

软件质量项目中需要避免的陷阱

部署太多的抽象指标

说到软件质量，很多文章都会提到为了避免项目因为技术债而崩溃，应该采用哪些比较好的方法以及开发步骤，如：识别存在高风险的组件，经常进行代码评审，重构复杂编码，检查编码规则，等等。但是，很少有指导手册是针对项目本身的：是的，为软件质量管理部署解决方案也应该被看作一个项目，要有自己独立的项目小组（首席信息官，开发人员，QA 经理，分包人）和时间表，而且和其它项目一样，一定要避免常见的陷阱。

这也是为什么我们决定在 Squaring 分享我们在软件质量管理工具项目中积累了一百多年的经验，讲述一些在开展软件质量管理项目的时候，必须避免一些陷阱。我们这里先讲第一个陷阱：部署太多抽象指标。

陷阱：部署太多抽象指标

这可能是最常见的陷阱，我们在过去的几十年里常常看到。和其它管理项目一样，指标是具有战略意义的，实现指标的过程是质量管理方针的体现。指标是主要的预期交付成果，对于软件开发生的管理项目而言，更是如此。

本文中的“陷阱”主要是指试图用一个神奇的综合指标来描述软件产品的质量，这种指标可能由各种各样相互矛盾的数据构建起来。简而言之，一些设计不良的指标显示在仪表盘上会非常难懂：没有人能够理解这些指标，甚至连设计者自己也可能无法理解。

当我们掉进这个陷阱时，我们常常会得到一个没有度量单位的无界值。比如：我的程序的整体质量为 128,可维护性为 72, 坚固性为 42, 等。这会让你想起些什么吗？你很快就会发现，在没有上下文的情况下，这些值是没什么大意义的。而且，它们还会引出一些尴尬的问题：

- 我们应该怎么看这个指标？值越高质量越好，还是值越低质量越好？
- 如果值是 132 而不是 128 的话，情况会严重多少？
- 这里表示坚固性的值可以和其它项目中坚固性的值相比较吗？
- 做什么努力可以将可维护性的值提高 10 点？

很快还会有很多其它问题来突显出指标设计不良的问题，这样在早期，质量管理项目就会受到冲击。

- 开发人员和管理人员存在相互理解和沟通的问题。
- 指标被误解的情况是不可以接受的。在某些情况下，这是要坚决杜绝的。
- 计算出这些指标的工具将受到质疑，进而被一些通俗易懂的指标取代：因为重复阅读的情况将会影响决策，并对由此产生的行动计划造成危害。

能够清楚地表达出所想的内容

设计质量指标是项目的关键环节，需要花时间和精力。指标必须明确、有说服力、无争议，以便质量项目能够被理解、被应用、被认可，并且不断得到改善。在确定指标之前，要明确它的主要作用：指标主要用来衡量实际情况和希望达到的目标之间的差距，并引导我们采取措施以实现质量目标。

比如，我的移动应用的一个基本目标就是确保其在开发阶段的可靠性：

- 我要保证代码易于测试。
 - 我还要检查代码是否已经经过了有效测试。
 - 我的可靠性指标主要依赖于以下数据：
 - 代码可测性度量（圈复杂度的方法，类之间的耦合，等）
 - 单元测试信息，比如 JaCoCo, Emma, Clover 等工具获得的覆盖率信息
 - 从功能测试活动中获取的数据
 - 为了更完整地显示软件可靠性等级，我还可以将负载测试工具获取的数据加进来。
- 实际上，如果我的应用无法支持上千个用户同时在线，就可能是由于产品性能和可扩展性的问题造成的，这会被视为缺乏可靠性。

高效质量指标的特点

如果不深入了解《NF X50 -171》或者《ISO 9001:2008》中的详细说明，我们可以直接认为如果指标能够符合以下要求，该指标即为高效的指标：

- **相关性和实用性：**指标是否能很好地指出产品的实际情况和质量目标之间的差距？
- **简明性：**好的指标只需要很简单的说明就可以解释清楚。说出要做的事，按所说的做，然后再检查。好的指标还必须能够明确指出，产品质量是否已达到目标要求。（参考 Victor Basili 的《目标问题度量》方法）。
- **具有代表性的指标，应该符合以下要求：**
 - **全面性：**每个级别的构件都需要有指标（产品组合，应用程序，程序包，等等）。实际上，利用能够自动进行数据收集的静态代码分析工具更容易实现这一要求。
 - **可量化：**比如，代码覆盖率度量，在执行过程中发现的所有错误，代码中的违规行为，新交付的代码行的总数，交付延迟的平均天数，等等。
 - **客观性：**指标的组成部分绝对不可以存在争议。测量软件构件大小的代码行数：要算上空白行吗？注释行呢？还是只包括编译指令？生成的代码也要算作软件的一部分吗？
- **可部署性：**在纸上起草指标是一回事，实际部署并用真实数据展现这些指标又是另一回事。实际运用中，常常会出现数据的格式不正确，数据分散在不同的工具中，甚至有些构件的数据根本不存在的状况。这就要求先确定指标的可用性和可扩展性。

总的来说，指标是活的：它会经历出生，发展，死亡的过程。如果某个指标含糊不清或者在实际情况下没什么作用的话，它就会“死”。评估指标时，要将上面详细说明的那些特点都考虑在内，对于作用不大的指标，要立刻剔除，这叫做自然选择！越早剔除越好，因为要剔除那些已经部署在整个项目组中的指标会更加困难。否则，你就要向项目小组成员解释，为什么一开始没能设计出一个好的指标，相信这会是非常棘手的事情。

以技术债（Technical Debt）为基础的指标：具有相关性，可以直接使用，便于部署

署

在开始一个软件质量管理项目时，先使用技术债指标是非常好的选择。技术债最先是沃德·坎宁安（Ward Cunningham）在 90 年代提出的概念，是指那些开发团队（情愿或不情愿）推迟

下一冲刺 (Sprint) 或产品发布而增加的任务 (功能, 问题修复, 代码重构, 结构优化, 等)。技术债可以立刻给团队喘息的时间, 但是这和其它借贷的性质一样, 还得越晚, 需要偿还的利息就越多。换言之, 如果某个功能可以在今天实现, 那么花费是 100, 但是如果是在明天完成, 则需要花费 100+, 如果是在一周之后或者更长时间内完成, 那么花费也会更多。Philippe Kruchten (加拿大英属哥伦比亚大学的教授) 最近用下面的模型解释了这个比喻, 他将技术债定位为一种消极的无形元素, 认为它会贬低软件的价值:

	可见	不可见
积极作用	功能	结构
消极作用	问题	技术债

© Philippe Kruchten

很多 IT 公司 (如 Squoring 或使用 SQALE 方法的 Inspearit) 已经开始实行这一概念, 他们为开发团队提供工具, 以识别并量化应用程序中的技术债, 一般以天为单位。因为有了这些度量, 技术债指标就成了引导软件开发项目的质量的有效工具。

- **有用性:** 技术债侧重于团队的敏捷性和创新能力。高技术债意味着在故障检修方面做了很多工作, 这就损害了新功能交付的附加值。技术债指标会突出软件构件在产品质量要求 (坚固性, 安全性, 可靠性, 可维护性, 等) 方面与目标的差距, 以及在项目中发现的不合格的地方。
- **易于理解:** 当技术债以 UOW (时间, 钱……) 形式显示的时候, DevOps 从首席信息官到开发人员, 都能够很清楚地理解它的意义。由于任务延期或产品交付时质量未达到预期标准而产生的工作量, 都需要计为应用程序、项目或构件的技术债。比如: 我们需要安排 10 天的人力来使这个项目达到质量模型中规定的可靠性要求。
- **易于部署:** 简单地说, 技术债主要是计算因为不合格而增加的工作量。所以, 一开始尝试度量项目的技术债时, 可以直接在 Excel 表中完成。
- **技术和管理视角的协调:** 如果项目小组的成员获得的信息是有意义的、可理解的、不是模棱两可的, 那么他们就可以直接用一个通用框架来解决实际的软件质量问题, 而不需要讨论某个指标本身的含义。
- **以任务为导向、具体化:** 这些通常是技术债指标的重要优势。通过估算项目 (代码, 文档, 需求等) 中发现的不合格的地方, 就可以非常轻松地列出一些必要的措施, 通过采用高效的、具体化的方法来减少技术债。

管理技术债的 SQUORE 指标

Squoring 技术建立在我们的顾问指导的软件质量管理项目的基础之上, 它定义并打包了技术债指标。

这些可从 Squore 仪表盘获取的指标有现成的质量模型, 所以能够很方便地应用于项目环境中。利用这个方案, 几分钟之内, 您基本上就可以进行高效的软件质量管理了。



除了上面所讲的显示开展补救工作而产生的工作量的形式, Squore 计算的技术债还能够以不同颜色的 (绿色, 黄色, 橙色, 红色……) 字母分级 (A, B, C, D, E……) 的形式显示, 这样可以很清楚地表示出所分析的构件的质量风险等级。

利用 Squore 指标, 无论应用程序是处于开发还是维护阶段, 都可以很方便地对该程序的质量进行客观监控:

- Squore 指标是一个非常容易理解的等级系统 (有 A, B, C, D, E, F 等级), 而且能够提供准确的信息 (计算人力或货币的

技术债)。

- 得益于从质量特性深入到软件组件（项目，软件包，算法，函数，等）的独特的向下钻取功能，Sqoure 指标能够快速突显出项目组合中存在的风险和质量问题。
- Sqoure 指标并不是只考虑代码收集数据（如：项目的功能需求，文档质量等）。
- Sqoure 指标能够根据标准或项目惯例中的质量需求，生成任务计划，优化项目以有效地减少项目中的技术债。

创提信息科技（上海）有限公司 – Trinity Technologies

专注于嵌入式软件研发质量和自动化测试的方案和咨询服务，提供覆盖软件测试整个流程的完整的解决方案，包括从研发前期的代码级测试到后期的系统级测试，从静态分析到动态测试，从编码检查，单元测试、集成测试到性能测试和测试覆盖率分析等。

公司通过专业的自动化工具（如 DT10, VectorCAST, PRQA, SQUORE 等）和服务满足不同客户对软件质量和测试的需求，持续协助客户改进软件研发质量和效率。客户主要集中在高安全和高可靠性领域，如国防和航空航天、轨道交通、汽车电子、医疗器械、工业控制、通讯和电力电子等行业。公司提供的领先的解决方案不仅为数以百计的客户提高产品质量，还协助客户遵循高安全和高可靠性行业的合规性要求，如 DO-178B/C, IEC61508, EN50128, ISO26262, IEC62304 和 MISRA 等行业标准，并获得相关机构认可和认证。

版权声明：本文档版权归创提信息科技（上海）有限公司所有，并保留一切权利。