Intellitrans 智运

江苏智运科技发展有限公司

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

本次基于 CMMI 基准标准，**我公司采用 CMMI V3.0（开发域）标准,** 对我司的软件开发和项目管理流程进行了全面地梳理和整合。借助这次的评估让我司的同仁们对本标准做了更细致深入的研究和探讨。在评估期间，进行了大量的文档审查和面谈工作，并在日常开发过程和管理中准确地识别了已知和潜在的问题。**HMLA Raghav Nandyal 领导的团队**凭借出色的专业知识，精准地发现了各方面的潜在问题。我们特别感谢 HMLA 和 ATM 他们使用幽默风趣的例子帮助我们更好地理解各种流程中的优缺点，并进行了多次面谈。同时，在基准评估中，评估团队除了软件工程流程外，还对我们的业务流程进行了全面审查，**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”**。根据 HMLA 的意见，我们在 SPRUM® 的基础上进行了重塑工作，帮助我们简化并建立了实用的回归模型。经过重塑，我们对流程绩效模型的重要性有了更深刻、更实际/更贴近实际的理解。我们现在有了一个更直观的模型，它能反映影响研发工作量偏差的真实因素（基于先行指标而非滞后指标）。

通过与 HMLA 的沟通讨论，我们获得了宝贵的建议和指导，这为公司未来的发展指明了方向。评估过程让我们进一步的深入审视了公司现行的 CMMI 高成熟度的实施现状。有了更为全面和深入的了解，发现了不少潜在的问题与改进空间。通过评估锻炼了我们的团队，提升了员工对标准化、规范化流程的认识和执行能力。在此次评估中，各位评估老师们从产品需求、设计、开发、测试、交付、决策分析、项目管理、项目资源的精细化管理、PPM 模型的引入、人才培训机制、编码规范等多个方面提出了宝贵的改进建议。公司高层、EPG 过程改进小组、项目经理、质量保证、以及配置管理等核心成员，经过对这些问题的深入追溯与研讨，达成了共识：这些建议与公司实际情况高度吻合，符合度达到 95%，我们将认真消化吸收评估结果，结合公司实际情况，制定具体的改进措施和计划，确保每一项建议都能得到有效落实。

这次 CMMI5 评估审核不仅是对公司研发能力和管理水平的全面检验，更是一次难得的学习和提升的机会。我们将以此为契机，不断总结经验、优化流程、提升能力，推动公司在追求卓越、实现持续发展的道路上迈出更加坚实的步伐。

二、经验教训

我们在学习和实践 CMMI5 的过程中，从项目本身出发，从工程过程、项目管理、支持过程、过程管理四个方面为切入点分别进行详细总结。

工程过程

项目过程中，对于需求变更后，需要对项目管理中的文档进行记录，并重新估算，这有利于项

目成本的准确计量。这个建议强调了需求变更的重要性，以及需求变更后的持续跟踪，以此保证项目的准确性。因此，我们将进行下列的改进措施：

1、制定标准化的变更流程：申请-初审-方案论证-CCB 审查-变更实施-变更监控-变更效果评估。按照标准化的流程进行变更管理。

2、实施变更监控管理：变更过程中，需要对所涉及到的所有人员进行沟通，涉及到的文档进行及时的更新，在变更完成后，要重新确定项目基线。

3、加强培训管理：加强对项目组所有成员的培训，建立大家对项目变更的统一认识，并组织实施。

4、审查和审计：变更完后后，需要对变更进行审查，确保变更得到正确的实施，项目已经恢复正常。

通过以上这些措施，将确保需求变更的标准化和完整性，以及在整个项目中的可执行程度，减少错误，保证项目的成功实施。

在软件开发和测试过程中，单一进入方法和多个退出的实例可能会导致测试开销的增加，同时死代码的存在也可能对系统的安全性和稳定性构成潜在威胁，因此优化程式码并考虑执行效能条件很重要。

如果一个方法中存在多个入口点和出口点，这确实可能增加测试复杂性和难度，同时也可能引入死代码或者其他不易察觉的问题，下面是一些建议，以此来优化代码并减少测试消耗和潜在风险。

1、为了降低测试开销，可以考虑重构代码，使其具有更清晰的逻辑结构，例如采用机构化编程的方法，将代码划分为具有单一职责的函数或模块，此外，使用自动化测试工具和技术也可以有效提高测试效率。

2、为了消除死代码，需要进行代码审查和分析，确定哪些代码段是冗余的或者不需要的，并将其从程序中移除，这有助于减少软件的大小，提高性能并降低安全风险。

3、可以采用算法优化，数据结构改进和并行计算来提高代码的执行效率。

4、可以通过封装和抽象来隐藏复杂的逻辑流程，提供清晰的接口供外部调用。

5、减少出口点，尽量减少方法中的 Return 语句数量，理想情况下，一个方法应该只有一个出口点，这有助于简化逻辑流并使测试更加直接。

项目管理

项目估算过程中，当前各阶段工作量是将返工工作量和各阶段本身工作量混淆在一起。没有明确体现相关返工工作量。

针对这个建议，我们会从以下几个方面进行实施。

1、项目开发阶段不同的工作任务都会出现不同程度的返工现象，可以对项目开发不同阶段：需求、设计、编码、测试、试运行与验收等工作任务的返工工作量进行单列，通过计算得出相关分配工作量比例。

2、项目开发阶段主要工作量分配在需求、设计、编码阶段，其占比占整个开发阶段近 90%，同步返工率也占整个开发阶段绝大部分。针对这三个主要返工阶段，单独对其进行返工工作量统计计算，得出相关返工比例作为整个项目开发返工工作量。

3、根据项目结束的估算分析，项目编码阶段占整个项目开发工作量的 53%，其中编码生产率低是编码阶段返工率的主要原因，并对编码生产率低问题做了原因分析和相关解决方案。

4、通过层层阶段抓住重点问题原因分析，使整个项目返工工作量在各个阶段更加明确。

项目利用 word 文档对项目进度计划进行管理，包含了相关要素；并有甘特图进行跟踪，但 word 中甘特图效果不佳；可以利用 excel 表对甘特图进行更细颗粒度的跟进，更有利于持续监控管理。

针对这个建议，我们会从以下几个方面进行实施改进。

1、利用 excel 表可以建立一个直观、易于理解的甘特图，可以更好地组织和跟踪项目进度。

2、如果项目中存在重要的里程碑，可以使用特殊符号或颜色来标记，并在 excel 表格中进行说明。

3、在甘特图管理中，创建 excel 表格，设置时间轴，绘制甘特条形图，添加依赖关系和标记里程碑可以更好地理解任务之间的关系和及时监控进度控制。

4、鉴于 excel 表对甘特图可以更细颗粒度的跟进进度管理，后续项目开发会一直使用此方法进行甘特图的编制。

支持过程

在质量保证方面，由于每个阶段的返工量是未知的，此项目的生命周期计划工作量是需求 20%，设计 15%，编码 53%，测试 9%，试运行和验收 3%，但是每个阶段的返工工作量没有进行拆分单独计算，导致无法直接更加细微的观察到质量成本耗费，我们应该以量化的方式改善这一项，将返工量也添加进去，这样能方便我们观察质量成本，更精确。

为了确保质量管理的准确性和有效性，为项目管理和决策提供更为可靠的数据支持。我们制定以下措施：

1、在今后的质量保证工作中，将软件研发实施中将需求分析、设计、编码、测试、验收阶段所产生的返工量单独拆分计算，这样能够准确的看出质量成本的消耗。在软件开发实施的各阶段，去统计各阶段产生的返工量，这样各个阶段都能够统计反馈出实际的数据，能够提升质量保证工作的准确性。

2、根据项目的工作量、返工率等估算项目的成本，因为软件开发项目的主要成本就是人力，有了成本估算，这样能帮助项目经理更容易的预算出项目的整体成本，在项目发生变更的时候，也能够快速得到项目成本变化的趋势，找到项目成本的控制办法，帮助项目经理更好的进行成本管理。

3、做好质量成本估算能够为过程改进指明具体改进方向，帮助企业实现过程优化和效率提升，也可以优化运营成本，对于返工率的成本估算能够更加明确，从而提高整体运营效率，节省成本。

回归建模时，因素选择是一个至关重要的步骤，它直接决定了模型的预测能力和对现实问题的解释力度。在选择影响因素时，必须着重考虑那些能够显著增强模型预测能力的因素，这些因素通常具有直接且可量化的效益或影响。

针对这个建议，我们会从以下几个方面进行实施。

1、首先，促进预测能力的因素在回归建模中占据核心地位。这些因素通常是与输出变量具有显著相关性的输入变量，它们能够有效地解释输出变量的变化。在选择这些因素时，需要依据领域知识、历史数据和统计方法进行综合分析，确保所选因素能够真实反映输入输出关系。

2、其次，具有直接效益或影响的因素也是选择过程中需要重点关注的。这些因素通常对输出变量具有直接作用，能够显著影响预测结果。在选择这些因素时，需要考虑它们对预测目标的实际贡献，以及在实际应用中的可行性和可操作性。

3、此外，在选择因素时，滞后指标通常比超前指标差。这是因为滞后指标反映的是过去的信息，而超前指标则试图预测未来的情况。在大多数情况下，过去的信息对于预测当前和未来的情况

更具参考价值，因为它们已经发生并且可以被准确测量。相反，超前指标通常基于假设和预测，其准确性和可靠性相对较低。

4、因此，在回归建模中，应该优先选择那些能够促进预测能力的、具有直接效益或影响的因素，并尽量避免使用滞后指标。当然，这并不是说滞后指标完全没有价值，在某些特定情况下，它们可能提供有用的信息。但是，在大多数情况下，应该优先考虑那些更能反映输入输出关系的、更可靠的指标。

5、最后，回归建模所用因素选择的决策必须基于对预测能力的促进和对实际问题的解释力度。在选择因素时，需要充分考虑因素的直接效益/影响以及其在时间维度上的可靠性，确保所选因素能够最大程度地提升模型的预测准确性和解释力度。

过程管理

对于设定的，2023 年底目标，降低研发工作量偏差改进目前。工作量偏差的减少，有多少是来源于需求的改进结果，或者是其他软件过程的改进，不够清晰。

需要将研发工作量偏差的来源量化到需求、设计、编码、测试等子过程，我们计划实施的过程如下：

1、数据收集：收集并分析每个子过程中的工作量数据，初收集各阶段本身的工作量外，同时一并收集各阶段的返工工作量。通过对这些数据进行统计分析，可以得出每个子过程返工情况对工作量偏差的影响度。

2、质量评估：评估每个子过程在项目中的质量表现，比如需求的准确性和完整性、设计的合理性、编码的质量和测试的覆盖率等。根据质量评估结果，可以量化每个子过程返工情况，以及其对工作量偏差的影响。

3、过程改进：针对每个子过程的问题和瓶颈，制定相应的改进措施，并在实施后对返工工作量进行监控和评估。通过比较改进前后的数据，可以量化每个子过程改进对工作量偏差的影响。

4、综合分析：综合考虑需求、设计、编码、测试等子过程返工的影响，结合实际情况和数据，量化每个子过程对工作量偏差的贡献度。这样可以更清晰地了解各个子过程在工作量偏差中的具体作用，为进一步的过程优化提供指导。

5、持续跟踪：对分析结果和项目返工数据持续进行跟进管理，获取优质的返工和偏差数据，更好的管理返工性能。为组织持续改进助力！

通过以上方法，可以将研发工作量偏差的来源量化到需求、设计、编码、测试等子过程，帮助团队更好地理解和优化工作流程，提高研发效率和质量。

研发工作量偏差模型过于复杂，需要将三层模型精简为两层模型。在考虑模型因子的时候应该将研发过程中的涉及到的因子作为优先考虑的重要指标使用。

针对提出的建议，以下是一些后续的改进措施：

1、新立项项目原则上采用二层模型进行建模。

2、在模型因子选择上，优先考虑研发过程的指标，其次根据项目情况再考虑是否将滞后指标加入模型。

3、建立持续改进机制。定期对改进措施进行回顾和总结，发现问题和不足，及时进行调整和优化。

三、现实意义

学习 CMMI5 高成熟能力模型让我们有了一次新的退变，使每个参与模型实践的同仁们了对 CMMI5 又一次更深入地了解：

1、实施 CMMI5 高成熟能力模型有助于我司在软件研发和管理领域达到业界领先水平。这代表着我司已经建立了一套成熟、高效、规范的软件开发和工程管理流程，能够在复杂多变的市场环境中迅速应对各种挑战，提升企业的竞争力和市场占有率。

2、CMMI5 的实施推动了我司对软件开发过程的持续改进和量化管理。通过对软件开发过程进行精细化的控制和度量，企业能够及时发现并解决开发过程中的瓶颈和问题，提升开发效率和质量。同时，量化管理使得我司能够对开发进度、成本和质量进行更加准确的预测和控制，为我司的决策提供有力支持。

3、CMMI5 的实施推动了自动化程序在软件开发中的应用。基于规则和量化分析的自动化程序可以大大提高软件开发的效率，减少人为失误，使得软件开发过程更加稳定、可控。这不仅有助于提升我司的研发能力，还能够为企业带来更高的经济效益。

4、CMMI5 适用于大规模、高质量、复杂度高的软件开发项目。通过实施 CMMI5，企业能够确保软件项目的顺利进行和高质量交付，满足客户的期望和需求。这不仅有助于提升我司的市场声誉和品牌形象，还能够为企业带来更多的商业机会和合作伙伴。

5、CMMI5 强调过程自我修正能力的提高，有助于企业降低软件开发风险。通过不断优化和完善软件开发过程，企业能够减少错误和缺陷的产生，提高产品的稳定性和可靠性。这不仅有助于提升客户满意度，还能够为企业节省大量的维护成本和修复成本。

6、CMMI5 在数据管理、人员管理、虚拟交付等方面对我们提出了新的要求，这将帮助我司更加灵活地根据其需求定制化采用 CMMI。

实施 CMMI5 高成熟能力模型对于我们公司来说具有深远的现实意义。它不仅有助于提升我司的软件研发和管理水平，还能够为我司带来更高的竞争力和商业利益。因此，我们会积极拥抱 CMMI5，不断推动自身的改进和发展。

四、改进措施

在本次评估中，HMLA 和 ATM 给我们提了很多的宝贵建议和意见，随后 EPG 团队将与项目人员紧密合作，共同针对评估中提出的建议进行深入讨论与识别。我们将形成一份详尽的《差距分析报告》---这份报告将成为我们现阶段组织过程改进的核心依据。基于这份报告，我们将精心制定《过程改进计划》，明确改进活动的步骤、时间安排、参与人员以及实施目标等关键信息。为确保改进工作的顺利进行，公司高层将与 EPG 成员共同讨论改进过程中可能出现的注意事项、潜在风险及障碍，并承诺提供充分的人、财、物等资源支持。我们将通过制定规范、调配资源以及迭代生产工具等手段，及时解决这些问题，为改进工作创造有利条件。在改进过程中，相关人员将保持高度的配合度。EPG 团队将根据发现的问题和分析报告，进行深入的根因分析，寻找有效的改进措施。随后，我们将选取合适的试点项目，对改进措施进行实际尝试。在试点数据通过假设检验后，我们将进一步推广部署这些改进措施，以期实现更大的效益提升。

在 EPG 团队实施改进的过程中，我们始终对改进流程保持全面的监控，详细记录每一步的效果，深入分析所收集到的度量数据，QA 团队、OT 团队等也会配合监督执行。我们精心挑选合适的模型

进行量化分析，构建预测模型和趋势分析，旨在精准预测软件开发过程和产品的质量走向。在项目实施期间，我们不断优化项目开发流程，努力提升工作效率，确保项目的顺利进行。针对代码规范性问题，EPG 团队进行了持续的改进工作，深入探索影响因子，评估并调整度量项，不断优化基线与模型，以期提升项目的整体水平。

此次的评估和日常实践，让我们对 CMMI5 有了更为深刻的理解，同时也让我们认识到自身的不足之处。实施 CMMI5 的过程就是一个不断追求完善、不断优化的旅程，我们始终秉持着不断进取、永不停歇的精神。通过培训提升团队技能、制定标准化流程、强化监控机制、加强内部沟通以及持续改进等多种方式，我们致力于提升组织的能力与成熟度，确保我们能够适应不断变化的软件发展环境和市场需求。我们对人员的技能要求也在日益提高，因为只有不断提升自身能力，才能更好地满足客户的需求，提高开发效率，从而在激烈的市场竞争中立于不败之地。提升过程管理水平，不仅能够增强客户满意度，还能提高我们的开发效率，使我们能够更好地适应市场环境，实现公司的长期愿景。因此，我们将把 CMMI5 作为提升产品质量和管理能力的重要工具，持续推动其在公司内的应用。

最后，我们衷心感谢评估师和各位 ATM 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细化吸收、认真落实改进，通过改进助力研发水平提升。持续提高软件开发效率和质量，提高客户满意，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。

江苏智运科技发展有限公司

发起人：

2024 年 5 月 19 日

江苏智运科技发展有限公司
3201000122067

EXECUTIVE SESSION BRIEFING - SPONSOR FEEDBACK

Overall findings

Based on the CMMI benchmark appraisal, CMMI V3.0 (Development Domain) standards were adopted to thoroughly review and integrate the software development and project management processes of our company. This appraisal allowed our colleagues to engage in detailed and in-depth research and discussion of these standards. During the appraisal, extensive document reviews and interviews were conducted, accurately identifying known and potential issues within our daily development processes and management.

The team led by HMLA Raghav Nandyal, with their outstanding expertise, precisely identified various potential problems. We are particularly grateful to HMLA and ATM for using humorous examples to help us better understand the strengths and weaknesses of various processes, and for conducting multiple interviews. Additionally, the appraisal team performed a comprehensive review of our business processes, besides the software engineering processes, to achieve the sponsorship goals. A proprietary strategic management tool named Systemic Process Review Using Measurements - SPRUM® (a registered trademark of Raghavan S. Nandyal) were used for systematic process reviews to gain deeper insights into "how to enhance the effectiveness of high maturity processes and make them practical with a focus on reliable measurement metrics." Remodeling exercise based on SPRUM® was undertaken based on the HMLA's inputs, has helped us to streamline and make practical, regression models. After the remodeling exercise, we have gained a deeper and a more practical/real-life understanding of the importance of process performance models. We now have a more intuitively appealing model that reflects the true factors (based on lead indicators instead of lag indicators), affecting the deviation of R&D workload.

Valuable suggestions and guidance were provided through communication and discussion with HMLA, pointing the way for the company's future development. The appraisal process allowed us to further scrutinize the current implementation status of our high maturity CMMI. A more comprehensive and in-depth understanding was obtained, revealing numerous potential issues and areas for improvement. The team was strengthened through the appraisal, and employees' understanding and execution capabilities of standardized and regulated processes were enhanced.

During this appraisal, valuable improvement suggestions were provided by the appraisers from multiple aspects, including product requirements, design, development, testing, delivery, decision analysis, project management, refined project resource management, PPM model introduction, talent training mechanisms, and coding standards. After thorough tracing and discussion of these issues, consensus was reached by the company's senior management, EPG process improvement team, project managers, quality assurance, and configuration management core members: these suggestions were highly consistent with the company's actual situation, with a conformity rate of 95%. The appraisal results will be carefully digested and absorbed, and specific improvement measures and plans will be formulated in line with the company's actual situation to ensure that each suggestion is effectively implemented.

This HM appraisal was not only a comprehensive examination of the company's R&D capabilities and management level but also a rare opportunity for learning and improvement. This will be used as an opportunity to continuously summarize experiences, optimize processes, enhance capabilities, and propel the company towards more solid steps in the pursuit of excellence and sustainable development.

Lessons Learned

We will provide a detailed summary from four aspects: engineering process, project management, support process, and process management.

Engineering management

During the project, when requirement changes occur, documentation in project management needs to be recorded and re-estimated to ensure accurate cost measurement. This recommendation emphasizes the importance of requirement changes and continuous tracking after these changes to maintain project accuracy. Therefore, the following improvement measures will be undertaken: A standardized change process will be established: application - preliminary review - solution demonstration - CCB review - change implementation - change monitoring - change effect appraisal. Changes will be managed according to this standardized process. Change monitoring management will be implemented: During the change process, all involved personnel need to be communicated with, and the affected documents need to be updated promptly. After the change is completed, the project baseline must be re-established. Training management will be strengthened: Training for all project team members will be enhanced to establish a unified understanding of project changes and organize implementation. Review and audit: After the change is completed, a review must be conducted to ensure the change has been correctly implemented and the project has returned to normal. These measures will ensure the standardization and completeness of requirement changes and their executable degree throughout the project, reducing errors and ensuring the project's successful implementation. In software development and testing, a single-entry method with multiple exit instances can lead to increased testing overhead, while the presence of dead code can pose potential threats to system security and stability. Therefore, optimizing the code and considering execution efficiency conditions is important. If a method has multiple entry and exit points, it can indeed increase testing complexity and difficulty, and may introduce dead code or other undetectable issues. The following are some recommendations to optimize code and reduce testing overhead and potential risks: To reduce testing overhead, consider refactoring the code to have a clearer logical structure, such as using structured programming methods to divide the code into functions or modules with a single responsibility. Additionally, using automated testing tools and techniques can effectively increase testing efficiency. To eliminate dead code, conduct code reviews and analyses to identify redundant or unnecessary code segments and remove them from the program. This helps reduce software size, improve performance, and lower security risks. Algorithm optimization, data structure improvements, and parallel computing can be employed to enhance code execution efficiency. Encapsulation and abstraction can be used to hide complex logical processes, providing clear interfaces for external calls. Reducing exit points and minimizing the number of return statements in a method is advisable. Ideally, a method should have only one exit point, which helps simplify the logical flow and make testing more straightforward.

Project management

During the project process, after requirement changes are made, it is necessary to record and reevaluate the project management documents. This facilitates accurate measurement of project costs. The importance of requirement changes is emphasized by this suggestion, along with the continual tracking of changes to ensure project accuracy. Therefore, the following improvement measures will be undertaken: Establishment of standardized change processes: Application - Initial Review - Scheme Demonstration - CCB Review - Change Implementation - Change Monitoring - Change Effect Appraisal. Change management will be conducted according to standardized processes. Implementation of change monitoring management: Communication with all involved parties and timely updating of documents are required during the change process. After the change is completed, the project baseline needs to be reestablished. Strengthening training management: Enhance training for all members of the project team to establish a unified understanding of project changes and organize implementation. Review and audit: After changes are made, a review is necessary to ensure correct implementation and project recovery to normalcy. These measures will ensure the standardization and integrity of requirement changes, as well as the feasibility throughout the entire project, reducing errors and ensuring successful project implementation. In software development and testing processes, a single entry method and multiple exit instances may increase testing overhead. Additionally, the presence of dead code may pose potential threats to system security and stability, thus optimizing code and considering execution efficiency conditions is crucial.

If a method has multiple entry and exit points, it can indeed increase testing complexity and difficulty, while also potentially introducing dead code or other less noticeable issues. Here are some suggestions to optimize code and reduce testing overhead and potential risks:

To reduce testing overhead, consider refactoring the code to have a clearer logical structure, such as using structured programming methods to divide the code into functions or modules with single responsibilities. Additionally, the use of automated testing tools and techniques can effectively improve testing efficiency.

To eliminate dead code, code review and analysis are needed to determine which code segments are redundant or unnecessary, and remove them from the program. This helps reduce software size, improve performance, and lower security risks.

Algorithm optimization, data structure improvements, and parallel computing can be employed to enhance code execution efficiency.

Use encapsulation and abstraction to hide complex logical processes and provide clear interfaces for external calls.

Reduce exit points and minimize the number of Return statements in methods. Ideally, a method should have only one exit point, simplifying logical flows and making testing more straightforward.

Support

In terms of quality assurance, because the rework volume for each stage is unknown, the planned workload for the lifecycle of this project is 20% for requirements, 15% for design, 53% for coding, 9% for testing, and 3% for trial operation and acceptance. However, the rework volume for each stage has not been separately calculated, resulting in an inability to directly observe subtle quality cost expenditures. This aspect should be improved quantitatively by including rework volume, facilitating the observation of quality costs more precisely. To ensure the accuracy and effectiveness of quality management, and to provide more reliable data support for project management and decision-making, the following measures will be implemented:

In future quality assurance work, the rework volume generated during the stages of requirements analysis, design, coding, testing, and acceptance will be separately calculated. This will accurately show the consumption of quality costs. By tallying the rework volume generated in each stage of software development, each stage can provide actual data feedback, thereby enhancing the accuracy of quality assurance work.

Estimate project costs based on project workload and rework rates, as the main cost of software development projects is labor. Cost estimation helps project managers more easily budget the overall project cost and quickly identify trends in project cost changes when changes occur, aiding in cost control. Proper estimation of quality costs can provide specific directions for process improvement, helping companies achieve process optimization and efficiency improvement. Cost estimation for rework rates can be more precise, thereby improving overall operational efficiency and cost savings.

In regression modeling, factor selection is a crucial step that directly determines the predictive ability of the model and its explanatory power for real-world problems. When selecting influencing factors, factors that significantly enhance the model's predictive ability should be given priority consideration. These factors typically have direct and quantifiable benefits or impacts.

In response to this recommendation, implementation will be approached from the following aspects: First, factors promoting predictive ability will occupy a central position in regression modeling. These factors usually exhibit significant correlation with the output variable and effectively explain changes in the output variable. When selecting these factors, comprehensive analysis based on domain knowledge, historical data, and statistical methods is required to ensure that the selected factors accurately reflect input-output relationships. Secondly, factors with direct benefits or impacts should also be given special attention during the selection process. These factors typically have a direct effect on the output variable and can significantly influence predictive results. When selecting these factors, their actual contributions to the predictive target, as well as feasibility and operability in practical applications, need to be considered. Additionally, lagging indicators are generally more reliable than leading indicators when selecting factors. This is because lagging indicators reflect past information, whereas leading indicators attempt to predict future situations. In most cases, past information is more valuable for predicting current and future situations because it has already occurred and can be accurately measured. Conversely, leading indicators are often based on assumptions and predictions, making their accuracy and reliability relatively lower. Therefore, in regression modeling, factors that promote predictive ability and have direct benefits or impacts should be prioritized, while the use of lagging indicators should be minimized. Of course, this does not imply that lagging indicators are completely devoid of value; in certain specific situations, they may provide useful information. However, in most cases, factors that more accurately reflect input-output relationships and are more reliable should be prioritized. Finally, the decision on factor selection for regression modeling must be based on promoting predictive ability and explanatory power for real-world problems. When selecting factors, the direct benefits/impacts of factors and their reliability over time need to be fully considered to ensure that selected factors maximize the predictive accuracy and explanatory power of the model.

Process Management

For the set target by the end of 2023, improving the current deviation in development workload is prioritized. It is not clear how much reduction in workload deviation stems from improvements in requirements or from enhancements in other software processes. The sources of development workload deviation need to be quantified into sub-processes such as requirements, design, coding, and testing. The proposed process to achieve this is as follows: Data collection: Collect and analyze workload data for each sub-process, including both the workload of each phase and rework workload. Statistical analysis of these data can determine the impact of rework situations in each sub-process on workload deviation.

Quality appraisal: Appraise the quality performance of each sub-process in the project, such as the accuracy and completeness of requirements, the rationality of design, the quality of coding, and the coverage of testing. Based on the quality appraisal results, the impact of rework situations in each sub-process on workload deviation can be quantified.

Process improvement: Develop corresponding improvement measures for the problems and bottlenecks in each sub-process, and monitor and appraise rework workload after implementation. Comparing data before and after improvements can quantify the impact of each sub-process improvement on workload deviation.

Integrated analysis: Considering the impact of rework in sub-processes such as requirements, design, coding, and testing, along with actual situations and data, quantify the contribution of each sub-process to workload deviation. This can provide clearer insights into the specific roles of each sub-process in workload deviation, guiding further process optimization.

Continuous tracking: Continuously manage and follow up on the analysis results and project rework data to obtain high-quality rework and deviation data, thus better managing rework performance and facilitating continuous improvement for the organization.

Through these methods, the sources of development workload deviation can be quantified into sub-processes such as requirements, design, coding, and testing, helping the team better understand and optimize workflow, thus enhancing development efficiency and quality.

The development workload deviation model is overly complex and needs to be simplified from a three-layer model to a two-layer model. When considering model factors, factors involved in the development process should be prioritized as important indicators.

In response to the proposed suggestions, here are some subsequent improvement measures: New project proposals should generally adopt a two-layer model for modeling. When selecting model factors, prioritize indicators from the development process, and then consider whether to include lagging indicators in the model based on the project situation. Establish a continuous improvement mechanism. Regularly review and summarize improvement measures, identify problems and shortcomings, and make timely adjustments and optimizations.

Relevance

The adoption of the High Maturity Capability Model has provided us with a new opportunity for further understanding of HM by all colleagues involved in model :

The implementation of the High Maturity Capability Model assists our company in achieving industry-leading levels in software development and management. This signifies that our company has established a mature, efficient, and standardized software development and engineering management process, capable of rapidly addressing various challenges in the complex and dynamic market environment, thus enhancing our competitiveness and market share. The implementation of HM drives continuous improvement and quantified management of our software development process. Through fine-grained control and measurement of the software development process, the company can promptly identify and address bottlenecks and issues in the development process, thereby enhancing development efficiency and quality. Additionally, quantified management enables the company to make more accurate predictions and controls over development progress, costs, and quality, providing strong support for company decision-making. The implementation of HM promotes the application of automation programs in software development. Automation programs based on rules and quantitative analysis can significantly improve the efficiency of software development, reduce human errors, and make the software development process more stable and controllable. This not only helps enhance our development capabilities but also brings higher economic benefits to the enterprise. HM are suitable for large-scale, high-quality, and high-complexity software development projects. By implementing HM , enterprises can ensure the smooth progress and high-quality delivery of software projects, meeting customer expectations and requirements. This not only helps enhance our market reputation and brand image but also brings more business opportunities and partnerships to the enterprise. HM emphasize the enhancement of process self-correction capabilities, helping enterprises reduce software development risks. By continuously optimizing and improving the software development process, enterprises can reduce the occurrence of errors and defects, improve product stability and reliability, thus enhancing customer satisfaction and saving a significant amount of maintenance and repair costs for the enterprise. HM propose new requirements for us in data management, personnel management, virtual delivery, etc., which will help our company customize the adoption of CMMI more flexibly according to its needs. The implementation of the HM High Maturity Capability Model has profound practical significance for our company. It not only helps improve our software development and management capabilities but also brings higher competitiveness and business benefits to our company. Therefore, we will actively embrace HM and continuously promote our own

Improvement measures

In this appraisal, valuable suggestions and opinions were provided to us by HMLA and ATM. Subsequently, close collaboration between the EPG team and project personnel will be initiated to thoroughly discuss and identify the recommendations made during the appraisal. A comprehensive "Gap Analysis Report" will be compiled, which will serve as the core basis for our current organizational process improvement efforts. Based on this report, a meticulous "Process Improvement Plan" will be formulated, outlining key information such as improvement activities, timeline, participating personnel, and implementation objectives. To ensure the smooth progress of improvement work, senior management will engage in discussions with EPG members regarding potential considerations, risks, and obstacles in the improvement process, and commit to providing ample resources such as personnel, finances, and materials. By establishing standards, allocating resources, and iterating production tools, we will promptly address these issues and create favorable conditions for improvement work. Throughout the improvement process, relevant personnel will maintain a high level of cooperation. The EPG team will conduct in-depth root cause analysis based on identified issues and analysis reports, seeking effective improvement measures. Subsequently, suitable pilot projects will be selected to implement these improvement measures in practical trials. Upon passing hypothesis testing of pilot data, we will further promote and deploy these improvement measures, aiming for greater benefits.

During the implementation of improvement by the EPG team, comprehensive monitoring of the improvement process will be maintained, with detailed documentation of the effects of each step and in-depth analysis of collected metric data. The QA team, OT team, and others will also collaborate to supervise execution. Carefully selected models will be used for quantitative analysis, constructing predictive models and trend analyses, with the aim of accurately predicting the quality trends of software development processes and products. Throughout the project implementation, we will continuously optimize project development processes, striving to improve work efficiency and ensure smooth project progression. Continuous improvement efforts will be made by the EPG team regarding code standardization issues, with in-depth exploration of influencing factors, appraisal, and adjustment of metrics, and continuous optimization of baselines and models to enhance the overall project level.

This appraisal and our daily have provided us with a deeper understanding of HM, while also making us aware of our own shortcomings. The implementation of HM is a journey of continuous pursuit of perfection and optimization, and we always adhere to a spirit of continuous advancement and never-ending improvement. Through various methods such as training to enhance team skills, establishing standardized processes, strengthening monitoring mechanisms, enhancing internal communication, and continuous improvement, we are committed to enhancing the organization's capabilities and maturity, ensuring that we can adapt to the ever-changing software development environment and market demands. Our requirements for personnel skills are also increasing day by day because only by continuously improving our own capabilities can we better meet customer needs, improve development efficiency, and stand undefeated in the fierce market competition. Improving process management capabilities not only enhances customer satisfaction but also improves our development efficiency, enabling us to better adapt to the market environment and achieve the company's long-term vision. Therefore, we will consider HM as an important tool for enhancing product quality and management capabilities and continue to promote its application within the company.

Finally, we sincerely thank HMLA and all ATM for their hard work and professional guidance. We appreciate the findings presented by the appraisal team, and we fully acknowledge these discoveries. We also thank HMLA and the appraisal team for their valuable insights. These findings greatly assist us in improving software development efficiency and quality. Moving forward, we will refine and implement improvements in our work, thus contributing to the enhancement of our development capabilities. Continuous improvement in software development efficiency and quality, along with increased customer satisfaction, will enable sustainable development.

Jiangsu Intellitrans Co., Ltd.

Sponsor:

May 19, 2024

江苏智运科技发展有限公司
3201000122067