

白皮书 | 2015 年 9 月

# 测试数据 管理

为何可能成为推动持续交付的唯一方法

Huw Price  
CA Technologies



## 目录

---

持续交付简介	3
糟糕的 TDM 会妨碍持续交付	3
基于需求的 TDM 方法	5
摘要	7
参考	8
CA Technologies 的优势	8
关于作者	9

---

## 第 1 部分

# 持续交付简介

“持续交付”在某种程度上已成为软件开发领域的流行词语。因此，众多供应商承诺可以将其变为现实，提供其工具作为项目延迟及失败的传统原因的补救措施。他们提议，通过采用这些工具，组织能够及时并在预算内持续创新和交付优质的软件。

持续交付的吸引力是可以理解的。在如今的现代应用程序经济中，组织依靠软件向客户提供价值。因此，业务和 IT 需要更加紧密地契合，公司在市场中的地位则取决于日常向客户提供价值的的能力。IT 需要能够迅速响应不断变化的市场和客户期望，开发可满足不断变化的业务关键需求的软件，同时减少测试成本的时间。

尽管许多工具声称可推动持续交付，但它们主要属于后勤方面，并且仅能在开发生命周期晚期部署。DevOps 仅以运营的形式出现，而工具使用已设计的软件，然后支持开发和回归测试。从需求收集阶段开始早期出现的系统性问题一直存在，并且导致代价高昂的瓶颈和项目延迟，使持续交付的成功实施成为泡影。

尤其是，针对测试用途的测试数据管理 (TDM) 或者在正确的时间将正确的数据传送到正确的位置的重要性被忽视了。糟糕的测试案例设计以及低品质数据的无效率部署意味着测试团队缺少对系统进行完整测试所需的数据。为了及时并在预算内交付软件，质量将受到影响。

## 第 2 部分

# 糟糕的 TDM 会妨碍持续交付

对大多数组织和供应商而言，TDM 开始和结束于复制、屏蔽，还可能包括构建生产数据子集。这些 TDM 策略完全属于后勤方面，仅专注于转移数据，而数据会在设计项目需求时被忽视掉。生产数据经复制并迁移到开发、测试或 QA 环境后，这些数据通常被称为“黄金副本”——即一组完美的测试数据，测试人员可通过这些数据执行对系统进行完整测试所需的任何测试。

屏蔽和构建子集是解决基础架构成本与合规性的传统痛点的良好开端和一部分。但是，构建生产数据库的子集和屏蔽生产数据库自身会带来使用生产数据本身固有的许多问题。这些问题使持续交付的成功实施化为泡影。

## 浪费的时间

大多数组织都缺少中央 TDM 团队或配置服务，以至于必须由本地团队查找或创建数据。测试人员可能会花费最多 50% 的时间来查找数据，而多达 20% 的总 SDLC 被用在等待这些数据。这样会导致测试瓶颈，使真正的敏捷和持续交付化为泡影。在一个实例中，Grid-Tools（现在是 CA Technologies 的一部分）遇到一个团队，他们理应参与三周的“冲刺”，却必须花费四周来准备用于冲刺的数据。

屏蔽或构建生产数据子集使测试团队必须实际行动并寻找要使用的数据。这是一项费时费力的任务，不受控制的电子表格中不一致的数据存储使情况恶化。例如，可以将个人可标识信息 (PII) 添加到“注

释”列，用于信用卡有三列而某客户碰巧有四列的情况。而且，这些数据必须保持引用完整性。数据越复杂，实现这一目标就越困难，而花费在屏蔽数据上的时间和精力往往会超过测试人员从头开始可能花费的时间和精力。

从头开始手动创建数据可能同样耗时。此外，由于通常会基于特定测试案例创建数据，数据会迅速变得过时和不相关，因为现实世界总是在变化。普遍给出的例子是交易图和货币汇率，数据每天都会变得过时。发生这种情况时，必须更新数据 — 又一个耗时的延迟 — 或者，更通常的做法是“销毁”。这类似于发生版本变更的情况，或者有人请求数据刷新，或者每当虚拟环境变化时必须创建数据。

几乎不可能在团队之间共享和重用手动创建的数据。无法利用在生成数据时花费的早期努力会造成额外的工作，并且强迫组织在减少成本及上市时间与交付具有所有必要功能的宝贵软件应用程序之间做出直接而不可取的选择。

## 依赖性限制

等待配置数据或数据变得可用会浪费测试人员大量的时间。组织和供应商通常会将 SDLC 视为一系列线性阶段，一个团队完成某些工作，然后传递给下一个团队。因此，团队必须等待来自上游团队的“适合”数据变得可用，并且可能由于其他团队正在进行工作而无法使用所需的数据。

这与处于持续交付核心的敏捷并行开发原则形成对照，在后者中，个体时间的每个时刻及创新的能力都会被用于实现最大的价值。此外，由一个团队所做的更改可能会以这样的方式对数据产生影响：当其他团队使用数据时，其应用程序测试将明显无端失败。

如果团队必须手动创建数据，或者等待数天或数周以获得数据，您就无法指望他们迅速响应不断变化的需求以及交付经完整测试的软件。

## 糟糕的质量

在非生产环境中使用生产数据的真正问题是质量。生产数据通常仅提供 10-20% 的功能覆盖率，因此任何抽样方法都不太可能提供满足构建新子系统所需的所有测试案例的数据。大量的数据都非常相似，提取自常见或“一切如常”事务，并经其本质消毒以排除将破坏系统的不良数据。因此，测试倾向于更多地关注“愉快路径”，而较少关注非功能和负面测试。

然而，负面测试应构成测试的 80% 左右，因为正是这些异常和边界情景造成系统崩溃。只要测试专注于愉快路径，缺陷就会总是将其带入生产，导致返工、关键延迟、成本螺旋形上升以及潜在的项目失败。行业研究显示，在测试期间修复缺陷所需的时间比在需求阶段找到缺陷的时间多 50 倍<sup>1</sup>，而这样的延迟和无法响应变化使持续交付化为泡影。

生产数据的功能覆盖率可以由手动创建来补充，但这是不精确、耗时且不科学的方法。没有任何方法能够验证是否实现最大覆盖率，也没有任何方法能够了解数据是否保持参考完整性。

## 高昂的成本

最后，除了返工和延迟交付造成的高昂成本，生产数据的实际复制也是过于缓慢而昂贵的过程。报告显示，某些组织发现自己拥有单个数据库的多达 20 个副本，带来高昂的硬件、许可和支持成本支出。

此外，大容量数据存储可能非常昂贵。一份相关第三方数据存储研究指出，将数据移动到包括云在内的低成本层会暂时降低存储成本，但不会解决或应对数据增长的基本问题。换句话说，任何以迁移数据、构建数据子集和屏蔽数据告终的 TDM 策略都无法应对如何以实惠的方式存储由日益复杂、现代、复合的应用程序处理的数据的问题。

此外还有代价更高的不合规危险，即使屏蔽生产数据也不会得到解决。一份独立研究揭示，2014 年的数据违规成本在平均数据违规成本方面增加了 15%，达到每份记录 350 万美元，而将在 2016 年强制实施的新的欧盟 (EU) 数据保护指令将导致因数据违规而征收的最高罚金达到 1 亿欧元<sup>2</sup>或全球营业额的 5% — 以较高者为准。

屏蔽生产数据不会保证合规性，因为真正的危险仍然是人为错误，其中 50%<sup>3</sup> 的数据违规与内部人员行为有关。此外，屏蔽时，数据的参考完整性必须保持完好，否则应用程序测试可能会失败。但是，数据越复杂，破解就越容易，因为更多的信息片段可以被关联。

避免此类高昂的基础架构成本并确保不泄露敏感数据的唯一真实方式是不使用生产数据。尽管这样的成本并非想要实施持续交付框架的组织所独有，但它们当然会与其原则形成对比。它们会阻止组织在预算内交付迅速履行不断变化的业务需求的优质软件。

---

### 第 3 部分

## 基于需求的 TDM 方法

想要实施持续交付的组织必须重新思考其测试和开发流程，而不是简单地重新安排。他们需要重新考虑通往 TDM 的方法。通过由需求推动的通往 TDM 的完整端到端方法，组织能够执行左移测试、缓解风险并将缺陷的产生降到最低限度，从而以更少的成本更快地交付优质软件。

与其以测试案例为基础思考测试案例上的数据，组织应在设计决策方面思考数据 — 以需求本身的形式，通过直接链接到测试案例的数据来设计测试案例。为需求定制测试数据可确保这些数据“适合”，同时，将数据迅速提供给测试团队的能力意味着他们可以迅速响应不断变化的业务需求。

在这一点上，持续交付无法在 SDLC 晚期开始：它必须始于一个创意 — 来自需求本身 — 以便测试和开发团队能够迅速响应变化的业务需求，自始至终从这个不断变化的创意开始工作。团队需要能够轻松验证和确认需求，派生出直接从需求链接到预期结果和虚拟数据的测试案例。此类基于模型的测试正是“左移”和持续交付软件的正确含义 — 将开发生命周期的所有工作压缩到需求收集阶段，即使需求发生变更，也能从该阶段轻松产生所有后续工作。

## 建立更好的需求

需求本身需要包含关于测试所需系统的所有定性信息，以便测试人员能够从最初“创意”派生出使用案例和测试案例，并在其之间引入可追溯性。需求发生变化时，如果测试人员要迅速更新其测试，这样的可追溯性很有必要。

凭借适当的工具，不管设计流程本身的简易性如何，流程图建模都能生成包含关于测试所需系统的所有定性信息的流程图。CA Test Case Optimizer（以前称为 Grid-Tools Agile Designer）通过自动派生具有最大覆盖率的最少测试案例所需的所有功能性逻辑来支持流程图，而不会像因果关系建模或成对方法可能的那样让业务“傻等”<sup>2</sup>。

此外，流程图可增加需求明确而完整的可能性。它们打破了不相干的“文字墙”和累赘的图表，后者通常细分为可做摘要的小块。这些流程反映系统的因果关系逻辑，实际上组成一系列“假如...那么”(what if, then this) 语句。

这样不仅帮助减少 56%<sup>4</sup> 从需求的歧义中浮现的缺陷，还强制需求团队在系统建模方面进行思考 — 在限制、约束和边界条件方面。通过此类需求建模为流程图，然后可以确定经过系统的每条可能路径。然后，可以派生出提供 100% 功能覆盖率的测试案例，以便测试覆盖经过系统的每条可能路径，包括消极路径和意外结果<sup>3</sup>。

凭借活动流程图，可以进一步将这些测试案例链接到复杂性指标、虚拟数据、测试数据、自动化脚本、意外结果和积压的工作，将 SDLC 的努力集中到需求收集阶段。

## 提供“适合的”数据

通过 CA Test Case Optimizer 等工具将测试案例设计流程自动化，测试人员能够生成以最佳方式测试需求所需并令其满意的正确测试案例。然后，这为我们引出了本白皮书的中心主张：持续交付需要更好的 TDM。测试人员需要访问能够帮助其实现 100% 测试案例覆盖率的“适合的”数据，在正确时间将这些数据提供到正确的位置。

## 质量

如前所述，生产数据无法提供对系统进行完整测试所需的覆盖率。相比之下，以合成方式生成测试数据能够产生覆盖所有可能情景的小而丰富的数据集，及时事件之前未发生过也是如此。每个真实世界情景可被视为另一个数据点，因此，即使是新的和即将来临的情景，也能为其创建合成数据。CA Test Data Manager（以前称为 CA Data Finder 或 Grid-Tools Data Maker）使用智能数据事件探查技术获取数据模型的精确图像，生成丰富而复杂的数据，提供以这些数据为基础的 100% 的功能性变化。

通过基于需求的方法，生成的数据匹配到测试案例，确保其适合满足每个单独测试人员的需求。这样的“测试案例数据生成”使测试团队更有可能第一次就找到缺陷，从而避免使持续交付化为泡影的耗时的返工。

## 时间

如前所述，手动创建或操作数据或者等待数据变为可用所花费的时间与持续交付的原则形成对比。



相比之下，CA Test Data Manager 等用于数据创建的自动化功能能够直接用于 RDBM 或 ERP API 层，允许测试人员在其处理能力允许范围内尽快生成数据。膨胀的脚本可使组织拥有的数据量加倍，与数据库基础架构能够处理这些数据的速度相同。这样确保“适合的数据”在数小时内出现，而非数周，以便测试人员在冲刺中需要时能够拥有其需要的数据。

强大的测试匹配功能还意味着，如果数据存在，匹配并分配到适当的测试案例之前，可以从多个不相干的源确定并挖掘数据。这样的测试匹配已证实，与手动流程相比，可将查找和提供数据所需的时间减少 95%，同时允许团队在运行前了解哪些测试将因数据问题而失败。

在集中化数据仓库中存储测试数据可进一步避免团队等待数据在下游变为可用时造成的瓶颈。CA Test Data Manager On Demand Test Data 门户提供动态表单构建，允许用户根据信用卡类型或地理位置等特定条件选择所需的数据种类。此表单将分配测试案例所需的数据，或者，如果不存在任何数据，将创建新数据。这意味着工程师有更多的时间来修复通过测试暴露的缺陷，而不是花费 50% 的时间来等待提供数据。它还意味着组织能够将数据所有权集中于 IT 安全团队，仅将敏感数据提供给请求这些数据的经授权的员工。

## 成本

最后，更好的 TDM 策略会降低成本超支的风险。及早检测并解决缺陷能够将缺陷的产生降低多达 95%<sup>5</sup> 并在每个缺陷上实现 5 万美元<sup>5</sup> 以上的节省。此外，使用更小、更丰富的数据子集能够将每个数据库的基础架构成本降低多达 5 万美元<sup>5</sup>，而运行更少、更优质的测试还会显著压低测试时间和成本。使用合成数据还会消除昂贵的非合规性的危险，因为敏感信息不需要离开生产环境。

---

## 第 4 部分

### 摘要

将生产数据复制到作为“真实”源的非生产环境会阻止持续交付的成功实施。需要适合的数据以对软件进行完整测试时，测试团队很可能发现自己没有这些数据。因此，他们无法迅速响应不断变化的业务需求。尽管构建子集和屏蔽工具可解决某些痛点，但生产数据的质量不足以发现并解决缺陷。

采用端到端的基于需求的软件开发方法并将测试数据视为宝贵的可重用资产对持续交付的成功实施至关重要。自始至终从“创意”开始工作，团队能够使用在正确的时间提供到正确位置的“适合的”数据迅速响应不断变化的业务需求，从而在所需最少的测试运行次数中对软件进行完整测试。

通过此方法，组织能够在其测试数据供应中建立效率，同时缓解使持续交付化为泡影的延迟、返工以及成本螺旋形上升的风险。它促成宝贵软件的持续交付，按时履行关键业务需求，成本更低。

## 第 5 部分

### 参考

- 1 <http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-life-cycle/>
  - 2 <http://www.agile-designer.com/resources/test-case-generation-bloor-market-report/>
  - 3 请参阅 Llyr Wyn Jones 的 Primer of A Critique of Testing 了解对流程图建模的概念化数学处理，地址为 <http://www.agile-designer.com/resources/critique-testing-primer/>
- 

## 第 6 部分

### CA Technologies 的优势

CA Technologies (NASDAQ: CA) 致力于提供有助于客户管理和保护复杂 IT 环境以支持敏捷业务服务的 IT 管理解决方案。企业组织利用 CA Technologies 软件和 SaaS 解决方案加快创新，转变基础架构并保护从数据中心到云的数据和身份信息。CA Technologies 致力于确保我们的客户通过使用我们的技术实现所需结果和预期业务价值。有关我们客户成功案例的更多信息，请访问 [ca.com/cn/customer-success](http://ca.com/cn/customer-success)。有关 CA Technologies 的更多信息，请转到 [ca.com/cn](http://ca.com/cn)。



## 第 7 部分

### 关于作者



#### Grid-Tools 联合创始人，CA Technologies 副总裁

Huw Price 被 QA Guild 评为“2010 年度 IT 总监”，担任多家美国和欧洲软件公司的首席技术架构师。他曾专门研究测试自动化工具，已推出多款创新产品，重新精心打造了适合软件行业使用的测试模型。



联系 CA Technologies, 网址: [ca.com/cn](http://ca.com/cn)



CA Technologies (NASDAQ: CA) 致力于开发促进企业转型的软件，为其抢占应用程序经济的先机。软件是各行各业的核心。从规划到开发再到管理和安全性，CA 正与全球各地的公司开展跨移动、私有和公共云、分布式和大型机环境的合作，以改变我们的生活、交易和沟通方式。要了解详细信息，请访问 [ca.com/cn](http://ca.com/cn)。

1 波耐蒙研究所：2014 年数据违规成本

2 欧盟提议数据保护法律

3 2013 年数据违规调查报告

4 Bender 报告：基于需求的测试

5 从 Grid-Tools 的实施经验中收集的指标