



# 借助 IBM Application Discovery 提高软件质量

## 3 步式方法

---

### 目录：

- 1 软件质量管理是什么？
  - 3 在组织内运用 3 步式方法嵌入软件质量管理
  - 4 第 1 步 - 解锁
  - 5 第 2 步 - 赋权
  - 6 第 3 步 - 评估
  - 7 建议
- 

通过软件质量管理提升业务价值，是当今许多大型组织的当务之急。

一直以来，许多质量保证计划和流程旨在满足特定法规或合规义务，而不是帮助 IT 满足成本、服务和敏捷性方面的业务目标。

低劣的软件质量 - 有的一开始便是如此，有的是随着时间的推移质量日渐下滑 - 是技术债务的关键推手，而在当今充满挑战的经济时代，降低这种债务水平是多数组织的重要目标。

要在应用程序所支持的实际业务环境下了解并管理软件应用程序及其质量，必须深入了解所有组件的完整结构。分析它们互相连接的方式以执行端到端业务流程，有助于确保在整个 IT 环境中实现情景软件质量。

IBM® Application Discovery and Delivery Intelligence 3 步式方法从实际的角度考量必须采取的措施，以及通过实施软件质量管理可以获得的成效。

### 软件质量管理是什么？

随着支持流程和实现流程自动化的工具取得进步，实施起来更实惠、实用，业务软件应用程序的质量管理方式正在迅速从“主观性”



转向“客观性”。即便如此，单纯地分析和评估单个代码组件，尽管对它本身而言可以提高开发工作的质量，但并不一定能从本质上交付以下方面的业务价值：

- 服务成本
- 运营绩效
- 服务交付
- 易维护性

除此之外，组织的技术债务水平也有部分是软件质量造成的，因为为了确保应用程序软件始终处于最新状态而产生的累积成本会因为低下的质量和较高的复杂性而显著增加。

提升软件质量意味着跨整个 IT 价值链进行评估、改进和管理——实际情况是多个应用程序套件服务于端到端业务流程。

考虑典型的业务流程；在当今世界，它可能从以下几个方面跨越 IT 界限：

- 软件包
- 应用程序语言
- 技术环境

它还有可能跨越由复杂接口或集成软件层和复杂映射所实现的种种界限。举例来说，在银行业环境下，使用 Java 语言写成的客户前端可能会链接到大型机上的信用处理（以 COBOL 语言写成），后者继而会链接到核心银行业务包（如图 1 所示）中：

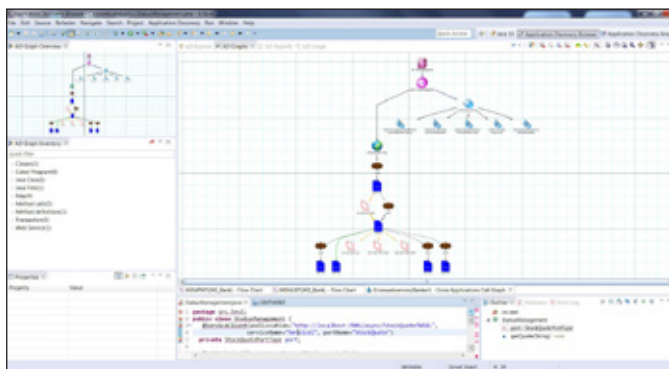


图 1：Application Discovery 银行应用程序跨应用程序调用图

---

应用程序软件质量是高质量代码组件与高质量交互和相互依赖性的结合体，可以形成一个正常运行的整体。

一旦代码组件嵌入到这个整体中，很少有 IT 开发团队能够确保一致性和质量。

在集成测试之前，有近 80% 的质量问题都不会浮出水面，而在这时去纠正问题通常成本高昂、耗费时间。

---

运用一致的质量指标并跨不同应用程序、组件和环境管理这些指标，是一项真正的挑战。实际上，这是支持转向敏捷开发方法所需的必要条件，这种方法专注于压缩交付时间并改善最终结果的品质，需要谨慎细致的管理。

通过在 IT 组织内部嵌入流程和自动化的集成工具，可以验证质量标准是由 IT 人员在整个应用程序生命周期内定义、应用和管理的，并在待应用软件质量管理的使用寿命内在适当的时间提供适当的信息。

---

“CIO 和 IT 管理团队需要与企业其他人员就如何维持重要应用程序资产的完整性展开新型对话”<sup>1</sup>

---

IT 角色	应用软件质量管理的重点领域
架构	架构标准、应用程序架构、集成架构技术平台、升级计划、IT 战略
设计	应用程序设计、影响评估、编码标准、集成、通用服务设计
开发	编码标准、代码质量、集成、单元测试、故障修复、增强
测试	代码范围集成测试、UAT、回归测试
支持	应用程序维护、故障分析/解决、应用程序性能监控
管理	应用程序产品组合管理、变更影响分析、服务管理、计划/项目管理

在整个生命周期内提升软件质量，有助于确保：

- 尽早发现并解决软件质量问题
- 通过复杂性和成本等重要指标监控软件质量
- 长期跟踪改进
- 按照高水平质量管理外包交付
- 交付的应用程序功能强大且适用目的
- 维护和增强的速度加快、成本降低
- 了解技术债务，并进而可以减少技术债务

这样一来，IT 组织可以利用应用程序开发和维护预算来交付以下几个方面的有形业务价值（通常是总 IT 预算的一半左右）：

- 降低应用程序中断、合规问题、安全威胁造成的潜在风险
- 为向敏捷开发流程转变提供支持
- 将宝贵资源重新分配给可创造业务价值的应用程序增强功能，而不只是用于应用程序维护和故障修复
- 最大限度利用现有应用程序资产

“优化应用程序开发和维护，可节省 50% 以上的成本”<sup>2</sup>

### 在组织内运用 3 步式方法嵌入软件质量管理

要在 IT 组织内成功实现软件质量管理流程和工具的制度化，需要采用易于理解、一致的方法，并为每个步骤使用相同工具集 — 为库存、结构质量和改善长期提供“单一版本的事实”。这应当基于对实际代码组件进行的客观分析，而不是对设计信息进行的主观评估，设计信息往往已经过时，或者不能为应用质量标准的开发人员提供足够深入的信息。



图 2：适用于软件质量管理的 IBM Application Discovery 3 步式方法

许多组织中缺少详细的库存和结构信息，可能导致 IT 管理团队对软件质量对成本、资源和服务的影响知之甚少。

在对变更进行深入分析的基础上，为了实现改善，可以建立变更基准并积极改进整个应用程序环境的架构和结构质量，从而实现更高质量的交付。

后续的软件质量管理离不开提供长期跟踪、定义明确的重要指标的精确信息，以验证(a)改进是否实现，以及 (b) 是否为组织创造了有形价值。

十几年来，许多组织里的应用程序维护预算一直吃紧，这导致应用程序维护和改进活动滞后，而意识到提高端到端软件质量可以有效克服这些挑战就意味着，流程各阶段准确易用的信息比从前任何时候都重要，并使用自动化分析和报告工具来奠定基础。

“用户管理、通过静态和动态分析了解架构和代码质量的需求，以及评估内部和外部资源性能的有效指标，都在加快自动化软件质量的普及进程”<sup>3</sup>

### 第 1 步 – 解锁

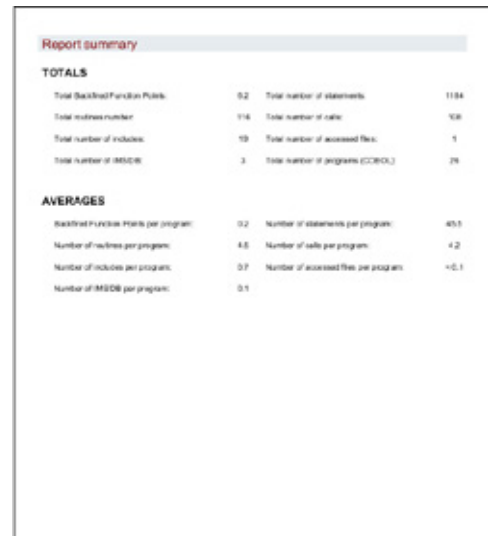
在许多大型组织内部，对应用程序软件组合存在不同的观点，具体取决于您的谈话对象。这在很大程度上受他们思维方式的影响；开发商关注的是代码组件，架构师关注的是整体环境/标准，项目经理关注的是成本和资源，而最为重要的业务用户关注的则是提供的功能性和服务水平。

制定整个应用程序软件产品组合的通用质量基准，最好的结果是有意义，而最坏的结果是成为 IT 管理人员的噩梦，除非您拥有：

- 明确定义的流程
- 普遍认同的指标体系
- 自动收集数据的集成工具套件（并保持最新状态）
- 易于使用的报告/跟踪机制

如果没有在第一阶段解锁应用程序产品组合（即库存），则几乎不可能实施战略性应用程序生命周期规划和产品组合管理。这应包括：

- 所有技术平台和语言
- 所有应用程序组件
- 所有相互依赖性



Report summary			
<b>TOTALS</b>			
Total Backlist Function Points	52	Total number of statements	1184
Total modules number	114	Total number of calls	108
Total number of includes	19	Total number of accessed files	1
Total number of MSDs	3	Total number of programs (CCRO)	29
<b>AVERAGES</b>			
Backlist Function Points per program	0.2	Number of statements per program	40.5
Number of modules per program	4.6	Number of calls per program	4.2
Number of includes per program	0.7	Number of accessed files per program	<0.1
Number of MSDs per program	0.1		

图 3：Application Discovery 库存报告

它应源自于实际部署的、并能反映真实情况的应用程序源代码，而不是可能不完整且已过时的设计文档。

维护员工花费 47% 的时间了解代码<sup>4</sup>

第二阶段“评估”是基于评估的下一层次信息。这是利用行业标准或自定义指标对应用程序及其组件所做的更加深入的分析，也是应用程序软件质量的首个视图。通常，它会包括：

- 复杂性（例如功能点、McCabe、易维护性指标、启发性）
- 质量（例如无用代码，故障作业）
- 成本（例如每周开发成本，计划成本与实际成本）

随后可以从整体上了解应用程序软件质量，并通过以下方式优化待改进的领域：

- 简化
- 增强
- 升级
- 现代化
- 替换
- 云迁移

它还应支持在成本、时间和资源方面估算变更影响。

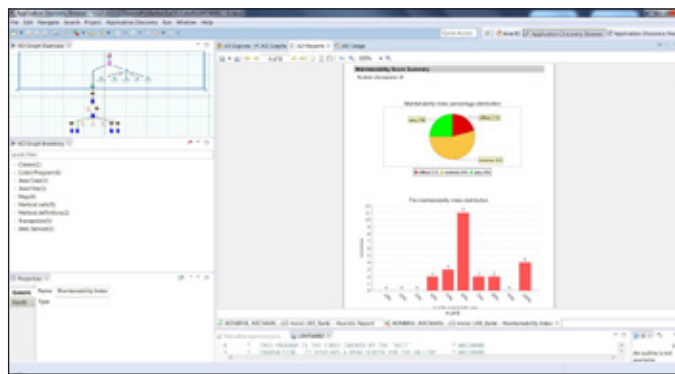


图 4: Application Discovery 易维护性评估

## 第 2 步 - 赋权

要给开发和测试团队赋权，让他们可以确定分别和以整个应用程序软件产品组合形式改进应用程序的最佳方式，并进而在开发环境中应用这些改进，就需要规范、准确的分析信息和自动化工具，以便应对跨应用程序平台、语言和版本的变更影响复杂性。开发和测试团队需要一个通用存储库，从中获得跨开发、测试和生产环境的应用程序代码组件持续更新视图，以及各工作职能部门基于角色的视图。

使用通用工具包使开发人员能够始终符合组织特定的：

- 架构限制，尤其是在大型核心系统中
- 编码标准
- 通用服务和接口机制应用
- 一致的数据应用，包括更新机制
- 符合法规

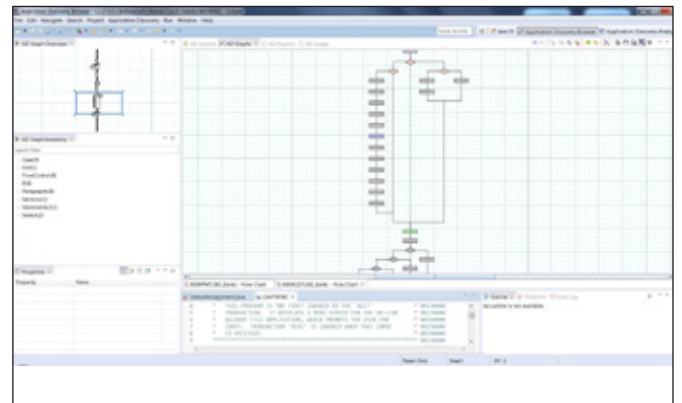


图 5: Application Discovery 浏览器

由于能够评估影响并执行“用途”分析，开发人员可以集中精力处理受影响的组件，并降低结构缺陷仅在集成测试时显现的潜在风险。

由于能够了解哪些应用程序组件受到变更影响，测试人员可以制定更加全面的测试计划，而代码覆盖率数据分析则能使测试人员优化这些计划。

技术团队可以跨技术平台交换信息，并改进应用程序软件。这为集成、基于风险的回归和用户验收测试提供了更为准确的基础，得益于变更可被并行执行并被打包发布。

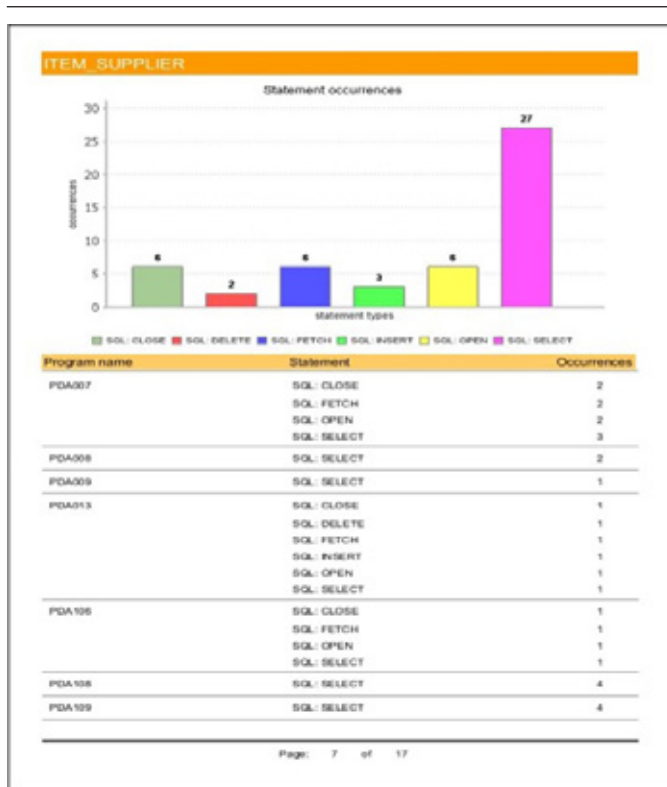


图 6：“用途”分析

基于变更的代码分析自动更新还可用于跨开发、测试和生产环境快速获得质量和一致性指标。对于那些将核心应用程序维护和增强活动外包给单个或多个供应商的组织而言，上述功能对于在这些组织中保持质量特别有用。

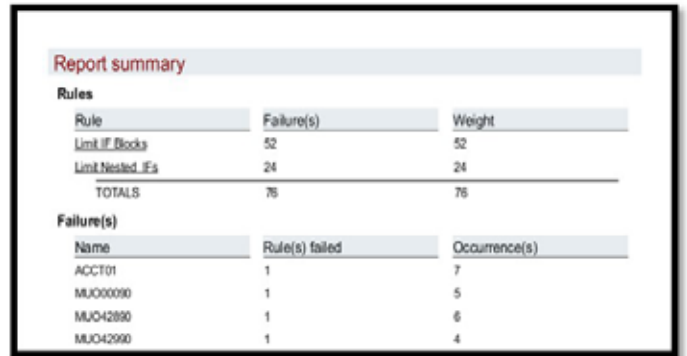


图 7：使用编码规则监控合规性

它使作为应用程序所有者的 IT 组织能够确保外包服务提供商：

- 遵从架构标准
- 遵从编码标准
- 致力于提升质量

实现这些工具和技术的制度化，将有助于 IT 在当前交付“软件资产”，而不是给明天留下麻烦。

应用程序中的结构质量问题常常就是造成中断、缺陷、性能不佳和安全隐患的根本原因。

没有深入的自动化了解和影响分析功能，几乎不可能发现这些问题。



### 第 3 步 – 评估

评估应用程序软件质量不再依赖于对应用程序代码组件有深入技术了解的 IT 管理人员。它还可以减少对过时或者不完整设计文档的依赖，杜绝发生人员流动导致所有应用程序知识流失的最糟情况。

在解锁和赋权步骤中得到的单个分析结果存储库的基础上，利用自动化分析报告工具，各级 IT 管理人员能够了解这些改进如何创造有形业务价值。每当对基础代码进行质量改进时，上述分析结果会自动更新。

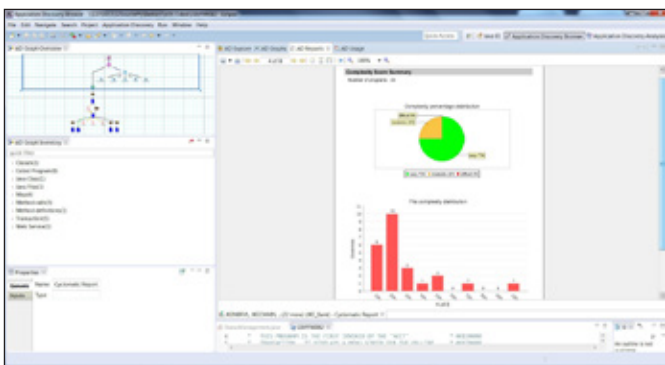


图 8: Application Discovery 自动报告

通过整合并关联与质量和复杂性相关的重要应用程序软件指标以及操作和代码率覆盖指标，例如：

- 使用频率
- 资源利用率，以及
- 性能

可以更好地了解当前的状况，以及 IT 组织在长期是否有所改进。它会推动：

- 新应用程序项目和计划的投资优先级
- 更高水平的项目时间、成本和资源确定性
- 释放 IT 维护预算，用于业务提升
- 企业和 IT 围绕交付的价值更好地开展对话

毋庸置疑，应用程序软件质量管理是一个长期的过程，需要维持一种常态。如果组织实施成熟的管理流程，而且这种流程得到了强大的专用应用程序软件质量管理工具的支持，那么这无疑会给他们的业务组织创造价值。

**长期积极主动地监控和管理质量改进，对于 IT 管理人员而言是一项重要的流程，有助于减少应用程序返工、缩短上市时间并更准确地预测性能。**

### 建议

无论造成组织应用程序软件质量低下的原因是什么，显而易见的一点就是需要高昂的成本来解决问题。

质量薄弱主要由两个原因造成：

- 违反或缺乏良好的架构和编码标准，导致结构缺陷
- 从各版本遗留下来的现有代码问题，导致应用程序不能与新应用程序正确集成，或者出现从前暴露的缺陷

事后看来，通过实施强大的软件质量管理流程，以及让您能够纠正以往错误并注意不再重蹈覆辙的支持工具包，许多以往错误和权宜性决策很可能不会重现。

立即采取行动，利用 IBM Application Discovery 3 步式方法，改善您的软件质量管理，并大幅减少技术债务以及未来的开发和维护成本：

1. 解锁您的应用程序软件产品组合
  - a. 通过了解所有应用程序组件和相互依赖性建立基准
  - b. 设定改进优先级
  - c. 设定改进目标并制定指标
  - d. 计划改进并诚实评估成本

2. 为您的开发和测试团队赋能，以提升应用程序软件的质量
  - a. 在可能的地方进行简化和现代化
  - b. 转变您无能为力的领域或有更好解决方案的领域
3. 持续评估软件质量
  - a. 长期跟踪/监控改进
  - b. 根据业务需求调整并运用指标
  - c. 了解整个应用程序软件产品组合，而不只是点代码

高质量软件不仅仅局限于高质量代码。

使用自动化工具和技术评估、改进并最终管理软件质量，可降低多达 50% 的应用程序开发和维护成本。<sup>5</sup>

这种 3 步式软件质量管理方法旨在由 IBM Application Discovery Solution（如图 9 所示）自动实施并提供支持。

使用 IBM Application Discovery 为软件质量管理提供支持，可为客户的应用程序维护和开发活动节省 30% 以上的成本。<sup>6</sup>



图 9：IBM Application Discovery 解决方案软件质量管理 3 步式方法

### z Systems 简介

如要了解更多有关 IBM z13 (z13) 的信息，请联系 IBM 代表或 IBM 业务合作伙伴，或者访问以下网址：[ibm.com/systems/z13](http://ibm.com/systems/z13)

此外，IBM 全球融资部可以帮助您以最经济高效和最具策略性的方式获得您企业所需的 IT 解决方案。对于符合信用要求的客户，我们可以定制最适合您业务需求的 IT 融资解决方案，实现高效的现金管理，并降低您的总拥有成本。IBM 全球融资部是您为关键 IT 投资筹集资金、推动您企业向前发展的最明智选择。如需更多信息，请访问：[ibm.com/financing](http://ibm.com/financing)





© Copyright IBM Corporation 2016

IBM Systems and Technology Group  
Route 100  
Somers, NY 10589

美国印制  
2016 年 10 月

IBM、IBM 徽标、ibm.com 和 z Systems 是 International Business Machines Corp. 在全球多个司法辖区注册的注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。IBM 商标的最新列表可在下述网页的“版权和商标信息”中查看：[ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Java 和所有基于 Java 的商标与徽标均为 Oracle 和/或其关联公司的商标或注册商标。

本文档为初始发布日时的最新文档，IBM 可能随时对其进行更改。IBM 并未在每个开展业务的国家/地区提供所有产品/服务。

本文档中的信息“按原样”提供，不带任何明示或暗示的保证，包括不带任何适销性、对特定用途的适用性的保证，以及任何不侵权的保证或条件。

IBM 根据提供产品时的协议条款与条件提供产品担保。

所有关于 IBM 未来方向和意向的声明都可随时更改或收回，恕不另行通知。

良好安全实践声明：IT 系统安全包括通过防范、检测和响应来自企业内部和外部的不正当访问，从而保护系统和信息。不正当访问可导致信息被更改、销毁或盗用或导致系统被破坏或滥用，包括攻击其他系统。没有任何 IT 系统或产品是完全安全的，而且在防范不正当访问方面，也没有任何单个产品或安全措施是完全有效的。IBM 系统和产品的设计旨在作为全面安全方案的组成部分，其中必然涉及其他操作程序，可能会要求其他系统、产品或服务具有最高的效率。IBM 不保证其系统和产品可免受任何一方的恶意或非法行为影响。



请回收利用

1 Kyte, Andy, “*Measure and Manage Your IT Debt*”, Gartner, 资料来源：RAS Core Research Note G00205265, 2010 年 8 月 9 日；[http://imagesrv.gartner.com/media-products/pdf/cast\\_software/gartner2.pdf](http://imagesrv.gartner.com/media-products/pdf/cast_software/gartner2.pdf)

2 “*Gartner Says Optimizing Application Development and Maintenance Can Cut Costs by More Than 50 Percent*”, Gartner 2014 年应用程序、架构、开发和集成峰会，5 月 19 日至 20 日英国伦敦和 7 月 21 日至 22 日澳大利亚悉尼。<http://www.gartner.com/newsroom/id/2711017>

3 Worldwide Software Quality Analysis and Measurement 2015-2018 Forecast and 2013 Vendor Shares: Code Visibility Enables Quality with Security to Drive Market Growth, Global Information Premium Market Research Reports, 2015 年 1 月 16 日，<https://www.giiresearch.com/report/id220931-worldwide-software-quality-analysis-measurement.html>

4 Mary Poppendieck 和 Tom Poppendieck, “*Implementing Lean Software Development: From Concept to Cash*”, 9/17/2006。

5 Gartner 同上，2014 年。

6 基于 ezSource（现在的 z Systems）开展的调查的百分比数。