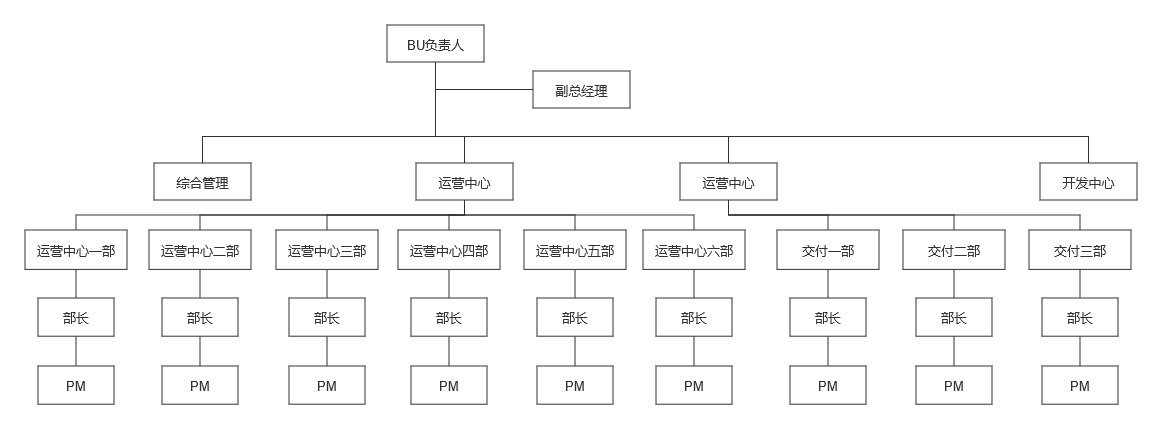


图 1M事业部组织架构

1.2.2项目主要成员及组织架构

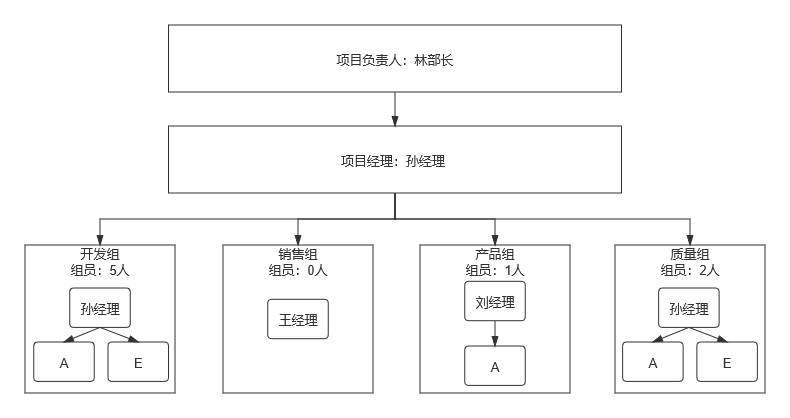


图 2项目组织架构图

1.2.3项目开发生命周期

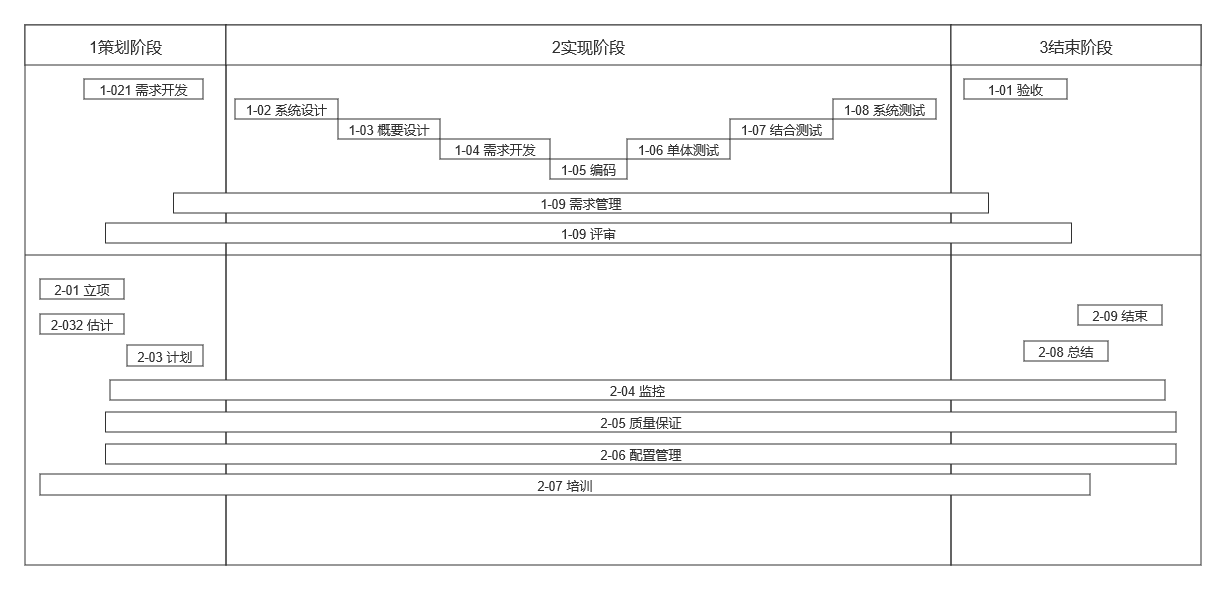


图 3YD公司标准软件开发生命周期模型

**1.2.4项目实施过程**

1.2.4.1需求定义过程

* 项目进行初期竞标时上层领导都十分重视，都分别前去。竞标完成之后，在敲定的需求的需求基础上，给用户提供了软件的UI/UE展示效果图。
* 随后立刻赶回公司召开项目启动会议，首先对项目需求进行培训，再根据成员能力对项目模块进行分工。
* 随着项目的推进，逐渐暴露出问题，问题就是客户经常提出一些未经推敲的想法，而这些方法没有整体的规划，也没有正规的流程可以遵守，但是客户还是不断地提出修改意见，不断推翻之前经过再三讨论而确认定下来的需求。每次需求变更，都需要进行新一轮的需求分析，甚至无休止的讨论，严重时甚至要重新设计系统架构。
* 但是部长认为“客户就是上帝，客户满意度就是终极目标”，必须要直面客户频繁提出的需求更改，于是团队员工加班加点，项目磕磕绊绊结束了。

1.2.4.2技术实行过程

* 项目经理根据需求划分了几个功能模块，并且分配给项目成员。
* 并且委派技术部长制定了适合本项目的编码规范，但是由于部分编码经验少的项目成员在编码前期没有对其负责的模块整体进行合理规划。部分项目成员不了解面向对象语言的真实含义，产生了大量冗余代码，降低其负责模块的执行效率；再者，每个人的编码风格在短时间之内很难改变过来。

1.2.4.3验证过程

* 项目经理、客户代表对项目执行全过程进行监督和跟踪。
* 开发人员本身存在严重的问题，及对业务知识缺乏了解，对业务逻辑的理解与实际存在偏差，因此导致系统产生了几个严重的问题。
* 测试人员对代码逻辑和实现不熟悉，无法进行专业的白盒测试，于是委托技术人员进行测试。
* 项目经理对验证过程制定了一系列安排。
* 测试期间产生了一系列问题，包括运行问题、测试覆盖率低的问题。

1.2.4.4配置管理过程

* 项目使用SVN进行文件即成果物的管理，同时安排其他项目人员对大家进行了简单的SVN的使用培训，使用初期问题不断。
* 针对问题，项目经理请教了几个项目组负责人对此问题的处理方法，提出相应的解决方案。

# 基于CMMI 2.0 L2的失败原因分析

## 实践活动强弱项分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实践域** | **实践活动要求** | **是否达到要求？** | **强项** | **弱项** |
| 需求开发和管理 | RDM 2.1 抽取利益相关方的需求、期望、约束条件、接口  或连接。 | √ | 在新项目中，与客户敲定最终需求之后，项目负责人和项目经理连续两天加班到凌晨，不断与客户沟通，探讨，最终确定软件需求及UI/UE的展示效果图 | × |
| RDM 2.2 将利益相关方的需求、期望、约束条件、接口或  连接转化为优先的客户需求。 | √ | 宣布项目开始之后，立刻对项目需求进行培训，根据 成员能力对项目模块进行分工 | × |
| RDM 2.3 与需求提供者就需求的含义达成一致。 | √ | 每次需求变更都会进行新的一轮需求分析 | 客户经常提出一些未经推敲的想法，而这些想法，没有整体的规划，也没有正规的流程可以遵循，但客户还不断的提出修改意见，不断的推翻之前经过再三讨论而确定下来的需求 |
| RDM 2.4 获得项目参与者的承诺，即他们可以落实这些需求。 | √ | 始终保持着“客户就是上帝，客户满意度是终极目标”，开发团队保证要时刻做好应对客户新需求的准备 | 无法对客户提供的不完整、不详细的文档采取有效的措施，计划会不断的调整，连交付时间都成为一个不确定因素 |
| RDM 2.5 开发、记录和维护需求和活动或工作产品之间的双向可追溯性。 | × | × | 两期项目完全没有可参考性，无法总结经验，也不可追溯 |
| RDM 2.6 确保计划、活动或工作产品与需求保持一致。 | √ | 为了及时应对客户频繁提出的不详细，不完整的需求变更，团队技术人员加班追赶工期，按期完成任务 | 长时间加班加点的高负载的工作量会让团队成员疲惫不堪、充满悲观情绪；并且没有制定高效率的开发流程，会让这种情况继续下去 |
| 估算 | EST 2.1 制定、使用并保持更新估算范围。 | √ | 对于每次的需求改变都会进行新一轮的需求分析 | 花费大量时间、人力和物力 |
| EST 2.2 制定并保持更新针对解决方案规模的估算。 | × | 本公司的资金、资源都十分充裕 | 没有进行估算，而是直接进行开发，全面运用一切有效方法 |
| EST 2.3 根据规模估算来制定并记录解决方案所需工作量、周期和成本及其依据。 | × | 本公司的资金、资源都十分充裕 | 为了努力达成用户的需求，团队承受巨大的工程量和精神压力 |
| 计划 | PLAN 2.1 开发完成工作的方法并保持更新。 | × | 项目经理、项目负责人的经验丰富，能够应对绝大多数情况 | 前期的大量需求变更无法跟踪，而导致前后期的经验无法对照，无法制定有效率的方法 |
| PLAN 2.2 计划执行工作所需的知识和技能。 | √ | 在项目实施的每一个阶段初期，都会对项目成员进行培训 | 但是由于某些技术人员的专业能力、编码习惯等不深入而导致开发进程效率低下 |
| PLAN 2.3 根据记录的估算，制定预算和进度并保持更新。 | √ | 对于用户不断进行的需求变更，都能积极应对并且当即制定解决方案 | 没有高效率的执行流程，使得每次耗费巨大的人力物力 |
| PLAN 2.4 计划已识别的干系人的参与。 | √ | 项目经理经验丰富，能够考虑多种因素包括已识别的干系人 | × |
| PLAN 2.5 计划向运营和支持的转移。 | × | × | × |
| PLAN 2.6 通过协调可用资源和估算的资源确保计划的可行性。 | × | 项目经理、团队成员的开发技术强大 | 为了达成客户提出的不计成本，给项目成员增加了巨大的工作成本；与此同时，客户不断增加需求 |
| PLAN 2.7 制定项目计划，确保其元素之间的一致性，并保持更新。 | × | 对于用户频繁提出的需求，都能够及时的完成 | 达成的代价就是团队成员加班加点完成的 |
| PLAN 2.8 评审计划并获得受影响的干系人的承诺。 | × | 对整体框架进行评审之后，各模块担当，找到对应接口，往接口处追加功能 | × |
| 监视与控制 | MC 2.1 从规模、工作量、进度、资源、知识和技能以及预算等方面，对比估算跟踪实际结果。 | × | 企业规模、资源、项目员工的专业技术都十分到位 | 没有对估算与实际结果进行对比分析 |
| MC 2.2 跟踪已识别的利益相关方参与和承诺情况。 | √ | 全程记录用户变更的需求 | 时间、资源、人力等成本投入巨大 |
| MC 2.3 监控向运营和支持的迁移。 | × | × | 没有运行和支持的迁移 |
| MC 2.4 当实际结果相较于计划存在显著差异时，采取纠正措施并管理直至关闭。 | √ | 能够对于存在的问题当即制定相应的规则进行规范行为 | 规定实行初期，效率低下 |
| 过程质量保证 | PQA 2.1 根据历史质量数据开发、持续更新并遵循质量保证方法和计划。 | √ | 借助于CMMI和ISO9001质量和流程标准以及完善的软件开发方法论 | × |
| PQA 2.2 在整个项目过程中，根据记录的过程和适用标准客观评价选定的已执行过程和工作产品。 | √ | 能够根据记录的过程和适用标准客观评价选定的已执行过程和工作产品，给客户提供全过程的应用开发及维护项目的服务 | × |
| PQA 2.3 沟通质量和不合规问题并确保解决问题。 | √ | 能够对每一轮的新需求进行分析 | 无休止的讨论，无法达成一致，严重的话甚至会重新设计现有的架构 |
| PQA 2.4 记录并使用质量保证活动的结果。 | √ | 对软件产品质量从项目初始到最终成果物的交付的整个过程进行实时监控的流程规范，从而提升软件产品质量、提交客户完整可用的产品、降低企业运营成本等 | × |
| 配置管理 | CM 2.1 识别将置于配置管理下的配置项。 | √ | 项目经理综合各种因素考虑，决定使用SVN对文件及成果物进行管理 | × |
| CM 2.2 开发、使用并保持更新配置和变更管理系统。 | √ | 在开发整个阶段，始终使用SVN对项目文件及其成果物进行管理 | 在项目使用初期团队成员对SVN操作出现了一些问题：误删除；对自己编辑过的文件没有及时迁入；将别人修改的版本进行覆盖；无法提交文件等问题 |
| CM 2.3 开发或发布供内部使用或交付给客户的基线。 | × | × | 没有进行客户基线的定义 |
| CM 2.4 管理对配置管理下的项的变更。 | √ | 在开发整个阶段，始终使用SVN对项目文件及其成果物进行管理 | 初期由于团队成员对配置工具不熟悉而产生了一些问题 |
| CM 2.5 开发、使用并保持更新描述配置管理下的项的记录。 | × | 在开发整个阶段，始终使用SVN对项目文件及其成果物进行管理 | 前期配置管理的问题导致衍生问题不断，在回滚到历史版本时，无法追溯稳定版本的历史记录，因此不得不从一个未知的状态开始开始修改BUG |
| CM 2.6 执行配置审计以保持配置基线、变更和配置管理系统内容的完整性。 | × | 制定一个项目组成员，作为配置管理员，负责整体的SVN管理，并且规定项目组的每一个成员及时提交文件，每天下班时，需确认编辑的文件是否被提交，重要版本需要更新最新版本之后，再提交 | 项目前期，都是在熟悉SVN中度过，后期没有好的配置管理流程，问题依然层出不穷 |
| 实施基础设施 | II 2.1 提供充足的资源、资金和培训来开发和执行过程。 | √ | 移动公司现有员工2000多名；注册资金达到一亿人民币；在日本也注册有分公司；作为受托方，运行专业领域的知识，调配人员和设备。 | × |
| II 2.2 建立和更新过程并验证过程是否得到遵循。 | √ | 基于行业最佳实践的基础上，在整个软件开发的生命周期中，移动公司的工程师和客户方面，通过优化的工作流程来保证每一个项目都能实现最佳的交付成果。 | × |
| 管理性能与度量 | MPM 2.1 从选定的业务需求和目标中推导出度量及性能目标并进行记录和更新。 | √ | 项目经理根据功能需求定义了多个模块 | × |
| MPM 2.2 开发、使用并持续更新度量项的操作性定义。 | √ | 将功能需求转化具体功能模块之后就分配给项目成员 | 由于项目成员都来自不同的项目，每个人都有不同的编码风格，所以在短时间之内无法很好地适应规定的新的编码规范 |
| MPM 2.3 根据操作性定义获取指定的度量数据。 | √ | 在搭建完成基本整体框架后，项目经理进行初步评审，技术人员进行白盒测试 | 技术人员不仅担负开发的责任，还要进行测试，没有专门的测试人员，技术人员身兼数职 |
| MPM 2.4 根据操作性定义分析性能和度量数据。 | √ | 评审和测试的结果指出缺失对软件可靠性和安全性处理，导致软件在测试阶段会出现死机、崩溃、自动退出等严重问题 | 对于团队技术成员的工作量很繁重 |
| MPM 2.5 根据操作性定义存储度量数据、度量规格和分析结果。 | √ | 所有的开发结果都以文件的形式用SVN进行管理 | 由于缺乏一定的配置管理规范，导致无法对历史稳定版本进行回溯 |
| MPM 2.6 采取行动解决已识别的阻碍实现度量及性能目标  的问题。 | √ | 根据前四周暴露出的质量问题，进行全面分析，认为主要问题在于软件系统的开发，重新理顺思路，制定测试计划 | 进行测试时，会出现测试不完全的问题，很多测试用例没有被覆盖，为进行极限测试等，直接导致质量问题、执行效率低 |
| 治理 | GOV 2.1 高级管理层根据组织需要和目标定义、维护并沟  通针对过程实施与改进的组织级方针。 | × | × | 项目缺少对于全过程的规范、高效率地管理 |
| GOV 2.2 高级管理层确保提供资源和培训用于建立、支持、执行、改进以及评价与预期过程的符合性。 | √ | 项目会在每一个阶段都对成员进行培训，并且也会对每个阶段的成果进行保存、评审和测试 | 缺少高效管理效率低下 |
| GOV 2.3 高级管理层确定其信息需要，并使用收集到的信息来治理及监督有效的过程实施和改进。 | √ | 对用户频繁修改的需求都进行分析并且制定了相应的修改方案 | 缺少高效管理效率低下 |
| GOV 2.4 高级管理层督促员工遵守组织级的方针并实现过程实施和改进的目标。 | √ | 会在开发的每一个阶段都会制定相应的规定，如：编码规范、测试规定、配置管理规定 | 会无形中给用户增加巨大的工作量 |

## 项目失败原因分析总结

2.2.1需求变更频繁

部长一职给项目组灌输“顾客就是上帝，客户满意度就是我们的终极目标”的思想，项目经理便毫无底线的答应客户各式各样的需求变更。在初期进行项目竞标后，项目经理进行最终的需求敲定，在敲定需求的基础上给用户提供了软件UI/UE的展示效果图，此时对需求的理解只是局限于当前的需求，并没有考虑到后续客户是否还会进行需求变更，也没有获取客户对于需求变更的承诺。

案例中，客户一个电话便可以要求进行需求变更，没有邮件或变更需求申请书，而项目组成员接到需求变更只是默默去接受，部分需求变更甚至没有落实到资源，以至于没有无法证明是否是客户提出的变更，无法维护需求的可回溯性，也导致测试人员，不了解最新的需求，导致测试不全面的恶性后果。

每次需求变更、作业担当都需要评估相应的作业量。

所以必须集中注意于现有的需求管理模式，重新定义新的需求管理模式，以应对需求变更的问题。

2.2.2设计、评审不足

项目工期紧，常成为很多人推卸责任的一种理由。

项目组成员拿到需求之后，没有对负责的模块进行整体的思考度量，没有罗列出几种解决方案，并从中选取最优的解决方案，而是自己想到什么实现方法就直接使用，完全没有设计的概念。部分开发担当认为，设计文档只是为了应对客户，完全没有设计价值，纯粹是浪费时间的一项功能。存在部分新人，不懂得利用公司存在的资源库，资源没有进行合理利用，造成了资源和时间的浪费，无法提交高质量的代码。经验丰富的员工因为担当着其他大量的高难度的开发工作，分身乏术，无法兼顾评审其他人代码的工作。

所以进行设计、评审工作时必须集中注意力于总体设计，即：拓展性、重用性、简洁性；设计评审时，注意是否能够快速发现设计缺陷，利用检查单，帮助评审人员进行评审工作；代码设计风格；代码评审等方面；注意技术解决方案实现流程等。

2.2.3 验证任务分配错误

所有的测试过程不应该有一个人来完成，即便他是最经验丰富的测试工程师，测试过程中，每个人考虑的随机测试方向均不同，不同的思路会增加更多不可见的测试方向。

完全依赖于开发人员进行白盒测试，会存在不可预期的风险。开发人员已经疲于应对需求的不断变更、功能的实现并且工期紧、任务重。

2.2.4 配置管理缺失

案例中，项目初始只是对项目组成员进行简单的使用培训，然而在使用过程中，依旧存在不同程度的问题。孙经理咨询几个其他项目组负责人对于此问题的处理方法，重新制定使用规则，然而孙经理忽略了一个问题，软件的配置管理不等同于软件的培训和使用规则的制定。配置管理涉及到七个特定的实践，即便最后设置一个配置管理员，配置管理员只是管理SVN的人员权限问题、是否迁入等。

# 参考文献

1. 柏雪. YD公司B软件项目质量管理案例研究[D].大连理工大学,2015.
2. 杨琳.中小型软件项目质量管理方法与应用探讨[J].软件产业与工程,2012(02):41-43+48.
3. 孙丁为.项目管理质量、进度、成本三要素探讨[J].现代商贸工业,2010,22(22):409-410.