



发起人高层会议总结报告

RAGHAV S. NANDYAL

SITARA TECHNOLOGIES PVT. LTD.

Dear Raghav Nandyal:

一、总体发现

多彩贵州数字科技股份有限公司本次采用 **CMMI V3.0**（开发域）标准，与评估师团队紧密协作，对软件工程和项目管理流程展开了全面的诊断、梳理与整合。通过详尽的文档审查和广泛的访谈，深入覆盖软件开发与管理的各个环节，这不仅让我们对 CMMI 国际标准有了深刻的学习与理解，更为公司的发展带来了诸多积极影响。**HMLA Raghav Nandyal** 领导的团队精准洞察潜在问题，凭借深厚的专业知识为开发团队的效率提升提供了切实可行的优化建议，在审查过程中，不仅关注软件研发流程，还对业务流程及管理方式进行了全方位审视。以实现赞助目标，使用名为 **SPRUM** 的专有战略管理工具，使用 **Measurements® Systemic Process Review Using Measurements - SPRUM®**

（**Raghavan S. Nandyal** 的注册商标）进行系统流程审查，以获得更深入的见解“如何以可靠的测量指标为重点，提高高成熟度流程的有效性并使其实用化”。运用 **SPRUM** 这一战略管理工具及相关测量系统进行深度分析，为公司的流程重塑指明方向，使我们对流程绩效模型的关键作用有了更为直观且贴合实际的认知，尤其是明确了影响系统测试缺陷密度的有效因素。

评估过程中，公司核心成员与评估师深入交流，从产品全生命周期管理、项目资源精细调配、人才培养等多个维度获得了大量前瞻性的指导和宝贵建议，这使我们更加清晰地认识到公司在 **CMMI** 高成熟度实施方面的现状，发现了潜在问题与改进方向，同时也极大地提升了团队对标准化流程的认知与执行能力。经过公司内部深层次的回溯研讨，我们发现评估师提出的建议与企业实际情况高度切合，公司将全面分析并吸收这些结论，依据实际情况制定详细且可行的整改措施，确保每条建议都能有效落地实施。

此次 **CMMI5** 评审对公司而言是一次全面检验研发实力与管理水平的重要契机，更是难得的学习提升机遇。我们将以此为新起点，持续追求卓越，不断优化流程，强化过程能力建设，构建并完善持续改进机制，全力推动公司向更高层次迈进，在数字科技领域不断书写新篇章，创造更多价值与成就。

二、经验教训

我们在学习与实践 **CMMI5** 过程改进的过程中，从实际项目出发，结合高成熟度模型统计知识，从工程管理、项目管理、支持过程、过程管理四个维度，由发现项目问题深入过程改进的各个环节，制定对应改进措施并进行详细总结。

工程过程

在用户需求调查过程中，一定要将功能性需求信息进行详细说明，以便于项目成员能够在没有

详细设计的情况下明白用户真实需求。举个例子，像“登录模块”应该描述清楚与天翼智慧智慧社区平台进行单点登录的方式描述清楚，附该平台的接入说明文档，以确保相关项目人员能够理解需求和快速开发。基于此，我们将进行以下改进：

在用户需求调查过程中，确实需要对用户的需求进行详细描述，因为需求的不确定性会导致项目的不确定和重复开发。因此，我们将进行下列改进：

充分和用户确认需求，并用文字详细记录下来；

在与项目实施过程中，多与项目成员进行需求沟通，确保需求的准确性，让项目成员能准确理解需求。如果用户需求用文字理解比较难懂，我们可以使用图片进行表达或者附上说明链接。

在编码过程中，基线代码必须确保完全符合 Java 编码标准。发现代码的方法中有几个无法访问的代码、注释的代码和未使用的变量声明实例。像这种与编码标准的不一致会增加后续的测试和代码维护而导致的返工。我们将进行以下改进：

1、针对无法访问的代码，借助专业代码分析工具（如 IntelliJ IDEA 的代码检查功能、Eclipse 的静态分析插件等）精准定位。若因业务逻辑变更不再需要，果断删除；若仅是路径错误，依据程序流程调整条件判断或方法调用，使其可访问，避免代码“死角”造成理解障碍。

2、面对注释代码，若为调试临时添加且调试完毕，直接删除；若涉及功能更迭，暂不启用但后续可能复用，添加详细注释，注明用途、搁置原因与预计启用场景，方便后续回溯。

3、未使用的变量声明方面，利用 IDE 内置的代码分析（如 Visual Studio Code 的代码智能提示）查找，确认无潜在引用后删除，精简代码结构，提升可读性。

项目管理

项目计划 WBS 中没有对计划任务的工作量进行评估和记录，使得规划数据无法进行更好的分析。此外，WBS 中对于任务的预计完成日期与实际完成日期过于一致，后续需要从实际出发，考虑到完成情况偏差。

针对这个建议，我们会从以下几个方面进行实施。

1、补充工作量评估：

首先对 WBS 中的每个任务进行详细的工作量评估，包括所需的工时、资源和人力等；其次使用项目管理工具或软件来辅助评估，确保数据的准确性和可靠性；最后将评估结果记录在 WBS 中，以便后续分析和跟踪。

2、调整预计完成日期和实际完成时间：

考虑到实际工作中的各种因素，如任务复杂度、资源可用性、团队成员技能水平等，对 WBS 中的任务预计完成日期和实际完成时间进行合理调整，实时记录实际完成时间。

3、实施偏差分析：

定期对项目进度进行监控，比较实际完成情况与计划之间的差异；对于存在偏差的任务，进行深入分析，找出原因并采取相应的纠正措施；更新 WBS 中的预计完成日期和工作量评估，以反映最新的项目状态。

4、加强沟通与协作：

确保项目团队内部以及与项目相关的其他利益相关者之间的有效沟通；定期召开项目会议，讨



论项目进度、问题和解决方案；鼓励团队成员提出意见和建议，共同完善项目计划。

5、持续优化 WBS：根据项目实际情况和经验教训，不断优化 WBS 的结构和内容；确保 WBS 能够准确反映项目的范围、任务、工作量和时间表等信息。

基于讨论结果制定改进措施，明确具体责任人、完成时间节点等，督促责任人按计划执行，团队持续跟踪执行情况，定期复盘总结，根据实际调整改进举措，让项目估算更精准，任务完成更高效。在项目结束后或每个关键阶段结束时，组织总结会议，回顾活动的执行情况、成本效益以及经验教训。将成功的经验和失败的教训纳入组织的知识库或最佳实践库，以便在未来的项目中借鉴和应用。鼓励团队成员分享自己的经验和见解，促进团队之间的知识共享和学习。持续改进和优化。

支持过程

在 PQA 检查单中增加对高成熟度内容的检查项，确实能够提升项目过程和产品的质量，确保它们符合更高的标准。有助于提升项目的整体绩效和组织的竞争力。

为了项目过程和产品的质量符合更高的标准，在 PQA 检查单中增加对高成熟度内容的检查项。我们制定了以下措施：

1、定义高成熟度检查项。基于高成熟度模型的要求，结合项目的实际情况，明确需要增加的高成熟度检查项，包括过程定义、项目管理、需求管理、风险管理、质量保证、团队协作与沟通、培训与持续改进等方面。

2、制定详细的检查清单与标准。将每个高成熟度检查项细化为具体的检查点，确保每个检查点都具有可操作性和可衡量性。为每个检查点设定评分标准，明确达到何种程度才能获得相应的分数，以便进行量化评估。

3、更新 PQA 检查单。将新制定的高成熟度检查项整合到现有的 PQA 检查单中，确保检查单全面覆盖项目过程和产品的关键方面。根据项目的规模和复杂度，调整 PQA 检查的频率，确保在高成熟度要求的关键阶段进行充分的检查和评估。

4、培训与宣传。对负责 PQA 检查的人员进行专业培训，确保他们熟悉高成熟度要求，能够准确执行检查任务。通过内部培训、会议和宣传材料等方式，向项目团队成员宣传高成熟度的理念和实践，提升团队的整体素质。

实施与监控。按照新的 PQA 检查单进行检查，记录检查结果，并对发现的问题进行记录和跟踪。根据检查结果，分析存在的问题和不足之处，制定改进措施，并持续监控改进效果。定期审查 PQA 检查单的有效性，根据项目的实际情况和高成熟度要求的变化，对检查单进行必要的调整和优化。

过程管理

编码生产率的模型基于代码复用率和开发人员技能这两个因素，而这两个因素又与多个子因素或影响因素相关联。为了提高模型的准确性并实现相关的业务目标，必须充分阐述和理解这些影响因素，特别是需求的稳定性和设计的清晰度对代码复用的可能影响，并确定“必要”和“充分”条件的影响。

同意，实施此提案，需要采取一系列步骤来确保模型的准确性和实用性。以下是详细的实施计划：



1. 需求分析与稳定性评估

收集需求：与业务部门紧密合作，收集并记录所有项目需求，确保需求完整性和准确性。

需求变更管理：建立严格的需求变更流程，控制需求变动的频率和范围，以提高需求的稳定性。

影响评估：定期评估需求变更对项目进度、成本和质量的影响，确保变更的合理性和必要性。

2. 设计清晰度提升

设计规范制定：制定统一的设计规范和标准，包括命名规则、代码结构、接口定义等，提高设计的一致性和可读性。

设计评审机制：引入设计评审流程，由资深开发人员或架构师对设计方案进行审查，确保设计的合理性和可行性。

设计文档完善：要求开发人员在提交代码前，必须编写详细的设计文档。

3. 代码复用策略优化

公共库建设：建立和维护一个公共代码库，将常用的功能模块和工具类集中管理，方便开发人员复用。**复用激励机制：**设立奖励机制，鼓励开发人员积极参与代码复用，分享优秀的代码片段和解决方案。

复用培训：定期组织代码复用培训，提高开发人员的复用意识和能力，教授他们如何有效地查找和使用已有代码。

4. 开发人员技能提升

技能培训计划：制定针对不同层次开发人员的培训计划，包括编程语言、框架、工具使用等方面的培训。

知识分享会：定期举办技术分享会，让开发人员交流心得和经验，共同学习和成长。

个人发展路径：为每位开发人员规划清晰的职业发展路径，提供必要的学习资源和支持，帮助他们不断提升技能水平。

5. 模型构建与验证

数据收集：收集历史项目的数据，包括代码复用率、开发人员技能水平、项目完成时间等关键指标。

模型构建：基于收集到的数据，使用统计方法和机器学习算法构建编码生产率模型。

模型验证：通过实际项目的应用和反馈，不断调整和优化模型参数，提高模型的准确性和可靠性。

6. 持续监控与改进

性能监控：实施持续的项目性能监控，跟踪关键指标的变化趋势，及时发现问题并采取措施。

反馈循环：建立有效的反馈机制，鼓励开发人员提出改进建议，不断优化开发流程和模型。

迭代更新：定期回顾和更新模型，以适应新的技术发展和业务需求变化。



三、现实意义

CMMI5 高成熟度评估为我们公司带来了丰富的项目管理和软件过程管理知识，提升了公司团队在研发与技术层面的整体竞争力。通过这一过程，我们不仅对 CMMI5 有了更深入的理解，也体会到了它的现实意义。

1、通过实施 CMMI 高成熟度流程，我们在项目估算中找到了影响项目研发工作量估算不够精准的问题，如在估算中采用人员平均能力进行衡量，未考虑到当前参与项目成员的具体能力，这使得在实际实施过程中就会存在高低偏差，针对这一问题，EPG 和项目团队随后积极建设成员能力库，使得在后续评估工作量时，根据具体人员的能力匹配度进行有效评估，从而减少估算偏差，提高估算的有效性。

2、高成熟度的量化实施使项目管理流程更加标准化和自动化，如引入自动化工具，可以减少手动操作和重复劳动，提高工作效率。同时，通过量化分析，可以更早地识别潜在的质量问题和缺陷，及时采取措施进行修正，从而提高最终产品的质量。

3、通过使用高成熟度模型，利用历史数据和统计进行预测分析，可以更准确地估算项目进度、成本和风险，为决策提供有力支持。基于量化数据的深入分析，还可以更合理地制定和调整项目计划，确保项目按时、按质、按量完成，提升我司产品交付率增强客户自信心。

4、在 CMMI3.0 学习中我们了解到它能为企业提供一套全面的、可定制的过程改进框架，有助于提升企业的项目管理能力和软件研发能力，从而提高产品质量和开发效率。同时，CMMI3.0 还强调了持续改进的重要性，鼓励企业不断优化自身的管理和开发过程，以适应不断变化的市场需求和技术环境。

5、高成熟度的实施能够向客户和合作伙伴展示其软件开发能力和质量保证水平，可以增强企业的市场竞争力，吸引更多的客户和合作伙伴，为企业带来更多的商业机会和发展空间。

6、CMMI5 适用于大规模、高质量、复杂度高的软件开发，通过量化的风险评估和管理，可以更准确地识别潜在风险，制定有效的风险应对策略，降低项目失败的可能性。高质量的产品交付和准确的项目进度预测有助于提升客户满意度，增强客户信任和忠诚度。

总之，通过实施 CMMI5 高成熟能力模型对于我们公司具有深远的现实意义。它不仅提升了我们的软件研发和管理水平，还为公司带来了更高的竞争力和商业利益。因此，我们将积极拥抱 CMMI5，不断推动自身的改进和发展。

四、改进措施

首先，感谢评估师和评估团队老师在这次评审过程中为我们提供的许多有价值的发现和建议，针对 CMMI5 评估过程中收到的问题发现项，我们专门成立了专项改进小组，负责监督和推进各项改进措施的落地执行，确保改进工作有条不紊地开展，并且我们会结合实际改进情况，编写成一份详尽的整改报告，此报告作为公司组织资产为后续项目做一个参考和依据。

根据本次评估团队发现的问题，我们将从以下 4 个方面进行整改

1、项目管理改进

首先我们会引入现有上级单位提供的研发云作为项目管理工具，细化工作分解结构（WBS）和项目进度计划，增加监控节点频率，及时发现并解决项目偏差。同时建立资源池管理系统，对人力资源和物力资源进行统一调配和动态平衡，提高资源利用率。

2、研发过程优化

在需求方面强化需求获取与分析过程，采用用户画像、原型设计等方法，提高需求的完整性和准确性；在技术研发方面完善技术选型和架构设计评审机制，邀请内外部专家参与评审，确保技术方案的可行性和先进性。同时，加强代码质量管理，实施代码静态检查和自动化测试，降低代码缺陷密度的同时提高产品的稳定性和可靠性。

3、组织过程资产建设与共享

通过对组织过程资产库（OPAL）进行了全面梳理和扩充，涵盖项目管理、工程过程、质量保证等各个领域，新增各类优秀实践案例、模板和检查单。同时，通过内部培训、知识库平台推广等方式，促进组织过程资产的共享与复用，有效避免重复劳动和知识流失。

4、质量保证与过程改进

通过强化质量保证（QA）团队的独立性和权威性，拓宽 QA 检查范围，从单纯的过程合规性检查向过程有效性评估转变。QA 发现的问题及时反馈给项目团队，并跟踪问题的解决情况，确保问题得到彻底关闭。通过持续的过程改进活动，组织的 CMMI 过程域实践得到进一步巩固和优化，使得我司成熟度水平稳步提升。

通过本次在评估中发现的问题，让我们意识到自身的不足，同时我们也清醒地认识到，CMMI5 高成熟度过程改进是一个持续的、渐进的过程，永无止境。未来，我们将继续学习 CMMI5 相关知识，根据我司实际发展战略不断探索和改进，不断优化组织的过程体系，持续提升过程能力和绩效水平，为实现企业的战略目标提供坚实的保障，向着更高更有效的过程成熟度迈进，在全球市场竞争中脱颖而出，创造更大的商业价值和社会效益。

最后，衷心感谢评估师和各位老师的辛勤付出和专业指导。感谢评估组提出的发现，我们很认同这些发现，感谢评估师和评估团队给出的宝贵意见。这些发现对于我们提高软件开发效率和质量，有很大帮助。后续我们会在工作中吸收应用、认真落实改进，通过改进持续提高软件开发效率和质量。并进一步实施高成熟度流程，帮助我们进一步识别关键因子和过程，同时对其监控、改进，从而提高客户满意和盈利能力，实现可持续发展。

我在此授权并同意您本人和 SITARA Technologies 在 SITARA 的出版渠道上分享我们的评估成果，在 SITARA Technologies 认为合适的情况下宣传我们的评估成果。

多彩贵州数字科技股份有限公司





EXECUTIVE SESSION BRIEFING - SPONSOR FEEDBACK

Overall findings

Colorful Guizhou Digital Technology Co., Ltd. underwent an evaluation based on the CMMI V3.0 (Development Domain) standard. Team members worked in close collaboration with the appraisal team, enabling a comprehensive diagnosis, refinement, and integration of software engineering and project management processes. Through detailed document reviews and extensive interviews, all aspects of software development and management were thoroughly examined. This process not only deepened the understanding of CMMI international standards but also brought significant positive impacts to the company's growth.

The team, led by HMLA Raghav Nandyal, accurately identified potential issues and leveraged profound expertise to provide practical optimization recommendations for improving the development team's efficiency. The review process focused not only on software development workflows but also on business operations and management practices, ensuring a holistic evaluation. To achieve the appraisal objectives, a proprietary strategic management tool, SPRUM, was employed. The [®] Systemic Process Review Using Measurements - SPRUM[®] (a registered trademark of Raghavan S. Nandyal) was utilized to perform a systematic process review, offering deeper insights into "how focusing on reliable measurement metrics can enhance the effectiveness and practicality of HM practice processes." The strategic management tool SPRUM, combined with its measurement systems, facilitated in-depth analyses, providing direction for process reengineering within the company. This approach offered a clearer and more practical understanding of the key role of process performance models, particularly in identifying factors that effectively influence the defect density in system testing. During the appraisal, core members of the company engaged in extensive discussions with the appraisers, gaining forward-looking guidance and valuable insights from various perspectives, including product lifecycle management, precise resource allocation for projects, and talent development. These discussions provided a clearer understanding of the company's current implementation status of CMMI HM practices, uncovering potential challenges and areas for improvement. Simultaneously, the team's awareness and execution of standardized processes were significantly enhanced.

Following internal retrospectives and discussions, the appraisal team's recommendations were found to align closely with the company's actual needs. These recommendations will be thoroughly analyzed, and detailed, actionable corrective measures will be developed to ensure their effective implementation. This CMMI Level 5 appraisal served as a comprehensive evaluation of the company's research and development capabilities and management practices, as well as a rare learning opportunity. Using this milestone as a new starting point, the company is committed to pursuing excellence, optimizing processes, strengthening process capability, and establishing a continuous improvement mechanism. These efforts aim to propel the company to higher levels of achievement, continually creating new value and success in the digital technology field.

Lessons Learned

In our journey of learning and applying CMMI Level 5 process improvement, we have taken a project-driven approach, integrating the statistical principles of the HM practice model. By focusing on four key dimensions—engineering management, project management, support processes, and process management—we have systematically addressed project issues, developed targeted improvement measures, and conducted comprehensive analyses and summaries of each step.

Engineering management

During user requirement surveys, it is crucial to provide detailed explanations of functional requirements to ensure that project team members can fully understand users' actual needs, even without detailed design documentation. For instance, a requirement like the "login module" should clearly describe the single sign-on (SSO) method for integration with the Tianyi Smart Community platform, along with the platform's access documentation. This ensures that the relevant project team members can accurately understand the requirements and proceed with development efficiently.

Based on this, we will implement the following improvements: **Thorough Requirement Confirmation:** Collaborate closely with users to confirm their requirements and document them comprehensively in written form. **Enhanced Communication with Team Members:** Maintain frequent communication with project members during implementation to ensure the accuracy of requirements and facilitate their understanding. If textual descriptions are challenging to interpret, provide visual aids or include explanatory links to clarify the requirements effectively. To ensure that the baseline code fully adheres to Java coding standards, we have identified issues such as inaccessible code, commented-out code, and unused variable declarations. These inconsistencies can lead to increased testing and maintenance efforts, ultimately causing rework. To address these issues, we will implement the following measures: **Resolving Inaccessible Code:** Use professional code analysis tools (e.g., IntelliJ IDEA's code inspection or Eclipse's static analysis plugins) to precisely identify inaccessible code. If the code is obsolete due to changes in business logic, remove it decisively. If the issue is caused by path errors, adjust condition checks or method calls according to program workflows to ensure accessibility and prevent "dead code" that could hinder comprehension. **Managing Commented-Out Code:** For code temporarily added for debugging, remove it immediately after debugging is complete. If the code is related to feature updates and temporarily unused but potentially reusable, add detailed comments explaining its purpose, why it is disabled, and when it is expected to be reused to facilitate future reference. **Eliminating Unused Variable Declarations:** Utilize IDE-integrated code analysis tools (e.g., Visual Studio Code's intelligent code suggestions) to identify unused variable declarations. Ensure there are no potential references to the variables, then remove them to streamline the code structure and enhance readability. These improvements aim to ensure a more efficient development process, minimize the risk of rework, and improve code maintainability, clarity, and overall project success.

Project management

The Work Breakdown Structure (WBS) in the project plan lacks a thorough evaluation and record of task workloads, making it difficult to analyze planning data effectively. Additionally, the estimated completion dates for tasks are overly consistent with the actual completion dates, failing to account for execution discrepancies. To address these issues, we propose the following improvements: **Workload Evaluation:** Conduct a comprehensive evaluation of the workload for each task in the WBS, including required hours, resources, and manpower. Use project management tools or software to ensure accuracy and reliability. Record the evaluation results in the WBS for future tracking and analysis. **Adjusting Completion Dates:** Revise the estimated completion dates in the WBS to reflect real-world factors, such as task complexity, resource availability, and team skill levels. Maintain real-time records of actual completion dates to capture any discrepancies and improve future planning.

Variance Analysis: Regularly monitor project progress by comparing actual outcomes with planned targets. For tasks with significant variance, perform root cause analysis and implement corrective actions. Update the WBS with revised completion dates and workload estimates to reflect the current project status accurately. **Enhanced Communication and Collaboration:** Strengthen communication within the project team and with stakeholders. Schedule regular project meetings to discuss progress, identify issues, and develop solutions. Encourage team members to provide feedback and suggestions to improve the project plan.



Continuous WBS Optimization: Continuously refine the WBS based on project realities and lessons learned. Ensure it accurately represents the project scope, tasks, workloads, and schedule, adapting as needed to align with ongoing project requirements. Based on these measures, assign clear responsibilities, set deadlines for task completion, and monitor progress to ensure accountability. Continuously track implementation, review outcomes periodically, and adjust strategies to improve project estimates and task efficiency.

At the conclusion of the project or key milestones, hold review meetings to evaluate execution, cost-effectiveness, and lessons learned. Document successful practices and shortcomings in an organizational knowledge base or best practices repository for future projects. Encourage team members to share their insights and experiences, fostering a culture of knowledge-sharing and learning. Emphasize ongoing improvement to enhance project management practices and outcomes.

Support

Adding HM practice items to the PQA checklist is expected to improve the quality of project processes and products, ensuring that they meet higher standards. This improvement will contribute to enhancing the overall project performance and the competitiveness of the organization.

To ensure the quality of project processes and products aligns with higher standards, the following actions have been taken to incorporate HM practice items into the PQA checklist:

- 1. Defining HM Practice Items:** Based on the requirements of the HM practice model and the specific needs of the project, the necessary HM practice items are identified. These include areas such as process definition, project management, requirements management, risk management, quality assurance, team collaboration and communication, training, and continuous improvement.
- 2. Creating Detailed Checklists and Standards:** Each HM practice item is broken down into specific, actionable, and measurable checkpoints. Scoring criteria are established for each checkpoint, clearly defining the standards needed to achieve a particular score, facilitating quantitative assessment.
- 3. Updating the PQA Checklist:** The newly defined HM practice items are integrated into the existing PQA checklist to ensure that key aspects of both project processes and products are comprehensively covered. The frequency of PQA checks is adjusted based on the project's size and complexity, ensuring thorough evaluations are conducted at critical stages as outlined by HM practices.
- 4. Training and Awareness:** Personnel responsible for performing PQA checks undergo professional training to ensure they are familiar with HM practice requirements and can execute inspections accurately. HM practice principles are promoted through internal training sessions, meetings, and informational materials to improve the overall skill set of the project team.
- 5. Implementation and Monitoring:** Inspections are conducted based on the updated PQA checklist, and results are documented. Any issues identified are tracked and addressed. Based on inspection findings, deficiencies are analyzed, corrective actions are implemented, and their effectiveness is monitored continuously. The PQA checklist is periodically reviewed for effectiveness, and adjustments or optimizations are made as needed, considering the project's actual conditions and the evolving requirements of HM practices.

Process Management

The model for coding productivity is based on two factors: code reuse rate and developer skill, both of which are associated with multiple sub-factors or influences. In order to improve the accuracy of the model and achieve related business goals, these influencing factors must be thoroughly defined and understood, particularly the potential impact of requirement stability and design clarity on code reuse, as well as the influence of "necessary" and "sufficient" conditions. To implement this proposal, a series of steps must be taken to ensure the accuracy and practicality of the model. The detailed implementation plan is as follows:

- 1. Requirement Analysis and Stability Assessment Requirement Collection:** All project requirements will be collected and recorded in close collaboration with the business departments to ensure the completeness and accuracy of the requirements.
- Requirement Change Management:** A strict change management process will be established to control the frequency and scope of requirement

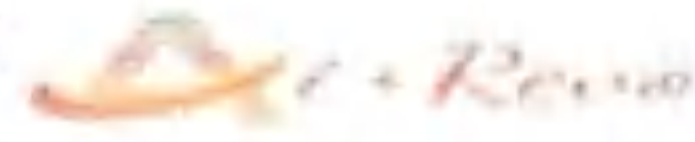
changes, aiming to improve the stability of the requirements. Impact Assessment: The impact of requirement changes on project schedule, cost, and quality will be regularly assessed, ensuring the reasonableness and necessity of the changes. 2. Improving Design Clarity Design Standards Development: Unified design standards and guidelines, including naming conventions, code structure, and interface definitions, will be established to enhance consistency and readability. Design Appraisal Mechanism: A design appraisal process will be introduced, where senior developers or architects review the design proposals to ensure the rationality and feasibility of the design. Design Documentation Improvement: Developers will be required to write detailed design documents before submitting code. 3. Optimizing Code Reuse Strategies Public Library Construction: A public code library will be built and maintained, consolidating common functional modules and utility classes for easy reuse by developers. Reuse Incentive Mechanism: A reward system will be established to encourage developers to actively participate in code reuse and share excellent code snippets and solutions. Reuse Training: Regular code reuse training will be organized to enhance developers' awareness and ability to reuse code, teaching them how to effectively search for and use existing code. 4. Enhancing Developer Skills Skill Training Plans: Training plans will be created for developers at various skill levels, covering programming languages, frameworks, and tool usage. Knowledge Sharing Sessions: Regular technical sharing sessions will be held to allow developers to exchange insights and experiences, facilitating collective learning and growth. Personal Development Pathways: Clear career development paths will be planned for each developer, with necessary learning resources and support provided to help them continually improve their skills. 5. Model Construction and Validation Data Collection: Historical project data, including code reuse rates, developer skill levels, and project completion times, will be collected, along with other key metrics. Model Construction: Based on the collected data, statistical methods and machine learning algorithms will be used to construct the coding productivity model. Model Validation: The model will be continually adjusted and optimized based on feedback from real project applications, improving its accuracy and reliability. 6. Continuous Monitoring and Improvement Performance Monitoring: Ongoing performance monitoring of the project will be implemented, tracking key metrics' trends, promptly identifying problems, and taking corrective actions. Feedback Loops: An effective feedback mechanism will be established to encourage developers to submit improvement suggestions, continuously optimizing the development process and model. Iterative Updates: The model will be regularly reviewed and updated to adapt to new technological developments and changes in business requirements.

Relevance

The CMMI High Maturity model has provided us with extensive knowledge and insights into project management and software process management, leading to an upgrade in our company's overall competitiveness in research and development (R&D) and technology. It has been a transformative experience for the team, and the colleagues involved in the model training have gained a deeper understanding of the model's practical significance:

By implementing the HM high maturity process, we identified key factors affecting project development workload deviations, such as "coding productivity," "rework rate," and "code reuse library." Subsequently, the EPG and project teams actively worked on building the code reuse library, effectively increasing code reuse, which in turn improved coding efficiency. On the other hand, by providing skills training for developers and training on code reuse, we helped developers adopt good development and reuse habits, ultimately reducing rework and workload deviations.

The implementation of high maturity quantification in projects helps reduce waste and improve process efficiency, as well as boost work efficiency. High maturity practices also help identify and eliminate defects, improve product quality, and, as a result, enhance project delivery rates and customer satisfaction. By introducing high maturity implementation, we established the PPM model, which will be applied in the project. The model will be used at various project milestones—such as planning, requirements, design, coding, and testing—to conduct Monte Carlo simulations to predict the likelihood of achieving goals, identify risks, and analyze any cases where certainty is less than 70%. Timely CAR (Corrective Action



Request) analysis will be conducted to effectively increase the likelihood of goal achievement.

CMMI HM is suitable for large-scale, high-quality, and complex software development projects. By implementing CMMI HM, companies can ensure the smooth progress and high-quality delivery of software projects, meeting customer expectations and demands. This not only helps enhance our company's market reputation and brand image but also brings more business opportunities and potential partners. CMMI 3.0 introduces new requirements for data management, personnel management, and virtual delivery, which will help our company adopt CMMI in a more flexible and customized way according to its needs. The implementation of high maturity makes us more competitive in the market, enabling us to attract more customers and partners. By establishing a culture of continuous improvement, we encourage employees to actively propose improvement suggestions and place importance on quantitative analysis methods to ensure the effectiveness and measurability of improvement measures. This culture fosters a sense of involvement and ownership among all employees, making them more attentive to the details and innovation in their work, ultimately driving the efficient operation of the overall business.

The implementation of the CMMI HM high maturity model has profound practical significance for our company. It not only helps improve our software R&D and management capabilities but also brings higher competitiveness and business benefits. Therefore, we will actively embrace CMMI HM and continuously drive our improvement and development.

Improvement measures

First, we would like to express our gratitude to the appraisers and evaluation team for the valuable findings and suggestions provided during this appraisal process. In response to the issues identified during the CMMI5 appraisal, a dedicated improvement team has been formed to oversee and drive the implementation of corrective measures, ensuring that the improvement initiatives are carried out systematically. Based on the actual progress of these improvements, a detailed corrective report will be prepared. This report will serve as an organizational asset for the company and act as a reference and guide for future projects.

In response to the issues discovered by the evaluation team, corrective actions will be taken in the following four areas: The R&D cloud provided by our higher-level organization will be adopted as the project management tool. The Work Breakdown Structure (WBS) and project schedule will be refined, and the frequency of monitoring nodes will be increased to promptly identify and resolve project deviations. Additionally, a resource pool management system will be established to centrally allocate and dynamically balance human and material resources, improving resource utilization.

The requirement acquisition and analysis process will be strengthened, utilizing methods such as user personas and prototype design to improve the completeness and accuracy of requirements. In terms of technical R&D, the technology selection and architecture design appraisal mechanism will be enhanced by inviting both internal and external experts to participate in appraisals, ensuring the feasibility and advancement of technical solutions. Furthermore, code quality management will be reinforced through static code analysis and automated testing, reducing code defect density and enhancing product stability and reliability.

The Organizational Process Asset Library (OPAL) will be thoroughly reviewed and expanded, covering project management, engineering processes, quality assurance, and other relevant areas. Best practice



case studies, templates, and checklists will be added. To promote the sharing and reuse of these organizational assets, internal training and knowledge-sharing platforms will be utilized, effectively preventing redundant work and knowledge loss. The independence and authority of the Quality Assurance (QA) team will be strengthened, and the scope of QA checks will be broadened, shifting from simple process compliance checks to evaluating the effectiveness of processes. Any issues identified by QA will be promptly communicated to the project team, with follow-up to ensure the issues are fully resolved. Through continuous process improvement activities, the CMMI process domain practices within the organization will be further strengthened and optimized, leading to a steady improvement in our maturity level.

The issues identified during this evaluation have highlighted areas of improvement for us. We also recognize that CMMI5 HM practice process improvement is an ongoing, progressive journey without an end point. Going forward, we will continue to learn about CMMI5-related knowledge, explore and improve our processes based on the company's strategic development, and continue optimizing our organizational systems. This will enhance our process capabilities and performance levels, providing a solid foundation for achieving our strategic goals. We aim to advance toward higher and more effective process maturity, stand out in global market competition, and generate greater commercial value and social benefits.

Finally, we sincerely thank the HMLA and the ATMs for their hard work and professional guidance. We appreciate the findings provided by the ATMs, which we fully agree with, and we are grateful for the valuable suggestions offered. These findings will significantly help improve our software development efficiency and quality. Moving forward, we will implement these suggestions in our work, carefully executing improvements to enhance software development efficiency and quality. We will continue to implement HM practice processes, helping us identify key factors and processes, monitor, and improve them. This will ultimately increase customer satisfaction and profitability, enabling sustainable development.

As the sponsor of this appraisal who has received the executive session briefing, I hereby authorize and give consent to you and SITARA Technologies to share our appraisal accomplishments on SITARA ' s publishing channels giving publicity to our appraisal accomplishment as SITARA Technologies deems it fit.

Colorful Guizhou Digital Technology Co., Ltd.

Sponsor: Kun Ren

Jan 08, 2025

