北京金融科技产业联盟
BEIJING FINTECH INDUSTRY ALLIANCE

金融大数据测试应用研究报告

北京金融科技产业联盟

2026 年 8 月

版权声明

编制委员会

编委会成员：

何　军　　黄程林　　童　蕙

编写组成员：

单姜一　　曹嘉欣　　蒋飞月　　陈永康　　闫宝旺　　王　莉

路　璐　　戴传飞　　张　源　　焦鹏欢　　张　婷　　王海标

陈理想　　崔雨萍　　龙志中　　吴兴杰　　张丙天　　马一龙

编　审：

黄本涛　　曹旭辉　　魏中宣

参编单位

交通银行股份有限公司

中国邮政储蓄银行股份有限公司

华为技术有限公司

海光信息技术股份有限公司

曙光信息产业股份有限公司

目 录

摘要：随着金融业数字化转型持续深入，数据形态日趋多样，跨系统交互日益频繁，传统测试手段已难以充分覆盖复杂的数据处理场景。本文从数据生命周期和系统安全质量两个维度出发，构建系统化的测试范畴，系统梳理典型应用场景与建设需求，提出涵盖数据质量、系统整合及处理性能的多维测试规范。结合实践案例，验证了所提方法的适用性与有效性，旨在为金融机构提升数据质量验证能力、测试执行效率及安全合规管理水平提供参考。

一、引言

（一）研究背景

在数据驱动决策的时代，大数据技术的复杂性以及数据处理过程中的不确定性，对传统质量保障方式提出了新的要求。大数据测试应运而生，作为一套系统化验证手段，旨在对大数据系统、应用及全链路处理流程的准确性、性能与稳定性进行综合评估，确保海量、高速、多源的数据能够被准确、高效、可靠地加工转化，最终输出可支撑业务决策的高质量数据成果。

在传统交易类系统中，测试对象通常聚焦于明确的交易功能或既定业务流程，其数据结构、处理规则及调用关系相对固定。测试重点涵盖功能正确性、接口衔接性、并发响应能力及事务一致性等关键质量属性。在实际执行中，测试通常依托预先准备的测试数据，在单一系统或少量关联系统范围内完成验证，并依赖人工比对方式确认测试结果。

大数据系统则涉及更为广泛的数据来源、繁杂的处理环节及多元的技术组件。数据通常需历经采集、清洗、存储、计算、分发等多重环节，方可交付至不同平台与业务应用使用，潜在问题往往在后续加工或实际应用过程中才逐步暴露。因此，大数据测试除关注基本功能实现外，还需从数据全链路完整性、系统运行稳定性及安全管理合规性等维度进行综合验证，其测试范围与实施路径均与传统交易类系统存在显著差异。

（二）研究目标

本报告聚焦金融业大数据测试应用需求，紧扣大数据系统数据规模大、处理链路长、组件协同复杂等特点，系统梳理金融机构开展大数据测试面临的主要问题与建设诉求。围绕数据质量、系统整合、处理性能、可靠性及安全合规等关键维度，研究提出相应的测试规范、通用技术指标与实施路径，并结合实践案例验证所提方法的适用性与有效性，旨在为金融机构构建大数据测试体系、提升测试能力提供系统性的参考指引。

（三）重要作用

在金融行业数智化逐步落地、监管体系日趋完善的背景下，大数据测试成为金融数据治理、业务创新、风险防控的重要支撑，其作用集中体现在保障数智化转型稳健落地、夯实合规监管风控体系、推动行业测试标准化规范化建设三个方面。通过前置化、体系化、常态化的全流程大数据测试，可在数据接入模型、嵌入系统、落地业务流程前，精准识别数据质量隐患与数据安全风险，全面提升金融数据应用的真实性、可靠性与安全性，为金融行业高质量发展筑牢数据根基。

1.赋能数智化转型，保障业务稳健创新落地

随着金融科技持续迭代升级，大数据、人工智能已成为驱动金融业高质量发展的核心引擎，大数据应用场景不断拓展，已全面覆盖智能风控、反欺诈管理、精准营销、智能投顾等关键业务领域，海量、多元、实时的金融数据已成为机构智能决策与业务

变革的核心支撑。数据质量直接决定模型运算结果与业务决策成效，而大数据测试通过对数据准确性、完整性、一致性、时效性的全方位严格校验，从源头规避“垃圾进、垃圾出”的行业痛点，确保输入业务系统与智能模型的数据真实有效，为深度挖掘数据价值、推动数据驱动型业务落地提供坚实保障。

同时，金融数智化转型对业务迭代速度、创新落地效率提出了更高要求，实时反欺诈、个性化金融服务、智能风控等创新业务，需要在风险可控的前提下快速上线、持续优化。大数据测试依托自动化、持续化、常态化的测试手段，大幅提升金融产品与数据服务的交付效率，有效缩短创新业务上线周期。此外，通过对生产环境数据质量、算法模型运行效果、大数据系统稳定性的持续监测与动态校验，能够全方位保障大数据系统平稳运行、核心业务连续运转，助力金融机构实现“控风险、提效率、优创新”的数智化发展目标。

2.夯实合规体系，规避监管与经营双重风险

当前金融行业数据安全、隐私保护及合规管控方面的监管要求持续趋严，针对客户身份信息、交易记录、征信数据等核心敏感数据的管控标准不断细化，对金融数据测试的全面性、严谨性、规范性提出了更高要求。围绕数据访问权限、传输与存储加密、敏感数据脱敏、审计日志留存等环节开展专项测试，能够有效检验相关安全管控措施是否得到有效执行，及时发现测试数据在提

取、传输、使用、留存和销毁过程中的风险隐患，为金融数据安全合规使用提供有力保障。

通过系统化的大数据测试，可生成翔实、规范且可溯源的测试评估报告，直观验证金融机构数据处理与数据应用各环节是否符合监管合规要求，有效规避违规罚款、法律诉讼、品牌声誉受损等各类经营风险。同时，常态化的质量与安全测试管控，能够持续提升内部数据可信度，为管理层与业务部门开展精准决策、风险管控提供可靠的数据支撑。对外则有助于充分展现金融机构在数据合规治理与隐私保护方面的能力，增强监管认可度与客户信任感，成为金融机构在数字化竞争中筑牢口碑、夯实市场优势的核心基石。

3.引领行业规范，推动测试标准化体系建设

大数据测试标准化是提升行业数据互通性、测试可复用性、跨机构数据互信度的重要基础，也是完善金融行业数据治理体系的关键环节。目前国内金融行业大数据测试体系仍有待完善，各金融机构的测试标准、校验方法、评估体系存在明显差异，导致跨机构数据协作、数据共享过程中存在标准不统一、信任成本高、协作效率低等问题。

统一的金融大数据测试框架、核心指标与校验方法论，能够为全行业数据共享、跨机构协作建立统一的质量基线，以标准化尺度衡量各类数据源的完整性、一致性、时效性等核心指标，有效消除因测试标准差异产生的数据壁垒，大幅降低跨机构数据互

信成本，提升行业整体数据协作效率。同时，标准化测试可统一数据脱敏有效性、权限控制合规性、审计日志完备性等核心校验标准，规避跨机构合作中的衍生合规风险。此外，标准化测试成果可作为行业风险管控、合规自律的重要依据，助力构建监管认可、行业通用的跨机构数据安全互认机制，持续推动金融大数据行业治理规范化、成熟化发展。

二、应用情况

（一）应用场景

金融机构以交易、客户、产品、风险等数据为基础，推进数据中台和智能化应用建设。大数据测试贯穿数据接入、存储、处理、共享和应用等环节，是保障数据在复杂链路中可用、可信、可控的基础活动。

1.风险管控场景

在风险管理、授信审批、反欺诈和反洗钱等场景中，数据需要经过实时采集、规则计算、模型识别和风险处置等多个环节。测试应关注源数据是否完整、加工逻辑是否正确、模型输入是否符合要求，以及实时处理结果能否及时、准确地支持风险识别和业务决策。

2.经营管理场景

在监管报送、财务核算、经营分析和管理决策等场景中，数据来源广、处理链路长，且对统计口径和报送时效要求较高。测试应重点关注跨系统数据的一致性、指标计算的准确性、数据追溯关系的完整性，以及批量作业能否在规定时限内稳定完成。

3.客户交互场景

在客户经营、精准营销、智能客服和智能投顾等场景中，客户信息、行为数据、产品信息和标签数据被持续加工和调用。测试需关注客户数据是否准确完整、标签和客群划分是否符合业务

规则、营销策略是否被正确执行，以及数据使用过程是否满足个人信息保护和权限控制要求。

4.交易运营场景

在支付清算、交易监测和渠道运营等场景中，数据处理具有高并发、低时延和连续运行等特点。测试除需验证交易数据传输和处理结果的正确性外，还应关注峰值流量下的处理能力、异常交易的识别效果，以及系统故障后的数据恢复和业务连续性。

5.平台建设场景

在数据平台建设、系统迁移、技术升级和跨系统协作场景中，测试重点则集中于数据接口适配、文件和报文传输、作业依赖关系、历史数据迁移以及多环境运行稳定性等方面。通过对上述场景开展针对性测试，可及时发现数据流转、系统协同和运行保障中的问题，为后续业务应用提供可靠基础。

（二）问题与建设诉求

金融机构在开展大数据测试时，常存在测试标准不统一、测试数据及场景还原不足、自动化和协同能力偏弱、安全合规管控不足等问题。这些问题分别影响测试结论的一致性、测试场景的有效性、测试执行的效率以及测试过程的安全性。围绕上述问题，从以下四个方面分析相关建设诉求。

1.统一测试标准与量化口径不足

部分机构在测试目标、测试过程、准入准出规则、质量评估和缺陷管理等方面缺少统一口径，数据质量和测试结果多依赖经

验判断，难以形成可复用、可审计的测试基线。这不仅影响不同项目、不同系统之间测试结果的横向比较，也不利于测试过程的规范管理。因此，需要建立覆盖测试目标、测试范围、测试流程、评价指标和结果判定的统一标准体系，明确不同类型测试的适用范围、质量阈值和准出要求。

2.测试场景覆盖和生产还原能力不足

金融数据来源广泛、关联关系复杂、处理方式多样，涉及的数据对象、处理规则和业务组合较多，测试场景设计难以做到全面覆盖。同时，测试环境常难以还原生产环境的数据规模和动态变化，会导致部分问题无法在测试阶段暴露，增加上线后的运行风险。因此，需要提升测试数据构造、场景模拟和质量校验能力，增强测试环境和测试数据对实际业务运行情况的反映程度。

3.测试自动化和协同能力仍有提升空间

大数据测试涉及表、字段、文件、作业和跨系统接口等多类测试对象，传统人工抽样和线下协作方式执行效率较低，回归测试成本较高，跨部门责任衔接和问题处置机制也有待完善。因此，需要提升自动化测试和跨部门协同支撑能力，完善测试任务管理、责任衔接和问题处置机制，提高测试执行效率和问题闭环能力。

4.测试过程中的安全合规管控仍需加强

测试数据涉及客户隐私、资金流水和交易明细等敏感信息，数据提取、传输、存储、使用和销毁等环节均需同步落实脱敏、权限、审计和留痕要求。若测试验证期间未做好合规管控，可能产生测试数据违规使用、权限管控失效或操作记录不全等问题。因此，需要完善与测试流程相衔接的安全合规管控机制，确保测试数据使用和测试活动全过程可追溯、可审计。

三、测试范畴

（一） 数据生命周期测试

1.数据接入测试

数据接入测试是为了确保数据从源头到存储、处理引擎传输的可靠及准确。重点从以下四个维度进行测试。

验证服务并发处理能力：在多通道、多分区、多消费组、多消费者场景，高并发进行生产消费数据时，应验证请求处理时延是否处于接受范围内，同时保证数据不丢失，不重复消费。

验证服务的数据传输能力：在高并发进行数据生产和数据消费时，在不受带宽限制时，保证能按照最大带宽进行数据高速传输，确保数据不阻塞。

验证实时接入数据分发与转储能力：在配置了下游转储服务时，可按照转储策略，实时将数据转移到目的服务，确保能够正确并完整地将数据传递到下游。

验证多种源数据类型的通道：对 BLOB、JSON、CSV 的源数据类型的通道进行功能覆盖验证，验证扩缩容、生产、消费、转储等功能是否正常，验证与类型相关的特有属性是否生效，验证各源数据类型之间是否正常转化，验证转化前后生产、消费等功能是否正常。

2.数据存储测试

海量数据存储功能是大数据提供对外服务的基石。重点从以下五个维度进行测试。

海量分级存储：支持海量数据存储，大数据存储服务 HDFS 采用分布式结构，根据业务存储诉求，扩容规模来实现大量数据存储。支持异构分级存储框架，可根据不同业务读写性能要求，将数据存储在不同的存储介质上。同时，通过数据校验机制保联数据在存储与传输过程中的完整性。用户数据默认采用 CRC32C 校验方式，保存在 HDFS 上，确保数据正确性。

可靠性：对于海量数据存储系统，其核心在于保证数据可靠性，即确保数据的完整性和读写处理不中断。HDFS 通过数据块的多副本机制，保证在部分节点故障时，数据不丢失。另外，HDFS 通过 NameNode HA 方案，实现单点故障时读写不中断。

数据写一致性：多个服务并发写入同一文件时，为保证数据一致性，HDFS 采用锁机制，确保同一时刻仅允许一个调用者执行写操作，而允许多个调用者同时执行读操作，即遵循“一次写入、多次读取”的并发控制原则。

读写性能：数据存储系统，还需重点关注读写性能，重点验证单文件大规模读写和海量小文件读写两类典型场景。

数据安全：HDFS 提供对文件内容的加密存储功能，避免敏感数据明文存储。业务应用只需对指定的敏感数据配置加密存储，加解密过程业务完全不感知。

3.数据处理测试

在完成数据的采集、存储之后，我们需要对数据进行一系列的分析和计算，来实现对数据价值的挖掘和应用。重点从以下五个维度进行测试。

测试数据多样性：ETL 是指按照一定规则对数据进行清洗、抽取与转换的过程。测试过程中需要保证任务在各类数据场景下稳定运行，如宽表、空表、空行、表单列值超大、特殊字符、空文件、空值、null 值、日期（标准，带时区）、小文件、数据倾斜、百万分区、脏数据等典型场景。

计算结果准确性：在对数据进行批量计算后，需要验证计算过程是否符合预期，验证最终数据的完整性和准确性，即保证结果无冗余，数值准确。在大数据量的场景下，输出没法直接做衡量，可以分两层测试，使用小数据量来验证逻辑处理的准确性，人工计算出期望输出结果。在大数据量的场景下，可以使用多种不同的计算引擎进行测试，如果得到的结果输出是一样的，则将此结果作为期望结果。

SQL 解析能力验证：该测试场景下，不关心 SQL 语句的执行结果，只关心任务能否跑通过，可以使用 SQLSmith 自动生成随机 SQL 语句，检查解析器是否正常。

多组件之间的交互测试：在数据湖或者湖仓一体场景下，底层数据是供上层各个计算引擎进行操作的。因此，在客户真实环境中，很可能数据是 Spark 写 Hive 读或者 Hive 写 Spark 读等，需要保证数据在跨组件计算时，任务正常运行且计算结果正确。

任务稳定性测试：测试时需要关注任务在高并发、高负载下的稳定性以及系统故障情况下的可靠可用能力。观察指标包括文件句柄、CPU 利用率、内存 OOM/FullGC、磁盘 IO、磁盘容量、进程状态、连接数、Handler 线程数、任务提交、运行时长等。

4.数据治理测试

数据治理测试的重点是数据安全测试。数据安全提供数据的加密保护能力，可以通过数据加密来防止敏感数据遭到有意或无意的误用、泄漏或盗窃，从而帮助用户采取合理措施来保护其敏感数据的机密性和完整性、可用性，帮助用户建立安全预警机制，增强整体安全防护能力。重点从以下五个维度进行测试。

不同类型的数据源：支持 Hive、DWS 等多种类型的数据源，可以作为源端和目的端对数据进行加密。

不同的数据类型：针对不同数据源源端和目标端组合，可能存在一些不同类型的字段转换问题，可以选择常见且较多使用的数据类型进行测试。

待加密数据的量级：数据量的大小，会影响执行数据加密作业的时长，可以通过调优集群的配置参数来提高数据加密的速率。

加密结果的正确性：需要确保数据加密任务执行完成后，未加密字段在目标端数据与源端数据保持一致，加密字段在目标端数据加密结果正确。数据未出现乱码、数据丢失等现象。

调度结果的一致性和正确性:调度任务需要保证在配置的调度周期内运行，同时调度任务的执行结果按照预期状态完成，数据正确。

（二） 数据专项测试

1.功能测试

功能测试是依据业务需求和功能规格，对系统应具备的功能及其运行结果进行验证的测试活动，重点检查系统在给定输入、条件和操作下，是否能够按照预期完成相应处理。一般从功能需求背景分析、功能达成、功能交互影响三个维度，分析测试影响因子和测试数据。功能测试用例设计常用的工程方法有等价类划分、边界值分析、错误推测、因果图和判定表等。

数据质量是数据应用的核心基础，在大数据功能测试中，数据质量测试愈显重要。质量测试主要从完整性、一致性、准确性、及时性四个维度开展。

数据完整性: 验证数据在存储、传输、处理过程中无丢失或损坏。验证时重点关注如 HDFS 文件块完整性、HBase Region 分裂后 Scan 全表数据无重复或缺失、Kafka 按偏移量消费到所有消息、Flink Source 端数据是否全部被处理（无丢失）等。

数据一致性: 确保数据在分布式环境下符合预期语义（强一致性、最终一致性）。典型的验证场景有 Hive 源表与目标表数据比对一致、Flink Checkpoint 实现 Exactly-Once 状态一致性、

Kafka 同一消息在不同分区的逻辑一致性、HetuEngine 跨不同数据源（如 Hive+ClickHouse）查询结果一致性等。

数据准确性：验证数据处理结果与业务逻辑预期匹配（无计算错误）。验证时重点关注 Hive/Spark SQL 查询和 UDF 计算结果的正确性、Flink 窗口聚合结果的正确性、ClickHouse 聚合函数结果和特定引擎计算精度的准确性、HetuEngine 查询下推和跨源查询语义的准确性等。

数据及时性：保障数据在业务时间窗内可消费（低延迟）。典型的验证场景有 Flink 端到端延迟指标监控、Spark 批处理/流处理作业处理时间是否符合 SLA、Kafka 消息生产到消费满足低时延目标、Yarn 资源分配调度延迟对作业运行性能的影响是否符合预期等。

2.性能测试

性能测试是指评估大数据系统在海量数据、高并发及复杂计算场景下的吞吐处理能力、响应时间及资源利用情况。

鉴于大数据系统的分布式复杂性、数据规模海量化及实时性要求高的特点，性能测试需覆盖数据从接入到分析的全链路。依托自动化框架，并结合基准测试工具与深度全链路监控，能有效保障系统在高并发、长周期运行中的稳定性与效率。性能测试从以下三个维度展开。

基准测试：使用标准化测试工具和数据集评估系统基础性能，表 1 展示了常用的大数据组件和基准测试工具。

表 1 常用大数据组件和基准测试工具

组件	类别	基准测试工具	测试场景
HDFS	分布式存储	NNBench/TestDFSIO	NameNode元数据读写性能/HDFS读写吞吐量
MapReduce	分布式计算引擎	HiBench	离线数据处理性能
HBase	NoSQL数据库	YCSB	随机读写/扫描性能
Hive	数据仓库	TPC-DS	复杂SQL查询性能
Spark	内存计算引擎	HiBench/TPC-DS	离线数据处理性能/复杂SQL查询性能
HetuEngine	交互式查询引擎	TPC-DS	跨源查询性能
Kafka	消息队列	Producer/Consumer Perf Test	消息生产/消费性能
Flink	批流一体引擎	HiBench/Nexmark	实时流数据处理性能/实时流数据处理性能
ClickHouse	OLAP数据库	SSB-Benchmark	高并发查询与批量导入性能

负载测试：逐步增加并发或数据量，定位性能拐点，评估系统在不同负载下的性能表现，包括响应时间、吞吐量、并发用户数等。

长稳测试：长时间高负载运行（如 7×24 小时），验证系统的承载能力和稳定性。

3.过载测试

过载测试是通过构造超出系统设计容量的负载，验证大数据组件在极限压力下的稳定性、降级策略和故障恢复，以保障系统在高流量负载下依然能够稳定地对外提供可用功能。常见的过载测试类型主要有以下四类。

流量过载测试：模拟超规格的突发数据流量，验证组件在高负载下的性能衰减与容错能力，通过监测吞吐量断崖下跌、进程死锁、资源异常等失效模式，量化系统崩溃临界点并优化弹性伸缩策略。典型的操作如:对 Kafka 注入 2 倍峰值流量,观测 Leader 切换延迟，以及 ISR 同步是否阻塞。

资源耗尽测试：构造业务压力主动耗尽 CPU、内存、磁盘或线程池等关键资源,验证组件在极限压力下的自保护机制与失效边界，通过观测资源枯竭引发的级联雪崩（如 OOM Killer 触发、任务死锁、存储写满阻断）和优雅降级能力，定位资源调度策略的脆弱点，给出优化手段。典型的操作如: 1）对 Spark Executor 注入超量数据集,验证堆外内存溢出时是否触发黑名单驱逐而非集群瘫痪。2) 将 HDFS DataNode 磁盘写满至 95%，观测副本自动均衡机制能否优先迁移数据释放空间，避免 NameNode 进入安全模式。

并发连接测试：模拟海量客户端同时请求（如十万级 TCP 连接），验证组件的资源分配极限与拒绝策略有效性，通过观测线程池耗尽、文件描述符枯竭、连接泄露等导致的雪崩效应，优化连接复用机制与熔断阈值。典型的操作如: 1）Flink 模拟 10 万并发客户端提交 Checkpoint 请求对 TaskManager 进行压测。2）ClickHouse 突发千级查询连接冲击，观测线程池拒绝策略能否保护内核免于 OOM 崩溃。

大 SQL 过载测试： 通过超高复杂度查询（如百表 Join、深度嵌套子查询）与高并发的组合冲击，验证 SQL 引擎的优化器崩溃边界与资源隔离机制有效性，重点暴露元数据库连接枯竭、计算引擎死锁、调度队列雪崩等深层故障链。典型的操作如：对 Hive 提交 50 并发的 DDL 操作（ALTER TABLE）与复杂 ETL 查询，观测 Metastore DB 连接池耗尽是否触发查询阻塞。

4.可靠性测试

数据可靠性测试是通过模拟极端故障场景（如节点宕机、网络中断、数据损坏），验证系统能否保障数据完整性、一致性与服务连续性的过程，其测试重点包括以下四个维度。

数据持久性： 验证硬件故障时数据是否不丢失，常使用混沌工程进行磁盘损坏、节点宕机等故障注入。

数据一致性： 验证数据在读写操作、状态恢复或副本同步过程中是否始终符合预期语义（如 ACID、Exactly-Once）；

服务高可用： 在部分进程故障时，组件仍能对外提供快速恢复服务（高可用性）或降级服务能力。

容错与恢复能力： 系统能在故障后自动或手动恢复处理，并能从故障点继续，保证处理逻辑的连续性和正确性。其中，典型的大数据组件可靠特性和测试场景如表 2 所示。

表 2 典型的大数据组件可靠特性和测试场景

组件	类别	可靠特性	测试场景

HDFS	分布式存储	• 多副本/机架感知策略 • NameNode HA（ZKFC自动切换）	1. 数据持久性测试：构造副本丢失（小于副本数）下的数据完整性和读写连续性； 2. NameNode故障切换及RTO。
Yarn	资源调度	• ResourceManager HA（基于 ZooKeeper） • 任务自动重试（ApplicationMaster容错）	1. 故障转移测试：主RM故障时备RM自动接管任务调度； 2. 任务弹性测试：随机杀死NodeManager，任务自动迁移至健康NM节点。
HBase	NoSQL数据库	• RegionServer HA + HMaster HA • WAL（Write-Ahead Log）保障写操作持久化	1. Region重新分配：RegionServer故障时Region重新分配，保障业务连续； 2. 读写数据一致性。
Hive	数据仓库	• 元数据高可用 • 查询容错（Tez/MapReduce 任务重试）	1. 依赖的服务DB存在单点故障时，元数据读写不中断； 2. 结果一致性：相同SQL在MR/Tez/Spark等引擎的执行结果一致。
Spark	内存计算引擎	• RDD 血缘（Lineage）实现断点重算 • Driver HA (Cluster Mode)	1. Executor容错：运行时杀死Executor，验证Task是否重新调度； 2. Shuffle可靠性：构造磁盘写满，验证Shuffle数据是否丢失。
HetuEngine	交互式查询引擎	• Coordinator HA + 状态持久化 • 动态故障切换（Worker节点）	1. Coordinator HA测试：SQL查询的主Coordinator被杀死，自动路由到备节点； 2. Worker节点失效场景验证大查询能否自动续跑。
Kafka	消息队列	• 副本同步机制（ISR） • 生产者ACK机制	1. 故障场景下的数据副本一致性验证； 2. 数据高可靠测试：Leader节点故障时，完

			整消息生产/消费。
Flink	批流一体引擎	• Checkpoint机制（精确一次语义） • JobManager HA（基于ZooKeeper）	1. 状态一致性测试: TaskManager故障重启，验证状态是否从Checkpoint恢复; 2. 反压恢复: 解除数据积压后，验证数据吞吐能力恢复。
ClickHouse	OLAP数据库	• 多副本表（ReplicatedMergeTree） • ZooKeeper 协调元数据	1. 副本同步延迟: 写入后Leader节点故障，验证副本数据一致性; 2. ZooKeeper依赖测试: 模拟ZK宕机，验证查询是否可读。

5.安全测试

安全测试活动包括白盒安全测试、渗透测试及 Fuzz 测试。

白盒安全验证: 一种软件可信、安全测试活动，基于对产品的软件系统设计、实现和开发流程的了解，对标第三方认证和业界实践，通过工具扫描检测、人工分析等方式，对产品的设计、开发、构建的过程及交付件进行可信安全检查，涉及产品的架构、源码以及使用的开源和第三方软件、操作系统，从而识别和发现存在的安全问题。

渗透测试: 在特定约束下模拟恶意黑客对系统进行攻击，来绕过系统已实施的安全防护功能，以此来发现系统安全防护薄弱点、技术缺陷、系统漏洞等，并利用漏洞来破坏系统的完整性、机密性、可用性，证明系统存在安全问题的一种测试活动。渗透测试流程主要包括信息收集、威胁分析（识别数据、服务价值资

产）、攻击路径分析、根据绘制的攻击树适配攻击模式库，对关键资产逐一进行漏洞挖掘和攻击。

Fuzz 测试：渗透测试方法中的一种，通过对被测对象的穷举、泛洪发包的方式，发现系统中存在的安全脆弱性（Vulnerability）与风险（Risk），它是健壮性测试的一种形式，聚焦通信接口，发现与安全相关的问题，如溢出和边值条件，从而推动产品安全问题整改，提高产品安全性。

6.自动化测试

大数据系统由于庞大复杂，手动测试存在效率低下且不可持续的问题，因此自动化测试技术就显得尤其重要。自动化测试重点分为 UI 自动化、接口自动化、SQL 类自动化。

UI 自动化测试：UI 层是用户使用产品的入口，所有功能通过这一层提供给用户，常见的测试工具有 Robot Framework、Selenium、Appium、UFT 等。

接口自动化测试：接口测试执行工具较为丰富，常用的有 Postman、Jmeter、Fiddler 等。另外，常常将接口测试代码或自动化脚本嵌入流水线，以实现 CI/CD 集成测试。

SQL 类自动化测试：使用数据驱动的测试框架可以显著提升测试效率，尤其适合验证大数据场景下的 ETL、数据仓库计算和 SQL 查询功能。它将测试用例的 SQL 语句和期望结果存储在外部文件（如 CSV、Excel、YAML），测试框架自动读取该文件，即可实现执行 SQL，比对实际结果与期望值，并生成可视化报告。

四、测试体系建设

（一）体系设计要求

1.设计目标

金融业大数据测试体系建设的目标，是建立覆盖数据质量、系统整合与处理性能的统一测试规范，形成可执行、可度量、可追溯的验证能力，并为跨部门协同和持续优化提供共同依据。

一是建立统一标准体系。制定覆盖金融机构、基础设施及金融活动的大数据测试标准，确保统计要素定义、分类及编码规则的统一性，为数据互联互通奠定基础，作为金融业大数据测试应用指导。

二是形成全过程质量管控能力，将数据接入、加工、校验、共享等关键环节纳入测试范围，并通过规则、脚本和平台工具提升验证的自动化与持续化水平。

三是强化合规与风险防控支撑，将数据分类分级、最小必要使用、权限控制和审计留痕要求嵌入测试过程，为金融业务稳健运行提供可靠保障。

2.设计原则

体系设计应坚持全流程覆盖、分类分级、风险导向、可度量可追溯和动态迭代五项原则，并与机构现有的数据治理、测试管理和安全管理机制相衔接。

全流程覆盖与分类分级。测试范围应覆盖数据接入、处理、存储、共享和应用等关键环节；同时按照数据敏感度、业务重要性和风险等级确定差异化测试深度，避免“一刀切”配置。

风险导向与合规嵌入。优先覆盖监管报送、风险管理、跨系统传输等高风险场景，并在测试数据准备、任务执行、结果留存和问题处置等环节同步落实合规要求。

可度量与可追溯。测试规则、指标阈值、执行结果、缺陷处置和责任主体应能够关联留存，支持测试过程复核、问题定位和审计检查。

动态迭代与组件兼容。测试规范、指标和工具能力应随着业务变化、监管要求和技术栈演进持续更新，并兼容不同数据库、计算引擎和测试环境。

3. 通用技术指标体系

通用技术指标用作衡量数据质量、系统整合、处理性能、可靠性和安全合规等方面的测试要求，为测试方案设计、测试执行结果提供统一依据，如表 3 所示。需注意，指标阈值仅供参考，具体阈值应由机构结合业务重要性、生产运行基线和服务等级协议确定。

表 3 通用技术指标体系

维度	核心指标	检测方法
数据质量	缺失率 ⩽ 0.1%	抽样统计
安全合规	脱敏覆盖率100%	规则引擎扫描
性能	99%查询响应<2s	压测工具监控

可追溯性	全链路血缘追溯能力		元数据关联分析
跨部门协作及时性	待办处理及时率≥95%		依据平台操作审计日志、处理及时率计算公式计算
测试准出	需求覆盖度=100%		依据测试管理系统或过程文档
	用例执行率=100%		依据测试管理系统或过程文档
	缺陷关闭率	轻微级别≥80%	依据测试管理系统或过程文档
		一般级别≥95%	
		严重级别=100%	
		致命级别=100%	

（二）多维测试规范构建

1.数据质量测试规范

数据质量测试规范应围绕数据质量管理、测试流程与工具链能力三个维度展开，核心目标在于全面验证数据的完整性、有效性、准确性、唯一性、及时性及业务规则符合性。

（1）指标体系构建

数据质量是保证数据应用的基础，用于评估数据是否达到预期设定的质量要求。质量指标设计时，它的评估要求主要包括完整性、有效性、正确性、唯一性、及时性和业务合理性六个方面。

完整性主要关注数据记录、关键属性及关联关系是否完整，例如数据读取前后记录条数应保持一致，关键字段不应缺失，数据 A 的外键取值应能够在数据 B 的主键范围内匹配。

有效性主要检查数据是否符合既定格式和取值要求，如身份证号应符合相应编码规则，字段取值应处于规定范围内。

正确性关注数据是否符合客观事实和逻辑规则，例如年龄不应为负数，概率类字段应在 0 和 1 之间取值。

唯一性用于检验主键以及指定字段组合是否存在重复记录

及时性则关注数据接入、更新与实际业务变化之间的时间差是否满足业务要求。

业务合理性主要验证数据是否符合预定义的业务规则和业务逻辑。

数据质量管理应贯穿数据接入、清洗、处理、分发和存储全过程，不同处理阶段对上述质量维度的关注重点有所不同，其对应关系如表 4 所示。

表 4 数据处理各阶段的数据质量管理要求

评估要求	数据处理阶段			存储数据
	数据读取	数据清洗	数据分发	
记录完整性	✓			
属性完整性		✓		✓
关系完整性				✓
格式有效性		✓		✓
值域有效性		✓		✓
逻辑合理性		✓		✓
接入及时性	✓			
更新及时性			✓	
主键唯一性			✓	✓
数据唯一性			✓	✓
业务合理性				✓

（2）测试流程构建

一个完整的测试流程应包括测试准备、测试设计、测试执行、测试评估、问题整改与回归测试五个阶段，形成从测试策划、验证实施到问题整改的全流程管理机制。

测试准备阶段，应结合业务需求明确测试范围，覆盖数据读取、清洗、分发和存储等相关处理环节。同时梳理适用的监管制度、管理规范及技术标准，统一完整性、有效性、正确性等数据质量术语的口径。在此基础上，搭建与实际业务处理场景相适应的测试环境，准备客户、交易等相关测试数据，并兼顾正常、异常及边界数据情形。

测试设计阶段，应围绕完整性、有效性、正确性、唯一性、及时性和业务合理性六个方面设计测试点和测试用例，并结合不同数据处理阶段确定测试重点。例如，在数据读取阶段可重点检查交易记录是否完整、数据接入是否及时；在数据清洗阶段可重点验证关键属性是否缺失、字段格式和取值是否符合要求；在数据分发和存储阶段，需关注关联关系、数据重复、更新时效及业务逻辑是否满足规定。测试数据应覆盖缺失、重复、格式错误、异常取值和逻辑不合理等典型问题。

测试执行阶段，应验证数据采集、质量检核和问题跟踪等功能是否能够正常运行。对于样例数据采集，应检查全量采集、抽样采集及指标信息收集是否符合预设规则，采集任务能否完成创建、修改、下发和定期清理等管理操作；对于质量检核，应验证

系统能否依据既定规则对处理过程数据和存储数据开展多维度检核，并通过任务调度实现离线数据的周期性检测。同时，还应检查质量报告生成、问题反馈、告警推送、处置跟踪和知识库沉淀等机制是否有效，推动问题发现、处置和复核形成闭环。

测试评估阶段，应对各测试点的执行结果、通过情况和发现的问题进行分析，并依据既定质量标准对数据的完整性、有效性、正确性、唯一性、及时性和业务合理性进行综合评价，判断数据是否满足业务应用要求。在此基础上形成测试报告，明确测试范围、测试环境、测试数据、执行情况、质量结论、问题清单及改进建议，为后续数据质量管理优化提供依据。

问题整改与回归测试阶段，应将测试发现的问题及时反馈至相关数据提供、处理和使用部门，明确责任主体、整改措施和完成时限。整改完成后，应针对原问题及其关联影响范围开展回归验证，确认问题是否得到解决，并检查整改是否引入新的质量风险，确保数据质量管理能力持续满足业务和管理要求。

（3）工具链搭建

数据质量工具链围绕规则配置、数据采集、质量检核、任务调度、结果反馈和问题处置等环节，形成自动化质量管理闭环。

其中，数据采集和质量检核是工具链的核心环节：前者负责从数据读取、清洗、分发和存储等环节获取样例数据及质量指标信息，为检核提供数据；后者依据预设规则，对过程数据和存储数据开展实时或批量验证。任务调度、结果反馈和整改跟踪则用

于保障检核任务有序执行，推动问题及时处置和规则经验持续沉淀。数据质量管理与数据处理流程的关系如图 1 所示。

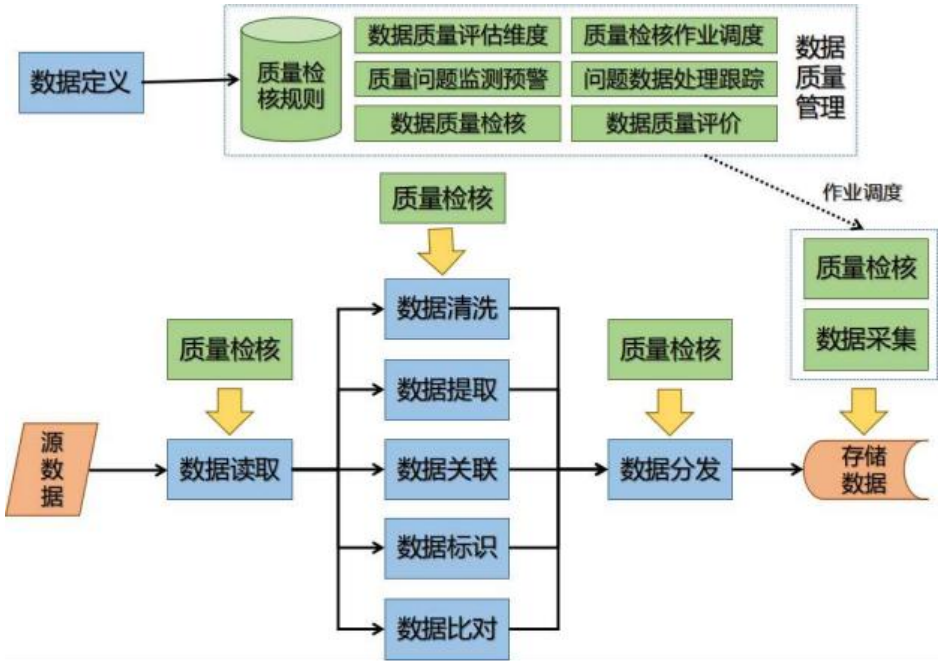

图 1 数据质量管理与数据处理流程关系

数据采集主要包括样例数据采集和指标信息收集两部分。样例数据采集是根据采集任务，从数据表、数据文件等存储对象中获取全部或部分数据，形成供后续检核使用的检测样例；指标信息收集则是汇集数据探查、读取、清洗和分发等环节产生的质量检核结果。指标信息可通过数据资源标识符、数据项标识符等与具体数据对象建立关联，并记录检测时间、检测数据量、错误数据量等内容，以反映不同数据资源和字段项的质量状况。

指标信息应覆盖记录完整性、属性完整性、格式有效性、值域有效性、逻辑合理性、接入及时性、更新及时性、主键唯一性和数据唯一性等质量维度。例如，记录完整性指标可反映数据读

取前后的应读取记录数和实际读取记录数；属性完整性、格式有效性和取值有效性指标可反映指定字段的检测范围及错误情况；及时性和唯一性指标则用于识别数据接入、更新和分发过程中的延迟或重复问题。不同处理阶段的输出指标存在差异，如表 5 所示。

表 5 数据处理各阶段输出的指标信息

指标信息	数据探查	数据读取	数据清洗	数据分发
记录完整性指标信息		✓		
属性完整性指标信息	✓		✓	
格式有效性指标信息	✓		✓	
值域有效性指标信息	✓		✓	
逻辑合理性指标信息			✓	
接入及时性指标信息		✓		
更新及时性指标信息				✓
主键唯一性指标信息				✓
数据唯一性指标信息				✓

采集任务管理时，工具链应以任务为基本单元，对数据采集任务进行创建、修改、下发、启用和停用等操作，并建立数据采集任务与质量检核任务之间的关联关系，实现任务同步执行。采集方式可根据实际需要选择全量采集或抽样采集，并灵活设置抽样比例、抽样时间范围和采集数据规模。对于已采集的样例数据，还需明确其存放位置和留存期限，并在完成检核后按要求清理，降低测试数据长期留存带来的管理风险。

质量检核可区分为实时数据检核和离线数据检核两类。实时数据检核侧重嵌入数据处理过程，主要在数据读取、数据清洗和数据分发等环节开展；离线数据检核侧重周期性执行，通过采集存储对象中的样例数据，结合调度任务开展批量化校验。两类检核均应依据预先定义的数据质量规则，对数据进行多维度探查，并输出相应的质量检核结果和分析报告。

实时数据检核通常包括规则定义、规则下发、过程检核和指标输出等环节。业务人员可在数据定义阶段，为不同数据资源配置相应检核规则，并通过接口将规则下发至数据读取、清洗和分发模块，由各模块在处理过程中完成质量校验；相关模块再定期将检核结果输出至数据质量管理模块，用于后续统计分析和问题识别。离线数据检核则包括规则定义、数据采集、质量检核和作业调度等环节，系统可按照设定周期对样例数据执行检核任务，并通过统一任务管理机制，对规则配置、任务创建、任务下发、任务执行和任务启停进行管理。每项任务可包含多项检核规则，并可根据字段特点从规则库中选用已有规则或进行自定义配置。

完成数据采集和质量检核后，应基于检核报告和业务反馈开展问题发现、结果反馈、告警推送和跟踪处置。对于实时数据，可定期汇总指标信息并生成质量报告；对于离线数据，可在每个检核周期结束后形成对应的分析结果。发现问题后，应将结果及时反馈至数据提供方、数据处理方和数据使用方；对于达到告警条件的问题，可通过页面消息、服务接口、邮件或短信等方式推

送给相关责任人员。问题跟踪可通过问题数据占比、发生频次和变化趋势等方式展示处置进展，离线数据还可结合清洗、格式转换、数据回填等措施进行整改，并通过二次检核验证整改效果。在持续检核和问题处置过程中，还应同步建设数据质量知识库，沉淀检核规则、典型问题、成因分析、处置方案和适用场景等内容，逐步形成覆盖质量维度和重点业务领域的规则与经验积累机制。

2.系统整合能力测试规范

系统整合能力测试规范用于约束开发与测试部门进行跨团队、跨测试环境协同测试时的操作流程、协作机制与权限管理。

其目的在于明确多角色协作的责任边界，提升复杂业务场景下的测试覆盖与执行效率，保障跨环境作业验证的一致性与安全性。通过统一平台实现对多测试环境的集中管控与可视化调度，规范测试行为，降低人工干预风险，确保测试过程可追溯、权限可控、结果可靠。

（1）跨环境操作兼容性验证

针对基于不同业务场景测试需求搭建的各测试环境，平台需能提供统一接口，提供自动化操作与可视化链路分析能力，支持开发与测试人员高效开展跨业务场景的协同验证。

围绕“统一测试”的目标，平台要兼容各测试环境，为开发测试人员提供统一访问入口。同时要能够将测试过程中的手工操作自动化，并能于各环境兼容执行。具体测试点如下。

一是验证不同角色账号均能通过平台统一门户访问所有环境，且各环境访问权限具有一致性。

二是验证待测作业已于对应验证环境部署，若未部署平台可生成待办给作业开发人，并可视化处理进度。

三是验证自动化操作于各环境均可执行，如跟踪链路、文件造空、作业执行等。

（2）跨系统数据一致性验证

针对不同应用系统传递的数据，平台应能进行数据一致性验证，保证不同数据库（如 MySQL、Oracle、DB2 等）、不同类型文件（如 dat、del、json 等）传输的数据文件加载前后内容不变。具体测试点如下。

一是验证不同应用系统字符集格式是否一致，包括 GBK、UTF-8、UTF-16 互相加载时出现乱码的场景。另外，针对不同字符集，上游系统应基于待测作业所在数据库，导出下游可以加载的字符集数据。

二是验证跨系统精度是否一致，包括字符型、字节型数据库加载金额、汇率等字段是否会发生截断、能否处理以科学记数法存储的数据。针对不同数据库，下游系统应基于上游系统的存储类型构建适配的数据表。针对下游不能加载的数据格式，上游应能进行精度换算处理。

三是验证不同系统空值、字段分隔符是否一致。针对不同表示的空值，如 Nan、NULL，以及默认分隔符，如^O、^O^M。在下

游系统加载前，平台应预加载数据文件并进行适配下游系统数据库的转换。

四是验证不同测试环境下数据文件是否一致。针对下发到不同测试环境的、同一作业的数据文件，通过抽样验证其中数据是否一致。验证时，抽样率应大于等于 5%，并对金额类字段做 100%一致性比对。

（3）跨部门协作及时性验证

测试任务发起到办结的过程中，会在开发测试部门之间流转。具体环节包括发起测试任务、确定待测作业清单、作业依链路执行、处理链路作业报错（若有）、反馈测试结果、办结测试任务。其中，测试与开发确认任务待测作业清单、多层依赖确认前置是否执行、找开发人处理链路问题作业三个环节，可能出现跨部门协作责任不明确问题。为此作约束如下。

一是测试任务由测试人员发起，发起后通过平台自动绑定任务待测作业。

二是待测作业具有唯一确认的作业开发人、链路前置作业具有唯一确认的作业负责人。

三是链路测试由测试人员独立完成，但当出现前置作业或待测作业报错时，测试人员可为问题作业负责人推送待办。

四是为保障操作合规性，所有操作均留存审计日志（包括操作人、操作内容、操作时间、涉及环境等），用于合规检查。

五是为控制操作人权限，测试人员仅可对本人录入任务造空

文件，并具备该任务链路作业的只读执行权；开发人员仅开放待测作业的脚本修改、日志查看与执行权限；链路前置作业负责人仅开放所负责作业的脚本修改权。

六是为保障跨部门协作处理问题及时性，设置测试点待办处理及时率。待办处理及时率=1-（处理不及时待办数/总解决待办数）×100%。其中负责人未及时处理待办超4小时，则标记为处理不及时。待办处理及时率纳入绩效考核指标，每月待办处理及时率应达到95%以上。

3.处理性能测试规范

处理性能测试规范主要包括系统性能测试、业务场景测试和性能测试指标三方面内容，重点验证大数据平台在长时间运行、高并发负载、批量处理及异常恢复等场景下的稳定性、处理能力和业务支撑能力。通过在仿真生产环境中进行性能测试，识别系统性能瓶颈，验证系统资源承载能力、故障恢复能力和业务连续运行能力。

（1）系统性能测试

系统性能测试用于评估大数据平台在高负载、长时间运行及极端条件下能否稳定运行。测试环境应尽可能与生产环境保持一致，操作系统、中间件、数据库等关键参数配置应与生产环境一致。基础数据量应与生产环境保持同数量级，或结合测试资源与生产资源配比等比例铺设测试数据，并充分考虑未来三年业务增长趋势，确保测试结果具备参考价值。

在系统资源方面，重点关注CPU、内存交换区、磁盘和网络等关键资源使用情况。测试过程中，CPU利用率原则上应不高于80%，SWAP利用率不高于70%，磁盘繁忙率不高于70%，网络吞吐量不高于最大能力的70%。对于软硬件故障、升级维护等可能导致系统不可用的特殊场景，应验证系统恢复能力，必要时可引入混沌工程方式模拟异常情况。对于批量处理系统，还测试数据装载能力、并行计算能力和I/O吞吐能力。

（2）业务场景测试

业务场景测试时应结合生产运行特征，优先选取高频、高风险或增长趋势明显的典型业务场景，并参考历史峰值交易量、渠道分布、批量处理窗口等生产数据进行测试设计。根据测试目标和场景重要程度，业务场景测试可分为必须执行测试和可选测试两类。

必须执行测试主要包括单交易基准测试、单交易负载测试、单交易梯度测试、混合交易负载测试、混合交易稳定性测试、批量连通性测试和批量铺数预期结果测试。其中，单交易基准测试用于测量单类型交易在无负载条件下的基础性能表现；单交易负载测试用于评估单类型交易在不同负载条件下的性能变化；单交易梯度测试用于通过逐步增加负载观察系统性能拐点；混合交易负载测试用于验证多交易类型并发情况下的系统处理能力；混合交易稳定性测试用于评估系统长时间运行下的稳定性；批量连通性测试用于检验批量处理任务与相关系统之间的连通能力；批量

铺数预期结果测试用于验证批量数据处理结果的正确性和时效性。

可选测试主要包括混合交易压力测试、业务突变测试、可靠性测试，以及批量测试对混合交易影响测试。其中，混合交易压力测试用于评估系统在极端负载下的处理能力；业务突变测试用于模拟业务量突发增长时系统的适应能力；可靠性测试用于检验系统在异常条件下的容错和恢复能力；批量测试对混合交易影响测试用于分析批量处理任务对实时交易性能的潜在影响。

（3）性能测试指标

性能测试指标用于衡量系统处理能力和服务效率，主要包括交易处理能力、单笔业务响应时间和批量处理时效性等。转账类交易处理能力方面，商业银行系统应支持每百万客户不小于50笔/秒的处理能力，证券公司系统应支持每百万客户不小于150笔/秒的处理能力。查询类交易处理能力方面，商业银行系统应支持每百万客户不小于100笔/秒的处理能力，证券公司系统应支持每百万客户不小于150笔/秒的处理能力。单笔业务响应时间方面，单笔业务从发起至处理完成的最长时间应小于30秒。批量处理时效性方面，批量处理任务应在T+1自然日内完成结果计算与批量服务接口下发。

（三）实施路径

金融业务大数据测试流程，需遵循软件测试的通用规范，同时也需结合金融行业特性进行增强和优化。通过标准化测试流程，

规范大数据测试组织和实施方式，确保测试工作的全面性和有效性。具体过程如下。

1.制定测试计划

结合金融业务场景，明确大数据测试的目标，包括数据准确性、处理性能、系统稳定性、合规性等要求，收集系统功能、数据处理等方面的业务需求，制定测试范围和测试目标，编写和评审测试计划或方案。

2.测试分析设计

根据不同应用系统的功能需求、数据处理要求等，选取与之对应的测试策略进行测试要点分析提取，输出测试大纲、测试用例及测试脚本。

3.测试环境搭建

构建与生产环境相似的大数据测试环境，性能测试环境需模拟生产环境的硬件规模，功能测试环境的硬件配置可缩减。

4.测试数据准备

需区分功能测试与性能测试。功能测试数据需尽可能还原真实的业务场景，满足数据多样性要求；性能测试数据则需尽可能还原真实数据规模。

5.准入检查

确保在进行正式测试之前，一系列预定义的条件（包括测试环境、测试数据以及其他准入项）已经检查完毕并满足测试要求，从而保障测试能够顺利推进。

6.测试执行

结合实际需要依次开展冒烟测试、用户故事测试、回归测试等多轮测试发现系统问题，对发现的问题遵循软件测试的缺陷管理流程，确保缺陷可追溯、可复现，修复后需进行回归测试，确保问题解决且未引入新的风险。

7.准出检查

测试准出条件应包括：1）各机构内部软件测试体系要求：如测试用例通过率、缺陷修复率、各性能指标等达标；2）数据质量指标：如空值率、一致性、完整性、唯一性等满足业务和监管要求。

8.测试总结

在全部轮次用例执行完毕后，汇总测试结果，评估系统是否符合业务需求、性能指标及合规要求。遵循软件测试的评估标准，输出测试报告，包括测试情况、性能瓶颈、风险点及优化建议等。同时，还需结合监管合规要求，确保测试报告满足审计要求。

在大数据测试过程中，测试环境、测试数据和测试脚本的准备是区别于传统软件测试的关键环节。测试团队基于用户验收标准，完成测试环境准备和测试数据构造。在测试设计阶段，除常规测试用例外，还需编写对应的测试脚本，通过执行脚本实现源数据与目标数据的比对验证，通过白盒测试检查开发脚本，验证数据加工规则是否符合业务需求，双管齐下，保障数据的准确性和数据质量。

五、实践案例

（一）案例一：交通银行批量数据测试平台

1.案例背景

因金融行业涉及用户隐私、资金安全数据的特殊性，大型银行通常分设软件开发中心、测试中心、数据中心，实现开发、测试、运维工作的职责分离。通过架构异构，减少操作失误、恶意操作的风险，达到减免银行财务及声誉损失、便于追溯生产问题原因的效果。细分环境与职责提升了精确性的同时，也带来了协同效能不足的问题。交通银行数据计算平台承担全行数据加工处理的职责，也存在着协同效能低导致的环境割裂、流程阻滞、职责模糊等问题。

2.问题描述

一是环境碎片化。按业务需求独立使用的测试环境较多,(如核心下移、村镇行专项等)开发测试人员每天需在不同系统间反复切换。

二是跨部门协作断裂。开发测试环节均有人负责，但环节之间缺少衔接。存在协同等待成本高、责任人信息透明度低的问题，响应机制有待优化。

三是权限僵化。受权限管控影响，测试无法访问开发环境，执行作业需开发人员协助。开发兼任作业开发、业务答疑、配合测试职责，负荷过重。非核心工作影响了开发进度，测试进度也因等待依赖而效率降低。

3. 解决方案

具体地，平台开发了以下功能解决上述问题，如表 7 所示。

表7 平台功能模块

功能模块	对应问题	实现逻辑
环境统一门户	多个环境需频繁切换	申请各环境 API 对接权限; 单点登录与界面聚合;
作业依赖链路可视化	测试出错难定位根因作业	解析作业 DAG 依赖树; 可视化未运行、报错作业层级;
作业责任人关联显示	协同等待成本高、责任人信息透明度低	同步开发库元数据; 自动绑定作业－开发人员;
一键式测试操作	测试执行作业依赖开发	分级权限控制; 按钮化测试操作;

（1）环境统一门户

批量测试平台实现了全环境一站式访问，以及测试进度自动跟踪，如图2所示。

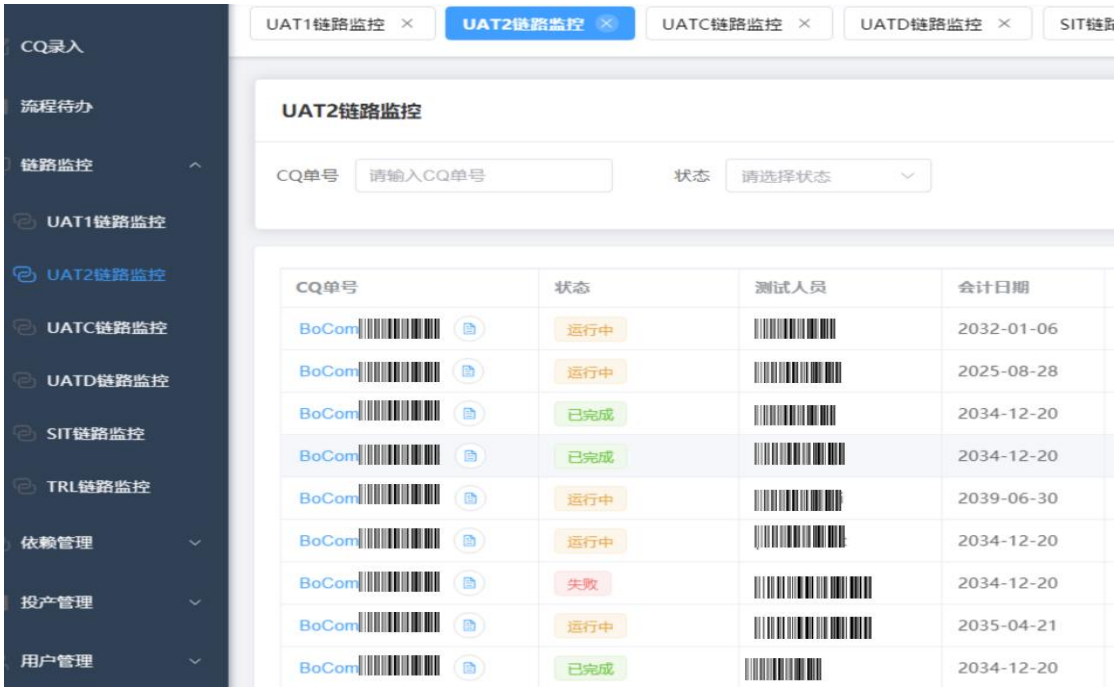

图 2 环境一站式访问并追踪测试进度

全环境一站式访问

打通了平台与 UAT、SIT 等各测试环境的访问通道，实现了执行权限统一管控。

通过脚本实时抓取各环境作业执行状态，并集中展示。

测试进度实时跟踪

通过同步开发库元数据，获取需求单与作业绑定关系。作业均测试通过后，单子测试状态自动更新。

实时判别测试进度，若单子下新增绑定作业，单子状态将同步更新，以免漏测作业。

（2）作业依赖可视化

批量测试平台可以图形化展示作业依赖关系，帮助开发测试快速定位测试失败根因，如图 3 所示。

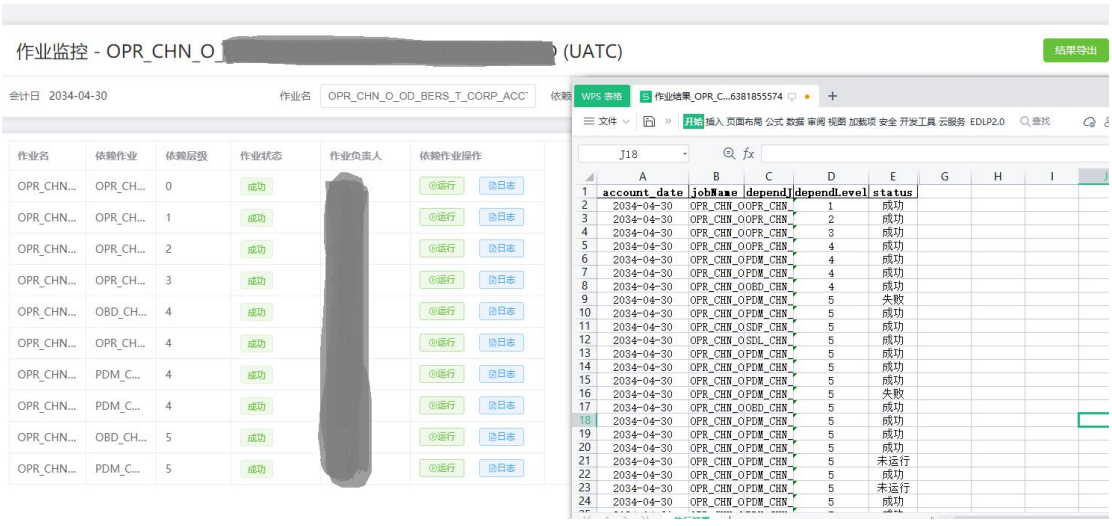

图 3 前置依赖作业可视化及运行情况导出

链路执行情况查看

若需求为待投产需求单，平台将自动识别其下各集市作业，

并拉取当前已绑定作业明细，可于作业监控界面按照依赖层级查询链路执行情况。

链路执行情况导出

开发作业状态导出功能，可以需求或作业为单位导出结果，方便跟进链路异常待测作业。

（3）作业责任人关联显示

平台通过同步开发库元数据，获取并展示待投产作业及存量作业负责人信息，使测试人员在作业报错时能直接定位联系对应开发人员，达到减少跨部门沟通成本，提升协同效率的效果。

① 于“链路监控”选择需求单对应环境，搜索待监控需求单。

② 选择需监控作业

③ 于作业监控界面查询整体链路执行情况。

（4）一键式测试操作

数据计算平台各测试环境与生产一致，均通过调度工具跑批。批量测试平台封装了测试高频操作，在权限隔离的前提下实现测试自主化。涉及功能包括如下。

文件造空

因测试环境不下发非待测文件，开发制造空文件功能，可批量或单次造空文件。待测作业上游文件已到达时，可通过造空功能自动跑批，如图 4 所示。

图 4 查看文件到达情况及造空功能

作业执行

当待测作业较少或涉及反复测试时，可通过作业执行功能，手动按链路执行作业，平台将显示各环境下作业最新执行结果，如图 5 所示。

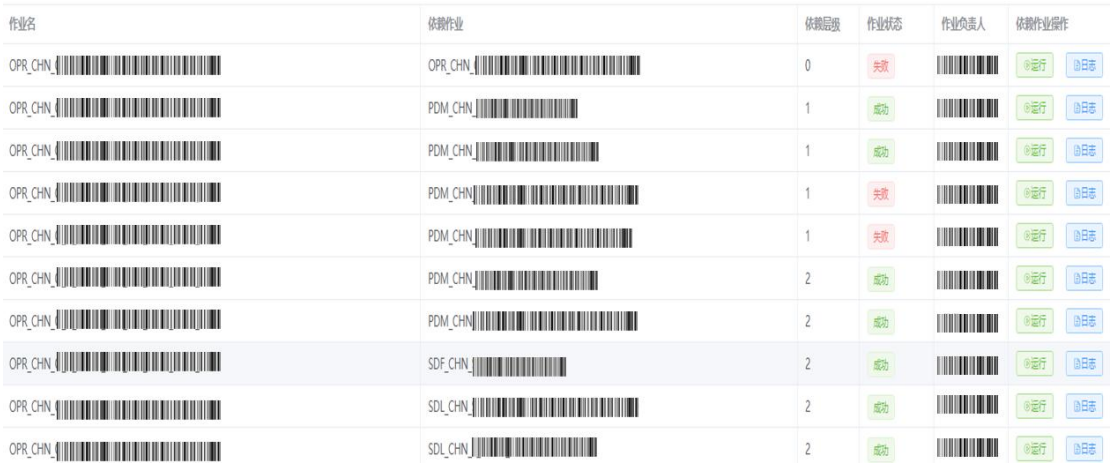

图 5 执行作业后作业状态始终显示最新结果

日志查看

当作业报错时，开发可于日志查看报错信息，如图 6 所示。

图 6 执行失败作业日志查看

4. 实施成效

应用批量测试平台后，开发测试人员的协同效能得到了有效提升，具体体现如下。

（1）测试周期压缩

需求单平均测试周期从 7—10 天缩短至 5—7 天，效率提升 20%—35%;

最佳案例可 3～4 天完成需求单测试（提速 50%），如表 8 所示;

表 8 测试周期压缩对比

指标	改造前	改造后	幅度
单任务沟通次数	3—5 次	≤2 次	-80%
问题响应等待时长	0.5—1 天	10 分钟	-99%
测试沟通阻滞总	1.8—3.96天	≈0	-100%

耗时			

（2）消除协作盲区

《批量测试平台使用手册》介绍功能之外，给出明确的链路测试步骤，及定位问题方法;

开发人员可专注于代码开发及报错修复，减少非核心工作负担;

测试人员可独立完成全流程测试，消除等待依赖。

（二）案例二：邮储银行大数据测试服务引擎

1.案例背景

随着金融科技的高速发展，银行业务模式正在经历从传统业务向数字化转型的深刻变革，保障数据质量是银行业提升服务质量、降低风险、增强竞争力的有力抓手。邮储银行深入贯彻国家“数字中国”战略，高度重视数据质量，重点打造基于运营决策、客户营销、风险管理与合规监测等多个大数据类工程建设。经过多年的大数据测试实战摸索，邮储银行在 TMMI5 体系框架指导下，引入端到端测试、分层测试等创新工艺，针对不同类型的系统定制差异化测试策略，构建标准化的大数据测试方法论，形成全面的大数据质量保障体系，有效填补了当前数据测试体系的空白。

由于大数据类工程数据与结构的复杂性，在测试阶段面临测试数据规模大、技术门槛高等挑战，测试人员往往需要熟练掌握多种主流大数据测试技术才能开展测试工作。为此，邮储银行自

主研发大数据测试服务引擎，技术架构如图 7 所示，该引擎通过自动化解决测试痛点，降低测试实施成本、促进大数据测试的高效验证，推动全行数据质量高效提升。

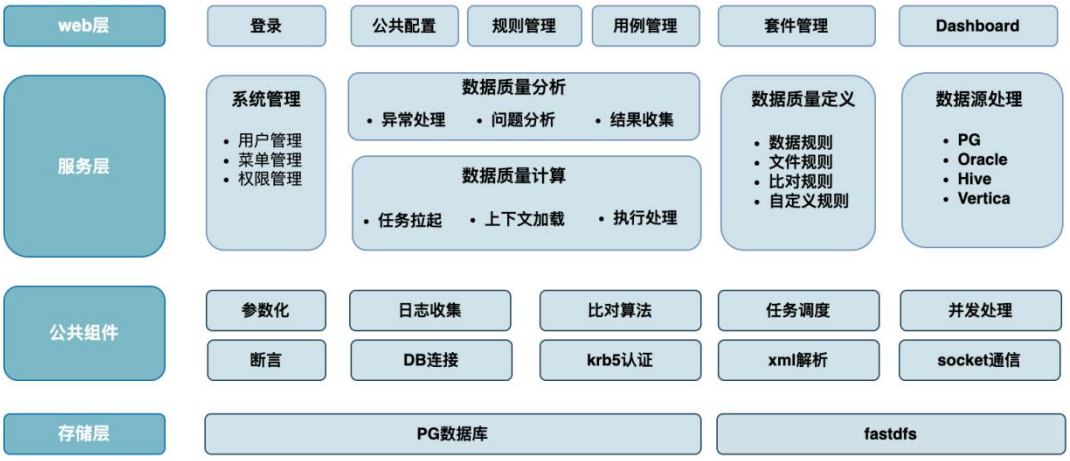

图 7 大数据测试服务引擎技术架构图

2. 解决方案

数据测试主要关注数据的完整性、准确性、一致性，在测试上主要采取规则校验和数据比对两种方法。规则校验通过对单表数据按照不同规则或约束编写校验 sql，并检查 sql 执行结果与期望是否符合，从而验证数据的准确性；数据比对通过对源表或文件数据与目的表中数据进行一一比对，验证数据加工过程中易出现的一致性或完整性问题。在设计实现了登录、权限管理、公共配置等基础功能外，还设计了以下功能：

多维度规则校验：通过对规则抽象和封装，以模型化的设计思路支持如字段格式校验、非空校验、逻辑校验在内的几十种数据规则验证，帮助测试人员简单快捷地扩展新的规则。

细粒度数据比对：支持同构、异构数据源字段级的比对，支持用户自定义比对规则，并能明确定位出不一致的数据记录。

全量数据文件检核：基于数据接口文档，对业务系统上送给大数据的数据文件的格式和内容进行校验，支持源头数据治理。

自动化批量执行：支持将迭代内测试用例放在一个套件中以自动化方式执行，同时提供套件维度的看板展示，支持层层下探查每条用例的执行结果。

3.应用实践

大数据测试服务引擎实现从测试脚本编写、执行、报告展示全流程线上化操作。目前已经在邮储银行大数据平台、数据仓库、数据中台等十余个大数据类系统中落地使用，同时拓展应用到信用卡核心等系统，累计编写数据测试脚本千余次，执行近万次，最大比对数据量达千万级，大幅提高了测试覆盖率和充分性，保障了大数据类系统建设质量。

（1）规则校验

当需要验证目标表中多个字段同时满足主键字段 A 唯一、字段 B 不为空、当字段 C=“女”时，字段 D>=20、字段 E 枚举值正确 4 条规则，相比于手工执行需编写 4 个 sql 脚本并执行 4 次，大数据测试服务引擎仅需勾选 4 个规则模板并配置参数，简单灵活，还可以实现一次配置多次执行，降低人力成本，如图 8、9 所示。

图 8 规则校验系统操作图

图 9 规则校验执行结果图

（2）数据比对

当表 A 中的字段由源表 B、C 中的字段加工而来，或数据经 A 库加工完成后进入 B 库以及新旧系统数据迁移等测试场景，需要验证数据的一致性和加工逻辑的准确性。测试人员编写源表测试 SQL 以及目标表查询 SQL，执行自动化数据比对，如图 10、11 所示。相比人工测试只能采用抽样比对覆盖率较低，大数据测试服务引擎实现全表全字段比对，几十万条数据比对可在分钟级时

间内完成，大大提高了执行效率，且相比于肉眼检查，测试结果更精确。

图 10 数据比对系统操作图

图 11 数据比对执行结果图

（3）数据文件测试

数据文件测试功能主要针对通用的数据文件格式和内容进行自动化检核，单次文件检验可从小时级缩短到秒级。目前已在信用卡核心系统、大数据类系统、商业票据等源业务系统中落地使用，累计检验数据文件百余份，有效节约了测试时间和人力成本。

以源业务系统的数据文件测试场景为例，如下图 12 所示，根据业务逻辑自定义配置校验规则，针对系统内涉及日终任务、数据报送等需进行数据文件发送、接收、数据内容核对的功能进行测试，自动完成数据文件校验。

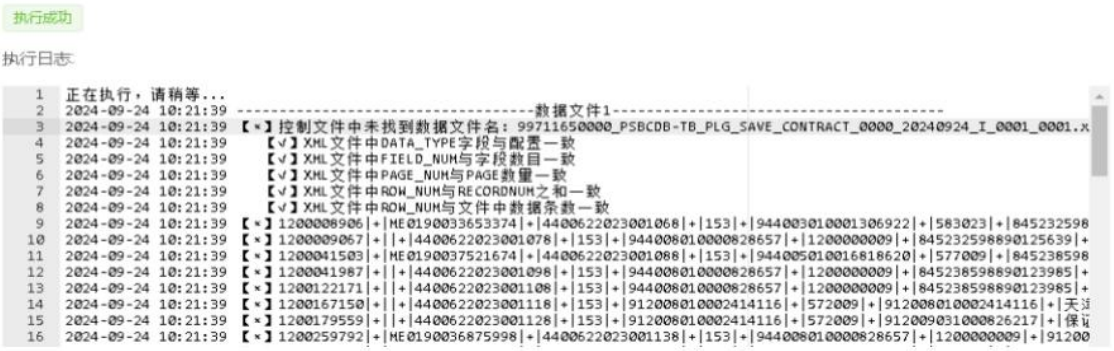

图 12 数据文件测试执行结果示例

（4）套件集批量执行

当目标表包含多个字段的规则校验以及数据比对，或者字段变更加工逻辑后需要开展回归测试以及动态数据检测，使用套件集自动执行结果更加直观，可以形成可视化、可追溯的测试报告和数据统计，并支持并行和串行执行两种方式，如图 13、14 所

示。此外，基于套件集可以自动开展批量数据质量监控，实现数据质量提升。

图 13 套件集执行系统操作图

图 14 套件集执行系统操作图

六、趋势与展望

金融业大数据测试正加速迈入标准化体系完善、技术融合纵深发展、监管要求持续升级的全新阶段。未来，测试模式将由传统的“通过式检测”转向持续验证、智能分析与协同保障并重的新范式。依托测试环境的共建共享、场景复用以及经验成果的系统沉淀，行业将实现质量与效能的双重跃升，系统地推动测试能力向标准化、协同化、智能化方向演进。

（一）模式重塑：从事后验证到全生命周期保障

随着 DataOps 体系在金融领域的深度落地与成熟应用，金融大数据测试正朝着“风险可控、合规先行、数据可信”的核心方向持续进阶，全面融入金融业务全流程，着力构建专业化、精准化的测试新生态。DataOps 所倡导的协作化、自动化、持续迭代理念，正推动金融大数据测试摆脱传统事后验证模式，转向覆盖交易结算、实时风控、监管报送、信贷审批等核心场景的全生命周期保障，为金融机构的数据安全与业务稳健运行筑牢双重防线。

（二）技术赋能：人工智能驱动测试实现智能闭环

人工智能正在重塑金融数据测试的技术路径，实现从“人工主导”到“智能闭环”的转型。生成式 AI 与自然语言处理技术结合，可自动解析金融业务文档，生成符合逻辑的测试用例与零代码脚本，实现测试全流程自动化。在测试数据生成上，生成式 AI 通过学习金融数据分布特征，能够产出高保真测试数据，有效弥补传统方法在场景覆盖全面性和数据真实性上的不足。

（三）场景拓展：从传统业务向新兴领域纵深演进

金融业务创新正推动测试场景从“传统业务”向“新兴领域”延伸，测试内容日趋专业化和复杂化。监管部门对风险防控“时效性”要求的提升，促使实时风控与动态合规测试向动态化、实时化升级，测试场景亦从“静态参数”向“宏观经济联动”演进。同时，金融大模型在多领域的广泛应用，催生了对其可靠性、安全性与合规性的专项测试需求，行业正逐步构建多维度的大模型金融应用评估框架。

（四）合规前置：从事后验证到全程监控的深度变革

金融监管“严监管、全覆盖”态势持续深化，推动金融大数据测试从“事后验证”向“事前预防、事中监控”前移，合规测试的深度与广度显著增强。“测试左移”策略成为行业普遍选择，将合规验证嵌入需求分析与代码开发阶段，有效降低缺陷修复成本与周期。压力测试作为监管评估金融机构风险抵御能力的核心工具，正与监管科技深度融合，构建“测试－监管－优化”闭环，提升测试效率与精准性。此外，面对金融全球化趋势，跨境数据流动需满足多法域合规要求，推动跨境数据合规测试向标准化方向迈进，以更好应对复杂数据合规挑战。

（五）AI 赋能：从前瞻应用到智能协同

当前，人工智能技术正从宏观趋势走向工程实践，其在金融业大数据测试领域的融合已从理论探索迈入应用落地阶段。前瞻性探索聚焦于将 AI 嵌入作业开发、测试及上线的全流程，构建

“设计－执行－分析－报告”的智能闭环。在此背景下，金融业大数据测试 AI 智能体应运而生，部署于 CI/CD 流水线中：在持续集成阶段，智能体于新版本部署完成后自动触发，模拟测试人员工作流，自动解析变更请求并关联待测数据作业，测试人员仅需确认归属，智能体即接管后续流程，自动分析建表及跑批脚本、生成高覆盖度测试用例与仿真数据；在执行验证环节，智能体实时监控执行过程，自动解析日志并进行智能分类与根因分析，精准定位缺陷来源并自动提交缺陷工单，大幅压缩协同响应时间；在集成周期结束时，智能体对增量程序进行全量验证并生成结构化测试报告，涵盖代码覆盖范围及合规性初步核验，为测试人员提供决策支持。这一实践将标准化、重复性测试工作自动化，实现了从“人工逐项核对”向“自动筛查+重点复核”的转变，显著提升测试效率，为数据安全、权限管理及信创适配等挑战提供了自动化解决方案。