发起人高层会议总结报告

RAGHAV S. NANDYAL

SITARA TECHNOLOGIES PVT. LTD.

一、总体发现

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”。
这次我们依托 CMMI V3.0 标准开展了 CMMI 5 级基准评估，从文档核查、人员访谈到量化分析评审。从评估结果来看，整体情况是客观的，也比较贴近我们当前的实际运行状态。确实暴露出了一些短板和提升空间，这些问题不是局部的，而是贯穿在我们整体管理链条中的关键环节。公司高层、EPG 过程改进小组以及项目、质量、研发等核心团队也围绕评估结果做了集中研讨，大家总体上达成了共识：本次评估结果是客观的、可信的，与我们实际情况的匹配度是比较高的，高达 95%。

二、经验教训

为了更系统地对所有改进项进行复盘与提炼，我们将分散的改进建议和实施方案，按照工程过程（Engineering）、项目管理（Project Management）、支持过程（Support）以及 过程管理（Process Management）四大核心领域进行深度的归纳与总结。

工程过程（Engineering Processes）

该领域暴露出的核心问题主要集中在需求合规与工程变更的正式化、代码的可读性与低级命名失误，以及因不良编码习惯导致的不可维护性与不可测试性。原型与实际实现的断层与范围蔓延风险
项目组使用 Pixso 进行屏幕原型设计作为功能设计的基础。虽然原型中构思的内容与 Java 实际实现的产品之间存在视觉不一致可能不影响核心功能，但客户对这种外观和感觉（Look-and-feel）不一致的认可是口头表达的，并未记录为正式协议。这种缺乏文档化和正式确认的模式带来了潜在风险，在真实产品中如果未形成文档并获得客户同意，极易引发范围蔓延（Scope Creep）、工作量浪费或计划外工作，从而直接威胁到项目设定在 15% 以上的高盈利目标和生产率目标。 在编写的代码中观察到多处难以阅读、从而在后期维护时极易引入缺陷的复杂逻辑实例。例如，当 dataitem 的 getStr 方法获取 isNormal 字段时，返回的条目却在判断逻辑中被用于测试是否为 "abnormal"（异常）。这种将获取正常字段的函数用来做异常校验的逻辑设计过于复杂，令人难以理解。 在逻辑设计上，将判断 siteDictionary 是否为非空（Non-null）与进一步验证其名称是否为硬编码值 "1" 结合在一起，是一种非常危险的逻辑组合方式。这种强耦合的方式不仅无法适应设计的变更，还会使集成测试变得非常复杂。此外，代码中存在多处直接在逻辑判断（如 if 逻辑）中调用函数返回值而不赋值给临时变量、并直接对返回值进行取反的糟糕编码习惯（例如 if (!(StringUtils.isEmpty(...)))），导致测试极其困难且无法维护。

项目管理过程（Project Management Processes）

该领域反映出组织在定量模型建立的合理性、新工具引入的量化评估（ROI）以及前沿技术引发的风险管理意识上存在薄弱环节。量化模型缺乏合理的数据分群与聚类。项目组收集了生产率和盈利能

力的数据用于建立研发生产率模型，其中平均盈利能力为 15.82%，平均生产率为 0.756 FP/人日。虽然这些指标的单位未提供，但非常重要的一点是：应当以平均值为基准，将数值划分为“低于平均值”和“高于平均值”两类数据群（Clusters）。例如，9.3% 和 9.6% 远低于 15.82%，属于低于平均值；而 15.xx% 及以上的数值应当划分为另一类。否则，所建立的生产率模型对于完全低于平均值的数值将彻底失去实用价值和预测意义。 AI 新工具引入缺乏量化评估与优缺点对比，项目组虽然积极在项目中引入了人工智能（AI）新工具，但是在实施相关的 AI 措施时，并没有对比分析这些工具的优缺点，也没有对其投资回报率（ROI）进行量化评估。 口头协议带来的项目多维度风险暴露。由于客户对产品外观和感觉不一致的认可是口头的、未形成正式合规协议，这在项目管理和风险控制层面构成了潜在的重大风险源。未形成文档的客户意向会导致范围蔓延和计划外工作，频繁的返工成为常态后，直接使项目设定的 15% 以上高盈利目标及生产率基准无法实现。

支持过程（Support Processes）

支持领域的核心缺陷主要暴露在代码评审与质量保证的职责脱节、定量资产的版本控制不力，以及决策分析缺乏透明度。质量保证与同行评审未拦截低级命名拼写错误。在代码赋值前的调用中，频繁出现调用 FileUtil.Savelamge(...) 这一方法的现象（例如在 siteCheckRecord.setInfraredImgUrls(aiImgUrls); 之前）。目前尚不清楚 Savelamge 是一个刻意的拼写错误，还是应该为 SaveImage。虽然这种不正确的命名选择可能不会直接破坏产品功能，但确保代码能够被作者以外的人员维护以及实现代码的可重用性是至关重要的。此类代码走查（Walkthrough）和规范考量未能有效拦截，说明相关职责未能在产品质量保证（PQA）和同行评审环节得到有效落实。 虽然组织在组织级层面已经建立了过程性能模型（PPM），但目前没有任何迹象或记录表明这些模型是什么时候建立的，以及它们与更早期版本模型之间的关联性和有效性。由于无法明确过去有哪些模型、这些模型随着时间的推移如何演进和改进、以及未来这些模型的下一步发展规划，导致这些关键的量化资产未能纳入良好的版本控制和配置管理之中。 决策分析权重设置缺乏客观规则说明。建议项目在确定评价标准、设置决策准则权重时，增加对部分决策准则权重设置的规则或逻辑说明，否则将不利于提升和保障最终决策的客观性。 逻辑结构不可隔离测试，阻碍质量验证。由于代码中存在大量直接调用函数返回值而不赋值给临时变量、并在 if 逻辑中对返回值进行取反的糟糕编码方式，导致测试人员在进行质量验证和确认时，几乎无法对这些复杂的逻辑结构进行孤立且有效的测试，严重阻碍了验证工作的开展。

过程管理过程（Process Management Processes）

该领域揭示了组织在面对流程资产精简、治理思维重塑、基础设施赋能及组织级人才培养机制方面的系统性不足。流程资产定义冗余臃肿，缺乏敏捷合规性。作为项目执行基础的过程资产定义似乎存在大量的冗余且体量过于臃肿。过去 5-6 年间创建的过程定义、模板、检查清单和工作辅助工具存在大量为了“勾选”表面合规而引入的浪费性做法。为了精简流程、 slim down 流程膨胀（Process Bloat），组织必须考虑进行过程变更，从而在确保项目保持敏捷性的同时，依然符合合规标准。 线性治理思维导致僵化，缺乏对前沿领域的识别。现行的“业务目标、流程绩效结果导向模型”的路径表现出一种僵化的、线性的治理思维。在这种模式下，在业务目标设定之前，与问题识别相关的几个关键步骤实际上被忽略了。在当下的技术环境中，基于对人工智能（AI）领域的研究来识别问题和机遇，是治理中一个至关重要的缺失步骤。 前沿技术胜任力的组织级培训机制缺失。项目涉及新兴技术（如机器人广泛应用以及将自动化集成到构建方案的监控环节）的特定领域需求，要求相关人员在开始项目前就具备相应的领域胜任力。然而，目前在前沿技术方面赋能知识获取的正式机制仍属于个人自发行为，组织级缺乏系统性的规划。组织必须在组织级计划中强化这一正式机制，以实现战略性的胜任力增长，并建议在下一阶段的过程变更管理倡议中探索人员（PEOPLE）领域。目前组织内已制度化的既定流程和基础设施，能否很好地满足从未尝试过的全新项目方案的需求尚

不明确。实施基础设施必须做出调整，使其能够同时适用于“已知（Known）”类型和“未知（Unknown）”类型的项目——这对于规划依赖人工智能（AI）且具有未来导向型的项目范围界定而言是必不可少的。

三、现实意义

这些发现对企业的核心意义在于，它明确揭示了企业在技术敏捷转型、精细化量化管理、项目风险控制以及组织级核心资产沉淀上所面临的系统性瓶颈。将其转化为企业的商业与运营语言，主要体现为以下四个层面的深远影响。第一，它直接关乎企业核心利润率与核心资产的保护，能够有效消除亏损与浪费。企业在战略上设定了 15% 以上的高盈利目标。然而由于开发团队在需求前端频繁使用口头协议来认可原型与实际实现之间的断层，导致这些未形成文档的客户意向在后期演变为频繁的返工与范围蔓延。这种工作量的隐形浪费和计划外开支正在直接吞噬企业的净利润。同时，企业过去 5 至 6 年积累的过程资产过于臃肿冗余，员工为了表面合规去填表、勾选检查清单，导致了大量的形式主义和效率浪费。精简这些流程膨胀可以让企业重新获得敏捷性，让员工把精力聚焦在真正交付业务价值上。 第二，它能为企业全面布局 AI 战略扫清基础设施障碍，打通转型通路。企业现行的治理模式表现出一种僵化的、线性的治理思维。但在面对人工智能这一颠覆性技术时，企业在设定目标之前缺乏对 AI 领域的深度研究与问题识别，这会导致企业的战略方向与技术前沿脱节。另外，企业现有的制度化流程和基础设施只适合处理已知类型的传统项目。当面对未来导向、高度依赖 AI 的未知项目时，现有的体系根本无法为其提供合理的范围界定和有力支撑。企业必须升级底座，否则无法承接高难度的 AI 新业务。 第三，它有助于将技术的个人行为转化为组织资产，沉淀组织核心竞争力。在机器人和自动化集成等新兴技术领域，企业虽然有项目需求，但目前知识的获取和胜任力的培养完全属于员工的个人自发行为。一旦这些核心员工离职，企业在该领域的战力将直接清零。组织级必须建立正式的赋能机制，将个人胜任力转化为战略性的组织资产，并探索人员领域。此外，企业虽然建立了过程性能模型，但这些模型何时建立、如何演进完全缺乏记录和配置管理。缺乏版本控制意味着企业无法进行长期的、跨周期的定量改进，过往的数据资产极易变成无人能懂的死资产。 第四，它能够显著提升底层交付质量，降低软件长期维护成本，消除代码技术债务。代码中观察到多处反直觉的复杂逻辑以及不给临时变量赋值直接取反的糟糕编码习惯，这导致测试人员在进行质量验证和确认时，几乎无法对其进行有效的、孤立的质量验证。同时，这种极难阅读、无法维护的代码 和低级的拼写错误，在同行评审和质量保证环节没有被有效拦截。它们一旦流入生产环境，在后期维护时极易引入新的缺陷，从而成倍增加企业长期的售后技术支持和日常维护成本。 对企业管理层而言，这些发现绝对不是简单的研发部门的小毛病，而是一个高管层面的战略警示。企业不能用传统、臃肿、依赖口头承诺的旧管理机制，去指挥面向未来、高盈利要求、依赖 AI 创新 的新型战役。企业亟需在组织层面发起一次轻量敏捷化与技术管理正式化的微调与升级。

四、改进措施

在得知这些核心意义与系统性瓶颈后，企业在下一轮推进工程过程组（EPG）的改进工作时，应当彻底摆脱过去为了合规而流于形式的传统套路，转向以“轻量、敏捷、面向前沿技术”为核心的精益改进路径。

首先，针对流程定义冗余臃肿以及敏捷性缺失的问题，EPG 的首要任务是开展过程资产的精简与瘦身行动。过去的流程资产积累了大量为了勾选 CMMI 合规条目而引入的浪费性做法。在下一轮改进中，EPG 应当重新审视过去 5 至 6 年创建的过程定义、模板、检查清单和工作辅助工具。通过剥离冗余的行政化审批和无实质交付价值的表单，精简流程膨胀（Process Bloat）。这样既能减轻开发团队的流程负担，实现项目执行的轻量化，又能确保项目在保持高度敏捷性的同时，依然在底线上符合 CMMI 的合规标准。

其次，为了支撑企业布局 AI 等前沿技术战略，EPG 必须重塑现有的治理思维并升级实施基础设施。现行的线性治理思维缺乏在业务目标设定之前对前沿技术领域的深度问题识别与技术研究。EPG 应当在治理流程中嵌入基于人工智能等前沿技术领域的识别与机遇研究步骤，将僵化的线性治理升级为前瞻性的创新治理。与此同时，原有的基础设施仅适用于处理已知类型的传统项目，无法承接依赖 AI 且具有未来导向型的未知项目。EPG 需要对现有的既定流程和实施基础设施进行重构与扩容，使其具备同时兼顾和支撑“已知（Known）”与“未知（Unknown）”类型项目的双轨制运作能力，从而能够对未来派的 AI 项目进行合理的范围界定与交付支撑。

再者，针对技术管理口头化、代码债务堆积等底层交付痛点，EPG 应当推行关键工程行为的标准化与自动化强拦截机制。针对 Pixso 原型与实际 Java 实现之间存在的断层，EPG 必须出台明确的工程管理规范，严禁对需求外观和可行性进行纯口头的 acknowledge 确认。所有原型的视觉差异、技术可行性或需求变更，都必须强制形成文档化的正式协议并获得客户签字同意，以此作为风控红线，从源头上遏制由于口头承接带来的范围蔓延与盈利流失。在代码质量层面，鉴于同行评审和质量保证流于形式、无法拦截低级命名错误和不可测试的反取反逻辑结构，EPG 应当将代码走查（Walkthrough）和 PQA 审查职责进行刚性固化。通过引入静态代码扫描工具，将不合理的函数调用、不赋临时变量直接取反的结构以及低级拼写错误转化为自动化的强拦截规则。同时，要求开发团队对核心复杂逻辑强制进行单元测试编写，确保代码结构可进行隔离测试，从源头降低二次缺陷注入和长期的售后技术支持维护成本。

最后，EPG 需要将定量的资产管理以及人才培养机制从“自发自留”提升到“组织战略”的高度。在量化管理方面，企业虽然建立了研发生产率模型和过程性能模型（PPM），但在实际应用中缺乏合理的数据聚类，且模型演进缺乏配置管理。EPG 在下一轮改进中，应当指导度量团队以平均值为基准，将生产率和盈利数据科学地划分为高于平均值和低于平均值的不同集群（Clusters），避免模型对低均值数据失去指导意义。同时，将所有 PPM 模型纳入严格的版本控制与配置管理中，明确模型的建立时间、版本关联与演进路线规划，激活定量资产的长期改进价值。在人才赋能方面，鉴于目前机器人的广泛应用以及将自动化集成到监控环节等新兴技术能力依然属于个人自发行为，EPG 必须在组织级计划中强化在前沿技术方面赋能知识获取的正式机制。通过建立战略性的胜任力增长机制，有计划地将个人的技术能力沉淀为组织的持久战力，并在下一阶段的过程变更管理倡议中深度引入和探索 CMMI v3.0 的人员（PEOPLE）领域。

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。

合肥小步智能科技有限公司

发起人：李兵

2026 年 6 月 17 日

合肥小步智能科技有限公司

3401230398709

EXECUTIVE SESSION BRIEFING: SPONSOR FEEDBACK

Overall findings

To achieve the sponsorship objectives, the proprietary strategic management tool SPRUM® (Systemic Process Review Using Measurements)—a registered trademark of Raghavan S. Nandyal—was utilized to conduct a systemic process review. This approach provided deeper insights into "how to enhance the effectiveness of high-maturity processes and make them pragmatic, with a steadfast focus on reliable measurement indicators." On this occasion, we conducted a CMMI Level 5 Benchmark Appraisal leveraged against the CMMI V3.0 model, spanning document reviews, personnel interviews, and quantitative analysis reviews. The appraisal results present an objective reflection that closely aligns with our current operational reality. While the findings have indeed exposed certain shortcomings and opportunities for improvement, these issues are not isolated; rather, they represent critical links throughout our entire management chain. Following intensive, collaborative discussions centered on these findings among senior executives, the EPG (Engineering Process Group), and core teams across project management, QA, and R&D, a clear consensus was reached: the appraisal results are objective, credible, and demonstrate a high degree of alignment with our actual practices, reaching as high as 95%.

Lessons Learned

For a more systematic review and refinement of all improvement items, we have deeply consolidated and summarized the scattered improvement recommendations and implementation plans into four core areas: Engineering, Project Management, Support, and Process Management.

Engineering Processes

Core Issues in this Category: > The core issues identified in this domain primarily center on requirement compliance and the formalization of engineering changes, code readability alongside low-level naming missteps, and issues of non-maintainability and non-testability triggered by poor coding practices. The project team utilizes Pixso for screen prototyping as the foundation of functional design. Although the visual inconsistencies between the concepts conceptualized in the prototype and the actual product implemented in Java may not impair core functionality, customer acceptance of these "look-and-feel" misalignments was conveyed only verbally, without being recorded as a formal agreement. This lack of documentation and formal validation poses a latent risk. In a live project, failing to document and obtain explicit customer sign-off on such deviations can easily trigger scope creep, wasted effort, or unplanned work—directly threatening the project's aggressive targets of over 15% profitability and established productivity goals. Multiple instances of complex logic were observed within the source code, rendering it difficult to read and highly susceptible to defect injection during downstream maintenance. For instance, when the getStr method of a dataitem retrieves the isNormal field, the returned entry is subsequently used in the decision logic to test for an "abnormal" state. Designing a logic flow where a function meant to retrieve a "normal" field is used to validate an "abnormal" condition introduces excessive complexity and is highly counterintuitive.

Furthermore, combining the non-null verification of a siteDictionary with the subsequent validation of its name against a hard-coded value of "1" represents a highly volatile logical construct. This tight coupling not only fails to adapt to design fluctuations but also significantly complicates integration testing. Additionally, poor coding habits were prevalent where function return values were directly negated and evaluated within conditional statements (e.g., if (!(StringUtils.isEmpty(...)))) without being assigned to temporary variables. This pattern makes the code exceptionally difficult to isolate for testing and ultimately unmaintainable.

Project Management Processes

The core problems exposed in this field reflect that the organization has weak links in the rationality of quantitative model establishment, the quantitative evaluation (ROI) of new tool introduction, and the risk management awareness triggered by cutting-edge technologies. The project team collected productivity and profitability data to establish an R&D productivity model, where the average profitability was 15.82% and the average productivity was 0.756 FP/person-day. Although the units of these indicators were not provided, a very important point is: the values should be divided into "below average" and "above average" data clusters based on the average value. For example, 9.3% and 9.6% are far below 15.82% and belong to the below-average cluster; while values of 15.xx% and above should be classified into another category. Otherwise, the established productivity model will completely lose its practical value and predictive significance for values that are completely below the average. Although the project team actively introduced new Artificial Intelligence (AI) tools in the project, when implementing relevant AI measures, they did not compare and analyze the advantages and disadvantages of these tools, nor did they conduct a quantitative evaluation of their Return on Investment (ROI). Since the customer's approval of the look-and-feel inconsistency of the product was verbal and did not form a formal compliance agreement, this constituted a potential major risk source at the project management and risk control levels. Undocumented customer intentions will lead to scope creep and unplanned work. After frequent rework becomes a norm, it directly makes the high profitability target of over 15% and productivity baselines set by the project unachievable.

SUPPORT PROCESSES

The core defects in the support field are mainly exposed in the disconnection of responsibilities between code review and quality assurance, poor version control of quantitative assets, and the lack of transparency in decision analysis. In the calls before code assignment, the phenomenon of calling the method FileUtil.Savelamge(...) frequently occurs (for example, before siteCheckRecord.setInfraredImgUrls(aiImgUrls);). It is currently unclear whether Savelamge is a deliberate typo or should be SaveImage. Although this incorrect naming choice may not directly destroy product functions, it is crucial to ensure that the code can be maintained by personnel other than the author and to achieve code reusability. Such code walkthroughs and specification considerations failed to effectively intercept, indicating that the relevant responsibilities were not effectively implemented in the Product Quality Assurance (PQA) and peer review links. Although the organization has established Process Performance Models (PPM) at the organizational level, there is currently no sign or record to show when these models were established, and their correlation and effectiveness with earlier versions of models. Due to the inability to clarify what models existed in the past, how these models evolved and improved over time, and the next development plan for these models in the future, these key quantitative assets failed to be included in good version control and configuration management. It is suggested that when determining evaluation criteria and setting decision criteria weights, the project should add rules or logical descriptions for the weight setting of some decision criteria, otherwise it will not be conducive to improving and guaranteeing the objectivity of the final decision. Due to a large number of bad coding methods in the code that directly call function return values without assigning them to temporary variables and negate the return values in if logic, testing personnel can hardly perform isolated and effective tests on these complex logical structures during quality verification and validation, severely hindering the implementation of verification work.

PROCESS MANAGEMENT PROCESSES

This domain reveals systemic deficiencies within the organization regarding the streamlining of process assets, the reshaping of governance mindsets, infrastructure enablement, and organization-level talent cultivation mechanisms.

The process asset definitions serving as the foundation for project execution appear heavily redundant and overly bloated. Over the past 5 to 6 years, a massive volume of process definitions, templates, checklists, and job aids have been introduced purely as wasteful practices to "tick the box" for surface-level compliance. To streamline operations and slim down this Process Bloat, the organization must initiate process changes that ensure projects maintain agility while strictly adhering to compliance standards.

The current path of the "business objectives and process performance outcome-oriented model" exhibits a rigid, linear governance mindset. Under this paradigm, several critical steps related to problem identification are effectively bypassed before business objectives are even established. In today's technological landscape, identifying problems and opportunities based on research into the Artificial Intelligence (AI) frontier represents a vital yet missing link in governance.

The domain-specific requirements of projects involving emerging technologies—such as the widespread deployment of robotics and the integration of automation into build-scenario monitoring—demand that personnel possess corresponding domain competencies prior to project initiation. However, formal mechanisms for knowledge acquisition and enablement in these cutting-edge technologies remain self-driven, individual efforts, lacking systematic planning at the organizational level. The organization must reinforce this formal mechanism within its corporate-wide programs to achieve strategic competency growth. Furthermore, it is recommended to explore the PEOPLE domain in the next phase of process change management initiatives.

It remains unclear whether the established, institutionalized processes and infrastructure currently within the organization can effectively accommodate the needs of unprecedented, novel project scenarios. The implementation infrastructure must be adapted to concurrently support both "Known" and "Unknown" project types—a flexibility that is indispensable for defining and planning the scope of future-oriented projects that rely heavily on AI.

Relevance

The core meaning of these findings for the enterprise lies in that it clearly reveals the systemic bottlenecks faced by the enterprise in technological agile transformation, refined quantitative management, project risk control, and organization-level core asset accumulation. Transforming it into the business and operational language of the enterprise, it mainly reflects far-reaching impacts on the following four levels.

First, it directly concerns the protection of the enterprise's core profit margin and core assets, and can effectively eliminate losses and waste. The enterprise has strategically set a high profitability target of over 15%. However, because the development team frequently uses verbal agreements at the requirements front-end to approve the disconnect between prototypes and actual implementation, these undocumented customer intentions evolve into frequent rework and scope creep in the later stage. This invisible waste of workload and unplanned expenses are directly devouring the net profit of the enterprise. At the same time, the process assets accumulated by the enterprise in the past 5 to 6 years are too bloated and redundant, and employees fill out forms and check checklists for surface compliance, resulting in a large amount of formalism and waste of efficiency. Streamlining these process bloats can allow the enterprise to regain agility and allow employees to focus their energy on truly delivering business value.

Second, it can clear infrastructure obstacles for the enterprise's comprehensive layout of AI strategy and open up the transformation path. The enterprise's current governance mode shows a rigid, linear governance thinking. But when facing the disruptive technology of artificial intelligence, the enterprise lacks deep research and problem identification in the AI field before setting goals, which will lead to the decoupling of the enterprise's strategic direction from the technological frontier. In addition, the enterprise's existing institutionalized processes and infrastructure are only suitable for handling known types of traditional projects. When facing future-oriented unknown projects that highly depend on AI, the existing system cannot provide reasonable scope definition and strong support for them at all. The enterprise must upgrade the base, otherwise it cannot undertake high-difficulty new AI businesses.

Third, it helps to transform individual behavior of technology into organizational assets and precipitate organization-level core competitiveness. In emerging technology fields such as robotics and automation integration, although the enterprise has project needs, the current knowledge acquisition and competency cultivation completely belong to the spontaneous individual behavior of employees. Once these core employees resign, the enterprise's combat capability in this field will directly clear to zero. The organization level must establish a formal empowerment mechanism to transform individual competency into strategic organizational assets, and explore the people domain. In addition, although the enterprise has established process performance models, when these models were established and how they evolved completely lack records and configuration management. The lack of version control means that the enterprise cannot carry out long-term, cross-cycle quantitative improvement, and past data assets are extremely easy to become dead assets that no one can understand.

Fourth, it can significantly improve the underlying delivery quality, reduce the long-term maintenance cost of software, and eliminate code technical debt. Multiple instances of counter-intuitive complex logic and bad coding habits of directly negating return values without assigning values to temporary variables are observed in the code, which results in testing personnel being almost unable to carry out effective and isolated quality verification and validation during quality verification and confirmation. At the same time, such code that is extremely difficult to read and unmaintainable, and low-level spelling errors were not effectively intercepted in the peer review and quality assurance links. Once they flow into the production environment, it is extremely easy to introduce new defects during later maintenance, thereby multiplying the enterprise's long-term after-sales technical support and daily maintenance costs.

For enterprise management, these findings are definitely not simple minor problems of the R&D department, but a strategic warning at the executive level. The enterprise cannot use the old management mechanism that is traditional, bloated, and relies on verbal commitments to command a new type of battle facing the future, high profitability requirements, and relying on AI innovation. The enterprise urgently needs to initiate a fine-tuning and upgrade of lightweight agility and formalization of technical management at the organizational level.

Improvement Actions

Armed with the understanding of these core implications and systemic bottlenecks, the Engineering Process Group (EPG) must entirely break away from the traditional, rigid routines that led to surface-level compliance in the past. Instead, the next cycle of improvement work must pivot toward a lean path centered on being "lightweight, agile, and frontier-technology-oriented."

Pruning and Streamlining Process Assets to Restoring Operational Agility:

First and foremost, to address redundant, bloated process definitions and the lack of agility, the EPG's primary task is to initiate a thorough pruning and streamlining of organization-level process assets. Past practices introduced a massive volume of waste simply to check the box for CMMI compliance items. In the upcoming improvement cycle, the EPG must critically re-evaluate the process definitions, templates, checklists, and job aids created over the last 5 to 6 years. By stripping away redundant bureaucratic approvals and forms that yield no substantive delivery value, the organization can systematically eliminate Process Bloat. This initiative will not only relieve the development teams of excessive process burdens and realize lightweight project execution, but also ensure that projects maintain high agility while strictly adhering to baseline CMMI compliance standards.

I hereby authorize and agree that you and SITARA Technologies may share our appraisal results on SITARA's publishing channels and promote them as deemed appropriate by SITARA Technologies.

Hefei Xiaobu Intelligent Technology Co., LTD.

合肥小步智能科技有限公司

3401230398709

Sponsor: 李兵

June 17th 2026