<面向工业制造领域的多模态数据治理平台>

测试计划

版本 <2.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <15/10/23> | <1.0> | 根据立项建议书和需求规约编写测试计划文档 | 曾宇欣、钱麒丹、陈治源 |
| <16/11/23> | <1.2> | 补充并完善测试计划描述、整理复核 | 曾宇欣、陈诺、张瑞轩 |
| <3/1/24> | <2.0> | 根据上课内容完善测试计划内容 | 钱麒丹、陈治源、陈诺 |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

1.4 项目标识 6

2. 测试需求 7

3. 测试策略 8

3.1 测试类型 8

3.1.1 数据和数据库完整性测试 8

3.1.2 功能测试 8

3.1.3 业务周期测试 9

3.1.4 用户界面测试 10

3.1.5 性能评价 11

3.1.6 负载测试 12

3.1.7 强度测试 13

3.1.8 容量测试 14

3.1.9 安全性和访问控制测试 15

3.1.10 故障转移和恢复测试 16

3.1.11 配置测试 18

3.1.12 安装测试 19

3.2 工具 20

4. 资源 21

4.1 角色 21

4.2 系统 23

5. 项目里程碑 24

6. 可交付工件 25

6.1 测试模型 25

6.2 测试日志 25

6.3 缺陷报告 25

7. 附录 A：项目任务 27

测试计划

# 简介

## 目的

<面向工业制造领域的多模态数据治理平台> 的这一“测试计划”文档有助于实现以下目标：

• 确定现有项目的信息和应测试的软件构件

• 列出推荐的测试需求

• 推荐可采用的测试策略，并对这些策略加以说明

• 确定所需的资源，并对测试的工作量进行估计

• 列出测试项目的可交付元素

## 背景

随着工业制造的发展，工业产品的生产工艺与流程日趋复杂，越来越多的制造业面临着跨地域、跨行业及跨模式等问题，工业项目管理的范围与难度逐步加大。为了解决工业生产方式中的问题，德国率先提出了工业4.0战略。此后，我国开始密切关注工业领域与信息领域高度融合的急迫性和价值。2020 年 5 月， 工信部发布《关于工业大数据发展的指导意见》，推动工业数据全面采集，加速工业设备互联互通，推动工业数据高质量汇聚，统筹建立国家工业大数据平台， 推动工业数据开放共享，激发工业数据市场活力，深化数据应用，完善数据治理。

企业的发展往往伴随着业务更加多元化，这也必然会推动更多的业务数据产生。在当今数字化时代，数据作为企业最重要的资产之一，其高效管理和治理对企业的长期发展至关重要。然而，随着业务的扩展和数据量的不断增长，也带来了一系列与数据存储、数据模型建设、数据质量、使用规范等有关数据治理的问题。不同的行业信息化发展水平不一样，其对数据的依赖程度不一样，数据治理水平也不一样。工业领域的信息化起步相对较晚，工业数据也更为复杂，涉及研发、生产、管理、运维、服务等多个环节，数据治理工作的推进也面临诸多挑战。

首先，大多数工业企业数据基础薄弱，管理手段落后，公司数据散落在不同的部门、系统和平台中，导致数据孤岛问题普遍存在；其次，企业缺乏有效的数据管理流程，数据录入方式缺乏规范，使得数据质量难以保障，数据共享困难； 加之工业数据来源的多样性，既有经营管理的数据，也有工艺流程的数据，更有 多种设备状态、控制数据，不同种类的数据特点要求数据的采集、清洗、存储和处理的技术不一样，进一步提升了工业数据治理的难度。

为了缓解以上问题，解决应用和应用间的信息共享问题，盘活整条数据链路， 保障企业数据的采集、存储、计算和使用过程的可控性和可追溯性，项目组拟设计并实现一个针对工业企业的智能数据治理平台，通过规范化、标准化和监控数据的全生命周期，提高数据的质量、确保数据的安全性，促进数据共享和决策。 工业智能数据治理平台需要结合工业特有的数据源、业务需求和智能化技术，通过元数据管理、数据标准管理、数据治理管理、主数据管理和数据资产管理五大模块全方位保障数据质量，改善数据决策能力，提升数据可信度，防范数据风险。

## 范围

单元测试：对各个模块的源代码的进行测试，保证各模块基本功能都能够正确地实现。

集成测试：将模块进行组合测试，保证所有功能都能够正确地实现。

系统测试：根据《需求规格说明书》对软件进行功能测试，对重点的模块进行性能测试，并结合可能的用户测试。

功能性测试：按照需求定义中的功能定义部分对系统实行的系统级别的测试。

非功能性测试：按照系统需求定义中的非功能定义部分（如系统的性能指标、可靠性指标等）对系统实行的系统级别的测试。

测试对象中将接收测试的功能包括微服务的定义与部署，工艺流程的定义、解析与执行，消息通知、状态监控等 。

测试对象中将接收测试的非功能特性包括易用性、性能、可支持性等。

编写此文档的过程中假设测试环境有稳定的网络连接，且测试人员具备基本的软件测试技能。

可能会影响测试设计、开发或实施的约束、风险和意外事件：测试人员不具备专业的技能，在测试过程中可能出现疏漏和不严谨之处；由于设备和资源的约束，有些非功能性需求的测试不够精准；有些原计划需要进行测试的功能可能随着开发的进行被删去，这些功能将不再需要测试。

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 曾宇欣/陈诺 |  |
| 功能性规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 钱麒丹/陈治源 |  |
| 用例报告 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 曾宇欣/张瑞轩 |  |
| 项目计划 | 🗹是 🞏 否 | 🗹 是 🞏 否 | 钱麒丹/陈诺 |  |
| 设计规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 曾宇欣/陈治源 |  |
| 业务模型或业务流程 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 陈治源/陈诺 |  |
| 数据模型或数据流 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 张瑞轩/陈诺 |  |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

|  |  |
| --- | --- |
| 功能性需求 | 注册 |
| 登陆 |
| 用户管理 |
| 数据模型管理 |
| 异构数据源接入 |
| 标签描述定义 |
| 质量规则定义 |
| 质量规则管理 |
| 数据清洗 |
| 数据加密 |
| 访问控制管理 |
| 数据脱敏 |
| 数据资产目录 |
| 数据资产检索 |
| 数据资产导出 |
| 数据监控 |
| 数据地图 |
| 报表生成 |
| 非功能性需求 | 数据库接口 |
| 性能需求 |
| 兼容性 |
| 负载测试 |
| 用户界面 |
| 安全性和访问控制 |

# 测试策略

## 测试类型

### 数据和数据库完整性测试

|  |  |
| --- | --- |
| 测试目标： | 确保数据库访问方法和进程正常运行，数据不会遭到损坏。 |
| 方法： | • 调用各个数据库访问方法和进程，并在其中填充有效的和无效 的数据或对数据的请求。  • 检查数据库，确保数据已按预期的方式填充，并且所有 数据库事件都按正常方式出现；或者检查所返回的数据，确保为 正当的理由检索到了正确的数据 |
| 完成标准： | 所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。 |
| 需考虑的特殊事项： | • 测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接 输入或修改数据。  • 进程应该以手工方式调用。  • 应使用小型或最小的数据库（其中的记录数很有限）来 使所有无法接受的事件具有更大的可见性。 |

### 功能测试

|  |  |
| --- | --- |
| 测试目标： | 确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。 |
| 方法： | 利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。 |
| 完成标准： | • 所计划的测试已全部执行。  • 所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | 确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的） |

### 业务周期测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。 |
| 方法： | 通过执行以下活动，测试将模拟若干个业务周期：  • 将修改或增强对测试对象进行的功能测试，以增加每项功能的执 行次数，从而在指定的时段内模拟若干个不同的用户。  • 将使用有效的和无效的日期或时段来执行所有与时间或日期相关 的功能。  • 将在适当的时候执行或启动所有周期性出现的功能。  • 在测试中还将使用有效的和无效的数据，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。 |
| 完成标准： | • 所计划的测试已全部执行。  • 所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | • 系统日期和事件可能需要特殊的支持活动  • 需要通过业务模型来确定相应的测试需求和测试过程。 |

### 用户界面测试

|  |  |
| --- | --- |
| 测试目标： | 核实以下内容：  • 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括 窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法 （Tab 健、鼠标移动和快捷键）的使用  • 窗口的对象和特征（例如：菜单、大小、位置、状态和 中心）都符合标准。 |
| 方法： | 为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。 |
| 完成标准： | 证实各个窗口都与基准版本保持一致，或符合可接受标准 |
| 需考虑的特殊事项： | 并不是所有定制或第三方对象的特征都可访问。 |

### 性能评价

|  |  |
| --- | --- |
| 测试目标： | 核实所指定的事务或业务功能在以下情况下的性能行为：  • 正常的预期工作量  • 预期的最繁重工作量 |
| 方法： | • 使用为功能或业务周期测试制定的测试过程。  • 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项 事务的迭代次数。  • 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为 基准），并在多台客户机（虚拟的或实际的客户机，请参见下面 的“需考虑的特殊事项”）上重复。 |
| 完成标准： | • 单个事务或单个用户：在每个事务所预期或要求的时间范围内 成功地完成测试脚本，没有发生任何故障。]  • 多个事务或多个用户：在可接受的时间范围内成功地完成测试 脚本，没有发生任何故障。 |
| 需考虑的特殊事项： | 综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：  • 直接将“事务强行分配到”服务器上，这通常以“结构化查询语 言”(SQL) 调用的形式来实现。  • 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客 户机。 此负载可通过“远程终端仿真”(Remote Terminal Emulation) 工具来实现。 此技术还可用于在网络中加载“流 量”。  • 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添 加负载。  性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。 |

### 负载测试

|  |  |
| --- | --- |
| 测试目标： | 核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。 |
| 方法： | • 使用为功能或业务周期测试制定的测试。  • 通过修改数据文件来增加事务数量，或通过修改测试来增加每项 事务发生的次数。 |
| 完成标准： | 多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。 |
| 需考虑的特殊事项： | • 负载测试应该在专用的计算机上或在专用的机时内执行，以便 实现完全的控制和精确的评测。  • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数 据库。 |

### 强度测试

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象能够在以下强度条件下正常运行，不会出现任何错误：  • 服务器上几乎没有或根本没有可用的内存（RAM 和 DASD）  • 连接或模拟了最大实际（或实际可承受）数量的客户机  • 多个用户对相同的数据/账户执行相同的事务  • 最繁重的事务量或最差的事务组合（请参见上面的“性能测 试”）。 |
| 方法： | • 使用为性能评价或负载测试制定的测试。  • 要对有限的资源进行测试，就应该在一台计算机上运行测试，而 且应该减少或限制服务器上的 RAM 和 DASD。  • 对于其他强度测试，应该使用多台客户机来运行相同的测试或互 补的测试，以产生最繁重的事务量或最差的事务组合。 |
| 完成标准： | 所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。 |
| 需考虑的特殊事项： | • 如果要增加网络工作强度，可能会需要使用网络工具来给网络加 载消息或信息包。  • 应该暂时减少用于系统的 DASD，以限制数据库可用空间的增 长。  • 使多个客户机对相同的记录或数据账户同时进行的访问达到同 步。 |

### 容量测试

|  |  |
| --- | --- |
| 测试目标： | [核实测试对象在以下大容量条件下能否正常运行：  • 连接（或模拟了）最大（实际或实际可承受）数量的客户机，所 有客户机在长时间内执行相同的、且情况（性能）最差的业务功 能。  • 已达到最大的数据库大小（实际的或按比例缩放的），而且同时 执行了多个查询或报表事务。 |
| 方法： | • 使用为性能评价或负载测试制定的测试。  • 应该使用多台客户机来运行相同的测试或互补的测试，以便在长 时间内产生最繁重的事务量或最差的事务组合（请参见上面的 “强度测试”）。  • 创建最大的数据库大小（实际的、按比例缩放的、或输入了代表 性数据的数据库），并使用多台客户机在长时间内同时运行查询 和报表事务。 |
| 完成标准： | • 所计划的测试已全部执行，而且在达到或超出指定的系统限制 时没有出现任何软件故障。 |
| 需考虑的特殊事项： | 对于上述的大容量条件，可以接受的时间段。 |

### 安全性和访问控制测试

|  |  |
| --- | --- |
| 测试目标： | * 应用程序级别的安全性：核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。 * 系统级别的安全性：核实只有具备系统和应用程序访问权限的主角才能访问系统和应用程序。 |
| 方法： | * 应用程序级别的安全性：确定并列出各用户类型及其被授权使用的功能或数据。   • 为各用户类型创建测试，并通过创建各用户类型所特有的事务来核实其权限。  • 修改用户类型并为相同的用户重新运行测试。对于每种用户类型，确保正确地提供或拒绝了这些附加的功能或数据。   * 系统级别的访问：请参见下面的“需考虑的特殊事项”*。* |
| 完成标准： | 各种已知的主角类型都可访问相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。 |
| 需考虑的特殊事项： | 无 |

### 故障转移和恢复测试

|  |  |
| --- | --- |
| 测试目标： | 确保恢复进程（手工或自动）将数据库、应用程序和系统正确地恢复到了预期的已知状态。测试中将包括以下各种情况：  • 客户机断电  • 服务器断电  • 通过网络服务器产生的通信中断  • DASD 和/或 DASD 控制器被中断、断电或与 DASD 和/ 或DASD 控制器的通信中断  • 周期未完成（数据过滤进程被中断，数据同步进程被中 断）。  • 数据库指针或关键字无效  • 数据库中的数据元素无效或遭到破坏 |
| 方法： | 应该使用为功能和业务周期测试创建的测试来创建一系列的事务。一旦达到预期的测试起点，就应该分别执行或模拟以下操作：  • 客户机断电：关闭 PC 的电源。  • 服务器断电：模拟或启动服务器的断电过程。  • 通过网络服务器产生的中断：模拟或启动网络的通信中 断（实际断开通信线路的连接或关闭网络服务器或路由 器的电源）。  • DASD 和 DASD 控制器被中断、断电或与 DASD 和 DASD 控制器的通信中断：模拟与一个或多个 DASD 控 制器或设备的通信，或实际取消这种通信。  一旦实现了上述情况（或模拟情况），就应该执行其他事务。而且一旦达到第二个测试点状态，就应调用恢复过程。  在测试不完整的周期时，所使用的方法与上述方法相同，只不过应异常终止或提前终止数据库进程本身。  对以下情况的测试需要达到一个已知的数据库状态。当破坏若干个数据库字段、指针和关键字时，应该以手工方式在数据库中（通过数据库工具）直接进行。其他事务应该通过使用“应用程序功能测试”和“业务周期测试”中的测试来执行，并且应执行完整的周期。 |
| 完成标准： | 在所有上述情况中，应用程序、数据库和系统应该在恢复过程完成时立即返回到一个已知的预期状态。此状态包括仅限于已知损坏的字段、指针或关键字范围内的数据损坏，以及表明进程或事务因中断而未被完成的报表。 |
| 需考虑的特殊事项： | • 恢复测试会给其他操作带来许多的麻烦。断开缆线连接 的方法（模拟断电或通信中断）可能并不可取或不可 行。所以，可能会需要采用其他方法，例如诊断性软件 工具。  • 需要系统（或计算机操作）、数据库和网络组中的资 源。  • 这些测试应该在工作时间之外或在一台独立的计算机上 运行。 |

### 配置测试

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象可在要求的硬件和软件配置中正常运行。 |
| 方法： | • 使用功能测试脚本。  • 在测试过程中或在测试开始之前，打开各种与非测试对 象相关的软件（例如 Microsoft 应用程序：Excel 和 Word），然后将其关闭。  • 执行所选的事务，以模拟主角与测试对象软件和非测试 对象软件之间的交互。  • 重复上述步骤，尽量减少客户机工作站上的常规可用内 存。 |
| 完成标准： | 对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。 |
| 需考虑的特殊事项： | • 需要、可以使用并可以通过桌面访问的非测试对象软 件  • 通常使用的应用程序。  • 应用程序正在运行的数据。例如，在Excel 中打开的 大型电子表格，或是在 Word 中打开的 100 页文档。  • 作为此测试的一部分，应将整个系统、Netware、网络服 务器、数据库等都记录下来。 |

### 安装测试

|  |  |
| --- | --- |
| 测试目标： | 核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中：  • 首次安装。以前从未安装过 <面向工业制造领域的多模态数据治理平台> 的新计算机  • 更新。以前安装过相同版本的 <面向工业制造领域的多模态数据治理平台> 的计算机  • 更新。以前安装过较早版本的 <面向工业制造领域的多模态数据治理平台> 的计算机 |
| 方法： | • 手工开发脚本或开发自动脚本，以验证目标计算机的状 况⎯ 新 - <面向工业制造领域的多模态数据治理平台> 从未安装过；已安装 <面向工业制造领域的多模态数据治理平台> 相同或较早版本）。  • 启动或执行安装。  • 使用预先确定的功能测试脚本子集来运行事务。 |
| 完成标准： | <面向工业制造领域的多模态数据治理平台> 事务成功执行，没有出现任何故障。 |
| 需考虑的特殊事项： | 应该选择 <面向工业制造领域的多模态数据治理平台> 的哪些事务才能准确地测试出 <面向工业制造领域的多模态数据治理平台> 应用程序已经成功安装，而且没有遗漏主要的软件构件 |

## 工具

此项目将使用以下工具：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 测试管理 | JIRA | Atlassian |  |
| 缺陷跟踪 | JIRA | Atlassian |  |
| 用于功能性测试的 ASQ 工具 | JMeter | Apach | 3.2 |
| 用于性能测试的 ASQ 工具 | JMeter | Apach | 3.2 |
| 测试覆盖监测器或评价器 | JMeter | Apach | 3.2 |
| 项目管理 | 飞书 | 字节跳动 |  |
| DBMS 工具 | Datagrid | MicroSoft |  |

# 资源

## 角色

下表列出了在此项目的人员配备方面所作的各种假定。

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理，  测试项目经理 | 1 | 进行管理监督。  职责：   * 提供技术指导 * 获取适当的资源 * 提供管理报告 |
| 测试设计员 | 1 | 确定测试用例、确定测试用例的优先级并实施测试用例。  职责：   * 生成测试计划 * 生成测试模型 * 评估测试工作的有效性 |
| 测试员 | 1 | 执行测试。  职责：   * 执行测试 * 记录结果 * 从错误中恢复 * 记录变更请求 |
| 测试系统管理员 | 1 | 确保测试环境和资产得到管理和维护。  职责：   * 管理测试系统 * 授予和管理角色对测试系统的访问权 |
| 数据库管理员 | 1 | 确保测试数据（数据库）环境和资产得到管理和维护。  职责：   * 管理测试数据（数据库） |
| 设计员 | 1 | 确定并定义测试类的操作、属性和关联。  职责：   * 确定并定义测试类 * 确定并定义测试包 |
| 实施员 | 1 | 实施测试类和测试包，并对它们进行单元测试。  职责：   * 创建在测试模型中实施的测试类和测试包 |

## 系统

下表列出了测试项目所需的系统资源。

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 | Localhost:dboard |
| —网络或子网 | 192.168.0.131 |
| —服务器名服务器名 |  |
| —数据库名 | TEST |
| 客户端测试 PC | MacOS Catalina |
| —包括特殊的配置需求 | 无 |
| 测试存储库 | jdbc:postgresql://124.222.140.214:5666/data\_management |
| —网络或子网 | TBD |
| —服务器名服务器名 | 121.37.220.123 |
| 测试开发 PC | MacOS Catalina |

# 项目里程碑

[对 <项目名称> 的测试应包括上面各节所述的各项测试的测试活动。应该为这些测试确定单独的项目里程碑，以通知项目的状态和成果。]

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 15人天 | <14/10/23> | <16/10/24> |
| 设计测试 | 15人天 | <2/11/23> | <20/11/23> |
| 实施测试 | 5人天 | <2/12/23> | <6/1/24> |
| 执行测试 | 5人天 | <27/12/23> | <6/1/24> |
| 评估测试 | 10人天 | <7/1/24> | <9/1/24> |

# 可交付工件

[本节列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。]

|  |  |
| --- | --- |
| 文档 | 作者 |
| 《测试计划》 | 曾宇欣 |

## 测试模型

[本节确定将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用 ASQ 工具来创建或引用。]

数据治理策略文档：通过ASQ工具创建，详细说明数据治理的原则、流程和标准。这包括数据的获取、存储、处理、安全性和隐私保护等方面的规范。

多模态数据处理流程图：使用ASQ工具设计流程图，展示不同类型数据（如文本、图像、声音等）是如何被整合、处理和分析的。

系统架构设计文档：通过ASQ工具创建，详细描述系统的架构设计，包括硬件和软件的配置、网络结构以及各组件之间的交互方式。

性能测试报告：使用ASQ工具生成，展现平台在处理大规模多模态数据时的性能表现，包括处理速度、准确性、资源消耗等关键指标。

安全性和合规性报告：通过ASQ工具编制，评估平台在数据安全性和合规性方面的表现，包括数据加密、访问控制和遵守相关法律法规的情况。

用户反馈和改进建议文档：使用ASQ工具收集和整理，包含从初步用户测试中获得的反馈和对平台可能的改进方向的建议。

## 测试日志

[说明用来记录和报告测试结果和测试状态的方法和工具。]

日志记录方法：使用自动化测试工具来捕捉每个测试案例的执行结果，包括成功、失败或异常。

数据记录内容：包括测试案例的描述、预期结果和实际结果，记录发现的任何缺陷或错误，包括错误描述、重现步骤、影响范围及严重程度，记录测试执行的环境设置，如软件版本、硬件配置等。

工具选择：TestRail用于管理测试案例和记录测试结果。Redmine，用于详细记录和跟踪缺陷处理过程。ELK Stack用于收集和分析日志数据。

定期整理测试结果，形成测试报告，供项目团队参考。通过会议、邮件或即时通讯工具及时沟通重要测试发现或问题。

定期审查测试日志，评估其有效性和准确性，根据需要进行优化。

## 缺陷报告

[本节确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。]

缺陷报告工具：对于多模态数据治理平台，我们选择了XYZ缺陷跟踪系统。该系统支持多模态数据输入（如文本、图像、音频），并且能够集成到我们的数据治理平台中。

缺陷记录方法：当系统检测到潜在缺陷时（如数据异常、性能下降），自动创建缺陷记录。缺陷将根据严重性、紧急程度和影响范围进行分类。根据缺陷类型和严重性，自动或手动分配责任给相应团队或个人。

缺陷跟踪和报告：缺陷状态将定期更新，包括“新建”、“进行中”、“待验证”和“已关闭”等，定期生成缺陷报告，包括趋势分析和关键缺陷的深入分析。对已解决的重大缺陷进行复盘分析，以提取经验教训和改进措施。

# 附录 A：项目任务

以下是一些与测试有关的任务：

• 制定测试计划

* + 确定测试需求
  + 评估风险
  + 制定测试策略
  + 确定测试资源
  + 创建时间表
  + 生成测试计划

• 设计测试

- 准备工作量分析文档

- 确定并说明测试用例

- 确定并结构化测试过程

- 复审和评估测试覆盖

• 实施测试

* + 记录或通过编程创建测试脚本
  + 确定设计与实施模型中的测试专用功能
  + 建立外部数据集

• 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

• 评估测试

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准