ALIBABA CLOUD

# 阿里云

## DataWorks

## 研发规范

文档版本：20220524

阿里云

# 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

| 格式 | 说明 | 样例 |
| --- | --- | --- |
| 危险 | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 | 危险<br>重置操作将丢失用户配置数据。 |
| 警告 | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 | 警告<br>重启操作将导致业务中断，恢复业务时间约十分钟。 |
| 注意 | 用于警示信息、补充说明等，是用户必须了解的内容。 | 注意<br>权重设置为0，该服务器不会再接受新请求。 |
| 说明 | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 | 说明<br>您也可以通过按Ctrl+A选中全部文件。 |
| > | 多级菜单递进。 | 单击**设置**> **网络**> **设置网络类型**。 |
| **粗体** | 表示按键、菜单、页面名称等UI元素。 | 在**结果确认**页面，单击**确定**。 |
| Courier字体 | 命令或代码。 | 执行 `cd /d C:/window` 命令，进入Windows系统文件夹。 |
| *斜体* | 表示参数、变量。 | `bae log list --instanceid`<br>*Instance_ID* |
| [] 或者 [a\|b] | 表示可选项，至多选择一个。 | `ipconfig [-all\|-t]` |
| {} 或者 {a\|b} | 表示必选项，至多选择一个。 | `switch {active\|stand}` |

# 目录

# 1.数据仓库研发规范概述

本文将为您介绍数据仓库研发规范的阶段规划、角色职责和整体流程。

在大数据时代，规范地进行数据资产管理已成为推动互联网、大数据、人工智能和实体经济深度融合的必要条件。贴近业务属性、兼顾研发各阶段要点的研发规范，可以切实提高研发效率，保障数据研发工作有条不紊地运作。而不完善的研发流程，会降低研发效率，增加成本与风险。

总而言之，数据资产管理实际上是对物的管理，而研发流程规范管理则是对人的行为的管理。只有落实了作为基础的后者，才能进一步实行数据资产管理方法论。

数据仓库研发规范旨在为广大数据研发者、管理者提供规范化的研发流程指导方法，目的是简化、规范日常工作流程，提高工作效率，减少无效与冗余工作，赋能企业、政府更强大的数据掌控力来应对海量增长的业务数据，从而释放更多人力与财力专注于业务创新。

## 阶段规划

鉴于对日常数据仓库研发工作的总结与归纳，本文将数据仓库研发流程抽象为如下几点：

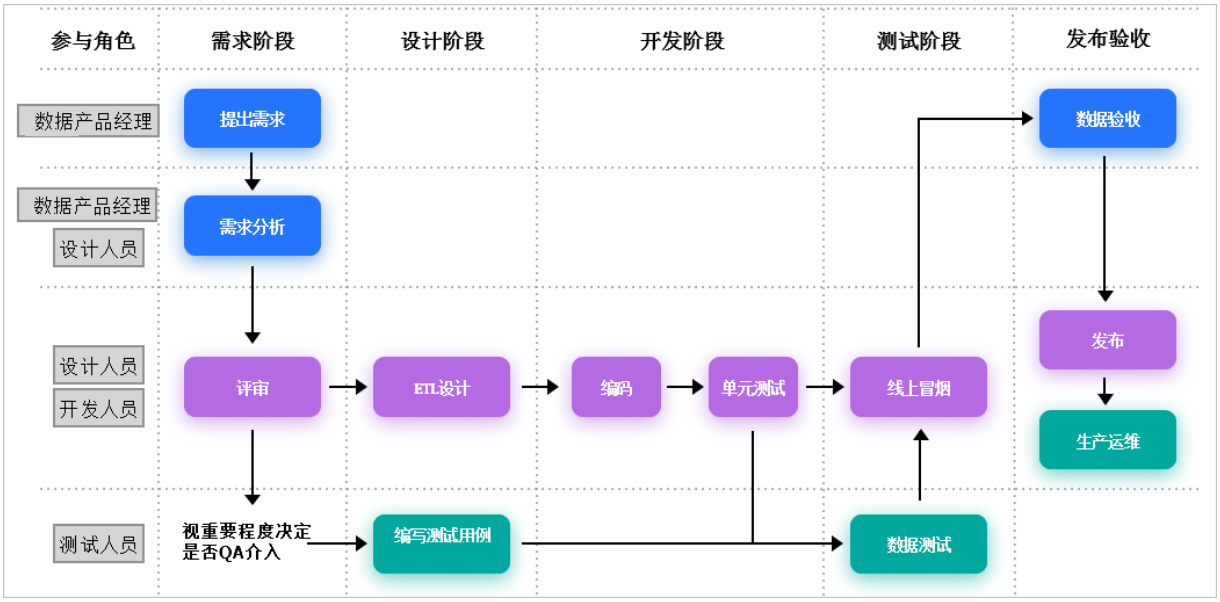
1. 需求阶段：数据产品经理应如何应对不断变化的业务需求。
2. 设计阶段：数据产品经理、数据开发者应如何综合性能、成本、效率、质量等因素，更好地组织与存储数据。
3. 开发阶段：数据研发者如何高效、规范地进行编码工作。
4. 测试阶段：测试人员应如何准确地暴露代码问题与项目风险，提升产出质量。
5. 发布阶段：如何将具备发布条件的程序平稳地发布到线上稳定产出。
6. 运维阶段：运维人员应如何保障数据产出的时效性和稳定性。

## 角色职责

- 数据产品经理：负责承接、评估业务方提出的数据需求，并组织需求评审、产出产品需求文档，同时需要把控其它更为细化的技术评审。
- 设计人员：根据已定稿的产品需求文档所述需求，进行数据探查，了解数据形态（数据质量、数据分布），同时根据探查结果实现表设计、Mapping设计、调度设计等细分设计工作。
- 开发人员：根据设计人员产出的稿件，制定计划并实现代码，同时进行单元测试与代码评审。
- 测试人员：负责验证需求与结果的一致性，发现代码问题与项目风险。
- 运维人员：负责发布任务，并处理数据、程序、调度、监控告警等异常事件，保障数据产出时效、程序高效运行和生产稳定性。
- 信息安全与合规人员：在需求评审前期，负责需求实现的安全性与合规性。

## 数据仓库研发规范整体流程

下图为根据阶段规划与角色职责的内容，整理出的数据仓库研发规范的整体流程。



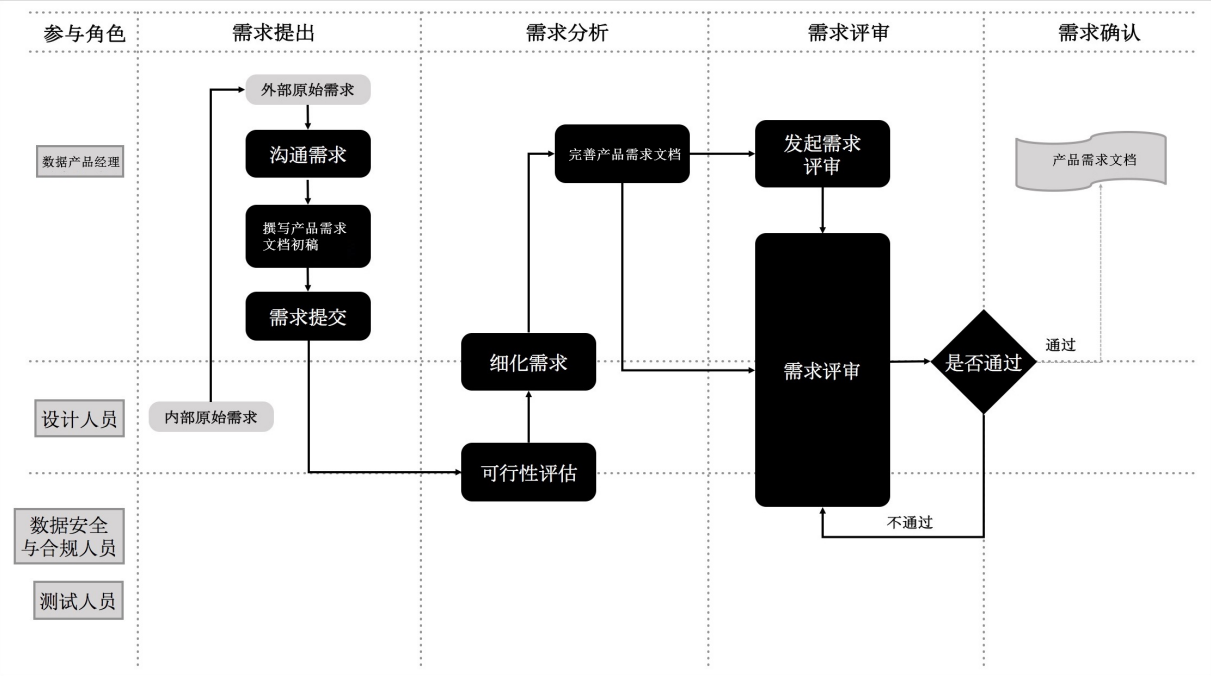
# 2.需求阶段

数仓的最基本职责是定义和发现在企业决策中使用的信息，随着企业战略方向的改变与业务方对行业判断的变化，需求会不断变化。该特性决定了数据仓库需求的多样性和迭代性。

作为承接业务方数据需求的数据产品经理，在需求阶段需要规范首次需求流程和迭代需求流程。

## 首次需求流程

对于业务方首次提出的需求，重点工作在于评估完成该需求的技术、数据、合规的可行性后，以细化需求的方式完成产品需求文档，并组织需求评审会议多方共同敲定需求最终实现方案。




首次需求流程包括以下步骤：

1. 提出需求
   - 外部沟通：数据产品经理主导，负责与外部门业务方充分沟通。力求获取并理解业务场景（背景）、目标和实现价值。

     > **说明** 此处不必与业务方讨论需求实现的途径或细节，双方只了解需要达到什么目标，而不讨论如何实现。

   - 完成产品需求文档的初稿：得到充分信息后，按照数据仓库需求模板中的常规需求申请单，将需求转化为产品需求文档的初稿。
2. 分析需求
   - 可行性分析：数据产品经理主导，邀请设计、数据安全与合规人员，对需求进行评估。
     - 需求合理性：评估该需求的合理性。
     - 数据可行性：评估当前已有数据能否支撑需求开发，如果缺少数据，则需要另行规划缺失数据的抽取方案。

       同时建议进行深入的数据探查，包括但不限于数据完整性、字段离散值分布情况、空值、零值、重复值占比等情况。

- 技术可行性：评估当前已有数据模型能否支撑需求开发，如果不能，则需要规划模型改造方案，并充分评估其影响。同时在测试环境进行模型测试。

  > **说明** 如果涉及资损、精确对账或其他关键模型的改造，测试人员必须进行测试。

- 是否满足安全与合规要求：根据企业自身数据安全的要求，严格控制数据内部流向，划分研发过程中数据可流入的库、项目、表、字段等。对于流出外部的数据，更需要严格评估流出数据内容、流出目的地是否符合公司数据安全的要求。

  > **说明** 此项评估是不可跳过的步骤。

  - 实现细节分析：数据产品经理主导，对实现需求的细节关键点进行确认，包括但不限于数据口径、接口格式、供数频率和需求优先级。
  - 完善产品需求文档：完善产品需求文档的初稿。

3. 评审需求

   数据产品经理主导，邀请设计人员、测试人员发起需求评审会。会议内容主要包括：

   - 各方提出对于产品需求文档中各细节的疑问。
   - 共同达成对于疑问的解决方案。

   > **说明** 评审会议上不得遗留影响后续研发流程的关键问题，否则视为评审不通过。

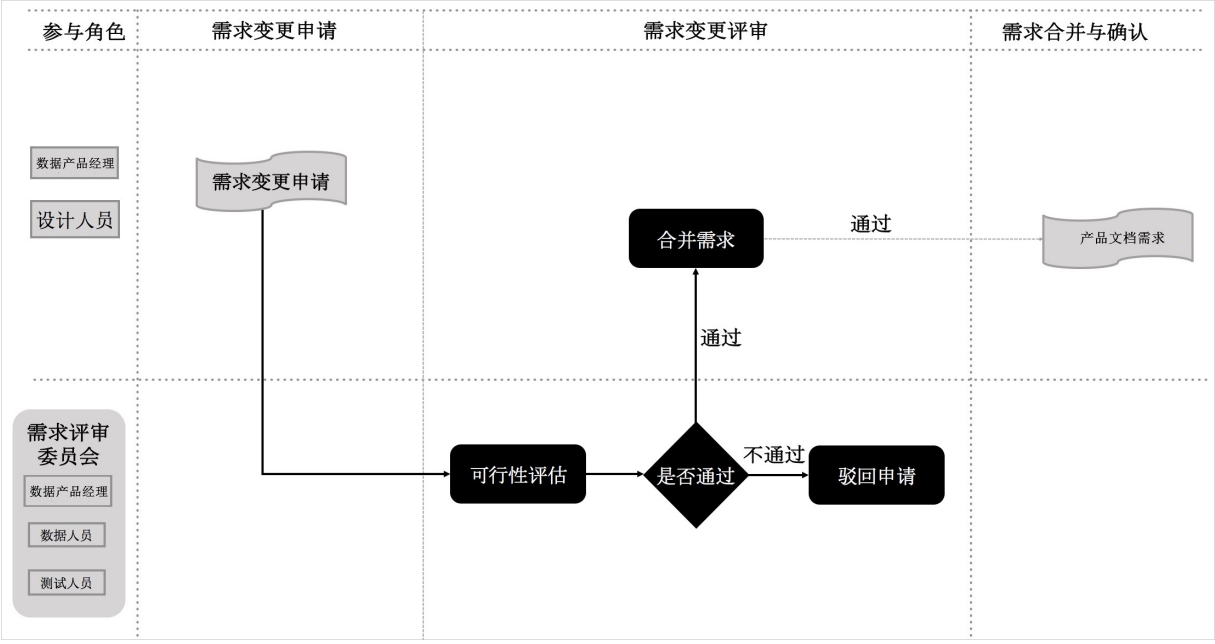
4. 确认需求

   N个工作日（视各企业实际情况而定）内如果无异议，则产品需求文档定稿，并开始进入后续的设计与开发阶段。

## 迭代需求流程

对于同一需求，在完成首次需求评审并定稿产品需求文档后，业务方再次提出的需求，均属于迭代需求。

迭代需求的流程与首次需求流程类似，均需进行可行性分析、实现细节分析。分析完成后，视实际情况来定是否需要再次进行需求评审，最终将新老需求合并至产品需求文档终稿。




迭代需求流程包括以下步骤：

1. 申请需求变更

   数据产品经理完成业务方迭代需求对接后，将新的需求录入数据仓库需求模板的迭代需求申请单中。

   > **说明** 如果企业具备需求相关管理平台，建议通过平台+数据库形式规范化存储不断迭代的每个需求版本。

2. 评审需求变更

   原则上需求评审需由数据产品经理发起评审会议来完成，但如果需求迭代内容不多，评审方式可视情况而定选择邮件或现场会议方式，具体视变更内容由变更委员会决定。

   评审内容仍为实现需求必须面对的技术可行性、数据可行性、安全与合规要求性展开讨论，如果多方有异议，则必须共同达成一致性解决方案。

3. 确认并合并需求

   数据产品经理将上一版本定稿的产品需求文档内容，与本次评审定稿的产品需求文档内容进行合并。

   如果两个工作日内无异议，则视为需求确认。

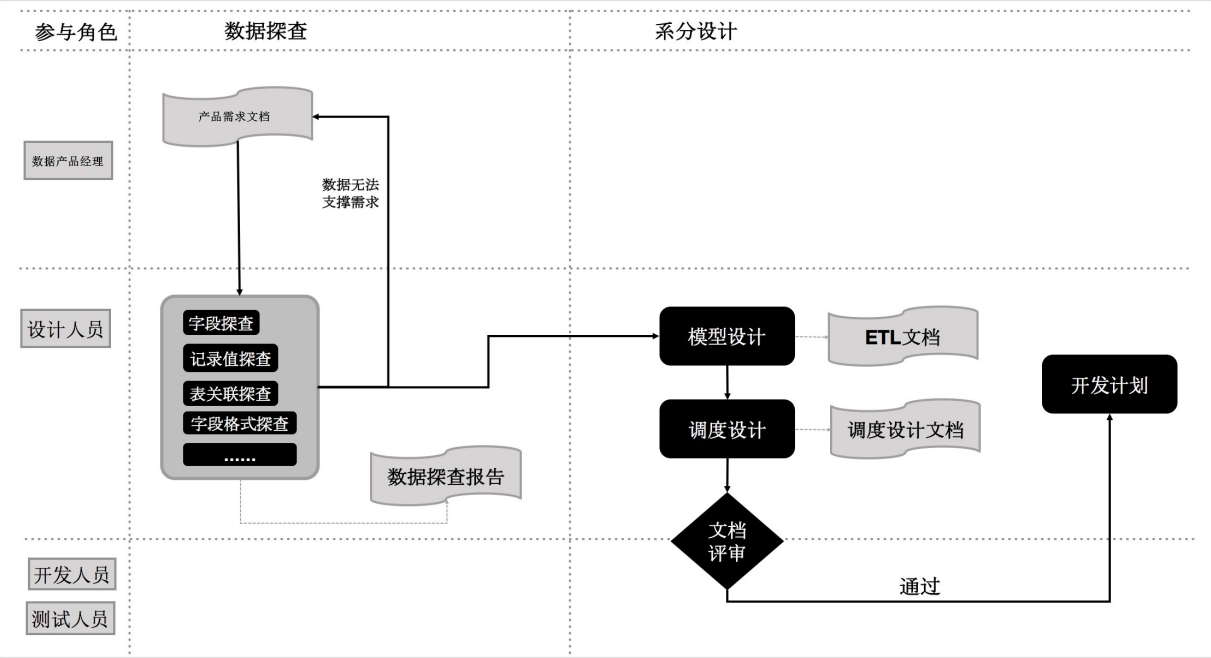
# 3.设计阶段

完成需求阶段的工作后，数据产品经理会产出最终版本的产品需求文档，以供设计人员进行设计工作。

设计工作包含数据探查和系分设计两部分：

- 数据探查旨在了解来源数据的数据形态，例如数据质量、数据分布等。结合业务场景，帮助分析和判断需求实现的可行性以及找出潜在的数据问题和风险。
- 系分设计则包括表设计、Mapping设计和调度设计等最实际的设计工作。

设计完毕后，最终将产出供开发人员参照实施开发的ETL设计文档、数据探查文档、调度设计文档，为需求的有效实现打下坚实基础。




设计阶段的流程包括以下步骤：

1. 数据探查

   数据探查的目的是了解数据的形态，找到潜在问题与风险。数据探查是决定数据可靠性的关键步骤。数据探查报告可以为后续开发提供指导，并作为依据指定开发计划。

   数据探查的内容主要包括但不限于以下内容：

   - 源表数据主键字段重复数。
   - 源表字段空值/异常值的统计数。
   - 源表之间关联关系。
   - 源表字段的数据格式。
   - 源表增量规则。

   探查完成后，最终产出数据探查报告。如果发现当前数据无法支撑需求的实现，则要将需求退回给数据产品经理，由数据产品经理发起迭代需求流程。

2. 系分设计

   系分设计包括表设计、Mapping设计和调度设计三部分。

   - 表设计

表设计是指依据需求设计目标产出表、中间产出表。包含表名、表名解释、字段名、字段类型、字段注释以及字段安全等级等。表设计的步骤如下所示：

a. 设计表名、字段名：要求相同的字段在不同表中的字段名相同，相关规范请参见命名规范。

b. 设计主键和外键。

c. 设计字段注释：通过标注字段注释、枚举值来表明字段含义，如果枚举值过多，建议为枚举值创建维表。

d. 设计表分区：建议所有表都创建为分区表。

e. 设计数据生命周期。

企业应根据自身实际情况来进行设置，也可以参考如下数值：

| 数仓分层 | 说明 |
| --- | --- |
| ODS层 | ■ 非去重数据：默认不保留。<br>■ ETL临时表：保留14日。<br>■ 镜像全量表：重要数据建议采用极限存储。<br>■ 流水全量表：如果不可再生，则永久保存。 |
| DWD层 | ■ 维度表：按日分区的极限存储模式。<br>■ 事实表：按日分区且永久保留。<br>■ 周期性快照事实表：采用极限存储或根据自身情况设置生命周期。 |
| DWS层 | 汇总指标：自行选择保留月初、特定日期数据。 |

f. 设计加密技术：根据实际情况对敏感字段设计加密方案。

- Mapping设计

Mapping设计采用图形化或伪代码的形式编写规划以下内容：

- 每个字段的生成逻辑。
- 表与表之间的关系。
- 目标字段与原字段间的算法逻辑。

将上述内容产出为ETL文档留存，ETL将作为后续开发流程的第一参考依据。

3. 调度设计

i. 依赖设计

将ETL抽象为多个相互依赖的代码节点形成上下游依赖关系，要求如下：

- 一个节点仅产出一张表，一张表仅由一个节点产出。
- 下游节点的输入数据来自于上游节点的产出数据。
- 多并行、少串行（在分布式系统下可发挥其优势）。

ii. 运行周期

如果数据研发的场景是在常见T+1离线计算场景，则应将不同调度任务按照实际业务需求，赋予小时、日、周、月和季度等不同的调度粒度。

> 说明
>
> - 程序必须支持重跑。
> - 如果SQL语句优化后，单次执行仍超过30分钟，建议拆表重新设计，建议每个节点运行时长不超过1小时。

iii. 设置基线：在传统T+1（每日计算的是前一日产生的业务数据）的场景下，数据理应在第二天某个时间点按时产出以支撑BI或其他应用场景，因此应设置如下基线报警策略。详情请参见基线管理。

- 最终产出任务基线：规定产出最终数据的任务必须在公司规定的X点X分完成，否则视为破线（同时推送相应报警）。
- 中间任务报警：产出最终数据的任务的上游任务应稳定、按时运行完成。如果出现出错、变慢（运行时间明显长于历史过往平均运行时间）等可能影响最终任务完成时间的事件，则应第一时间推送报警给第一任务责任人。

iv. 设置优先级：基于有限的计算资源来设置任务优先级，以保证在已有资源被充分调配利用的情况下，可以按照顺序产出数据，保证重要任务的准时产出。调度设计完成后，需要产出调度设计文档。

v. 数据流设计

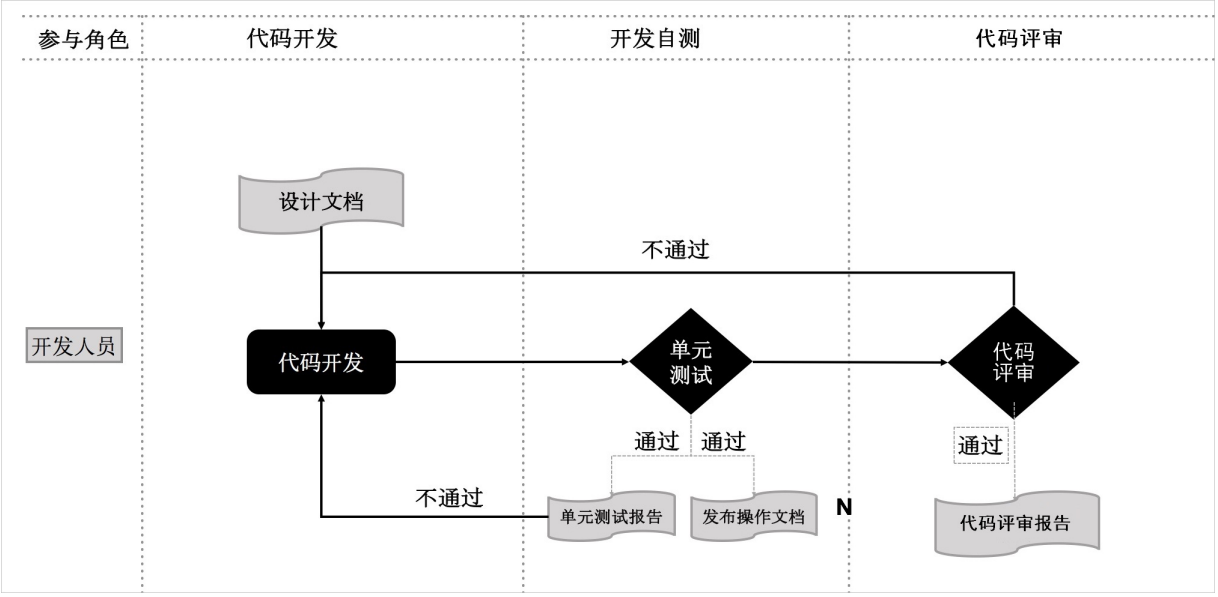
ETL过程中，数据流向有如下限制：

- 数据流向仅支持由低到高，即ODS->DWD->DWS->ADS。
- 数据不能且不能跨层引用、逆向引用。
- DWS层不同集市的数据不能相互引用，必须沉淀到DWD层。

# 4.开发阶段

您在完成需求评审、模型与调度设计后，即可进入数据开发阶段。

开发阶段的主要任务是将设计阶段的产出转化为具体代码。开发过程中，开发人员必须保证代码的规范性、准确性。同时进行适当的单元测试，以便后续测试工作可以顺利开展。




开发阶段的流程包括以下步骤：

1. 代码开发

   该部分内容请参见编码规范，编码时需要注意以下问题：

   - 层次分明、结构化强。
   - 增加必要注释，以增强代码的可读性。
   - 充分考虑执行速度最优的原则。
   - 四个空格为一个缩进量，所有缩进皆为一个缩进量的整数倍，按照代码层次对齐。
   - 不建议使用 `select *` 操作，所有操作必须明确指定列名。
   - 所有产出表都需要有物理主键或逻辑主键，并纳入周期性数据质量监控。

2. 单元测试

   代码开发完成后，开发人员需要对代码进行单元测试，单元测试阶段包括以下内容：

   - 规范性检查。
   - 代码质量检查：建议单条SQL执行时间不超过30分钟。
   - 数仓特殊需求检查。
   - 指标特性检查。

   单元测试完成后，需整理输出单元测试报告和发布操作文档，以便开展后续发布工作，详情请参见单元测试报告和发布操作文档。

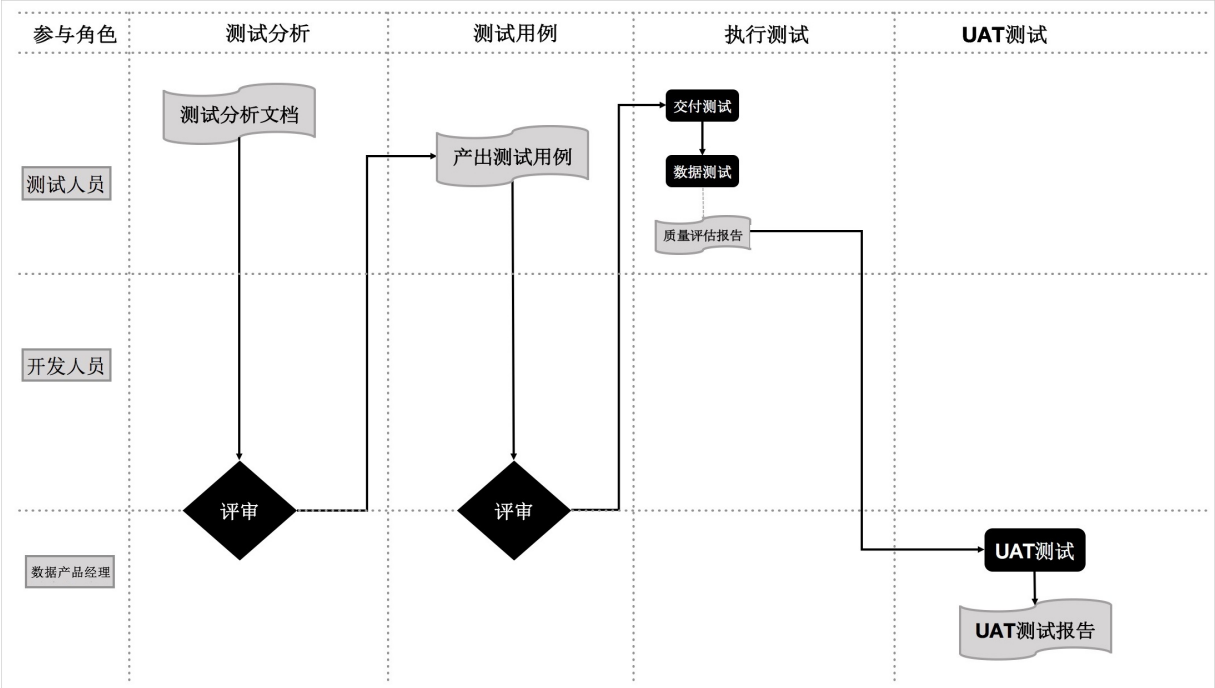
3. 代码评审（Code Review）

   单元测试完成后，需要由其它开发人员进行代码评审，最后查看代码评审报告，详情请参见代码评审报告。

代码评审包括数据一致性检查、数据完整性检查和指标间逻辑检查。

# 5.测试阶段

开发阶段已经完成了代码的实现，为了发现代码问题、暴露项目风险、提升产出质量，需要进入测试阶段，通过测试用例对代码进行分析，为最终发布提供决策的依据。




测试阶段的流程包括以下步骤：

1. 测试分析

   根据需求阶段、设计阶段的要求，结合来源数据的探查来明确整个测试流程的目标、方案、风险与难点：

   - 测试范围
   - 测试策略和方法
   - 具体交付物、退出标准
   - 预期风险
   - 测试环境、测试数据的准备

   此外，测试分析应经过企业内部评审或项目组评审，以保证测试的科学性。

   测试分析完成后，需输出测试方案分析报告，详情请参见测试分析方案报告。

2. 准备测试用例

   测试方案明确后，需要编写测试用例、测试代码和准备数据。

   测试用例编写需遵循结构有序、条理清晰、他人可执行的原则，同时各团队需有效维护和保存，以便日后进行复用、故障问题回溯。建议测试用例编写完成后组织公司内部评审。

3. 执行测试

i. 交付测试：为了将问题在前期设计、研发和自测环节完成收敛，需进行交付测试，以便保障流入到测试执行环节的代码达到一定的质量标准。

交付测试的标准包括编码是否符合规范、是否完成代码评审、是否提供数据探查报告、交付缺陷的严重程度和用例占比、选用测试用例集的执行通过率。

测试完成后输出交付测试报告，详情请参见交付测试报告。

ii. 数据测试

测试期间需重点关注以下事项：

- 代码规范性：命名规范、编码类型是否符合要求。
- 数据规范性：命名规范、表结构规范、精度要求、空值处理方式、时间类型格式等是否符合要求。
- 数据基础：主键唯一性，空值、重复值、无效值占比是否符合要求。
- 业务正确性：各业务点是否被正确实现，可以通过划分边界值、等价类等样本数据进行验证。
- 代码性能：验证代码是否可在业务要求产出的时间成功运行完成。

测试期间，需要严格按照事前制定的测试策略和测试用例执行测试，建议将测试过程中的测试点修改补充到测试用例中，为今后线上问题进行回溯和排查提供参照和依据。

iii. 测试报告：测试完成后需发布质量评估报告，报告中需表现当前项目缺陷修复情况、遗留问题排期评估、发布后的预期风险，以及最终关于发布或延期的结论。

测试报告请参见质量评估报告模板。

4. UAT测试：交付测试、数据测试完成后，数据产品经理需要站在在业务角度，对产出数据进行验收测试，最终提供验收测试报告。

UAT测试报告请参见验收测试报告模板。

# 6.发布阶段

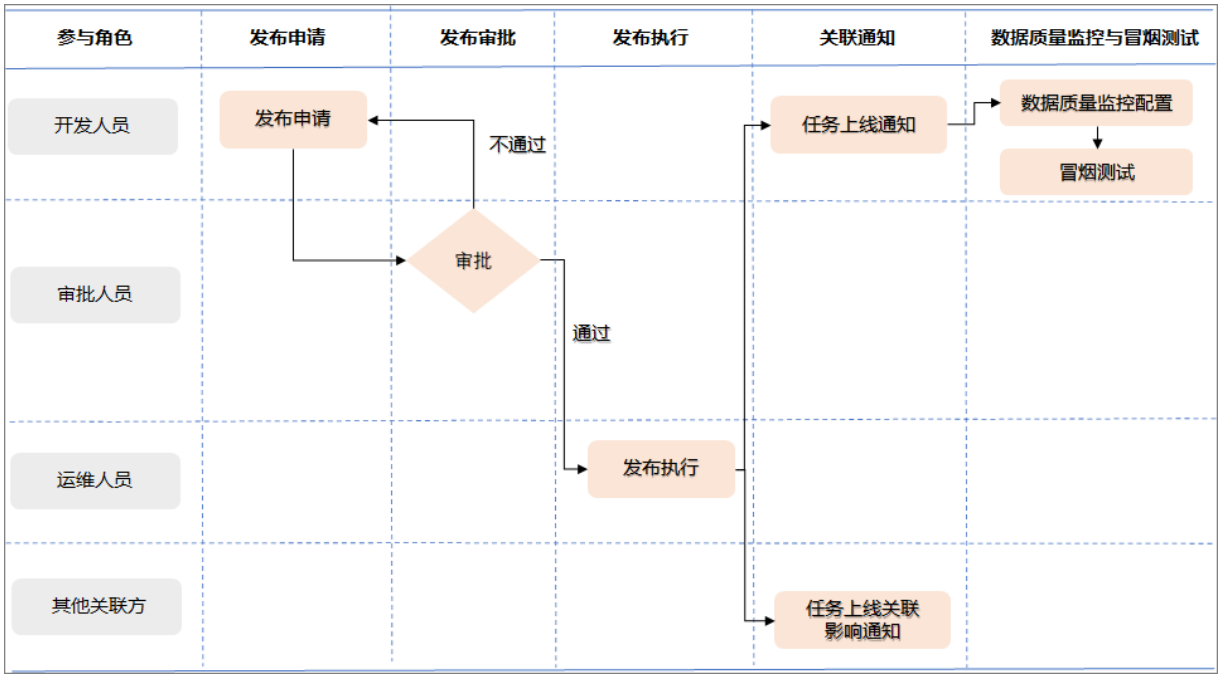
发布是将具备发布条件的程序发布到线上系统，并以生产标准进行数据产出的过程。

发布分为正常发布和紧急发布：

- 正常发布：发布节奏在原则上是可预见性、周期性的，发布计划可提前制定和公布。正常列入排期计划的需求，都必须按照正常的节奏安排发布计划。
- 紧急发布：紧急发布是为应对突发性、紧急性状况而额外开启的可选发布，如线上BUG紧急修复、突发性需求等。

  在接到紧急发布需求后，第一时间应评估是否可以随最近一次正常发布窗口期发布。如果不可以，则根据企业实际情况发起紧急发布申请。

发布阶段的流程主要包括发布申请、发布审批和发布执行。




1. 发布申请：发布申请是发布工作的进入环节，该环节主要包括程序源代码、质量评估报告、UAT验收报告和发布版本。
2. 发布审批：审批环节是对发布申请合法性的赋权和放行环节。在该环节，需要对发布申请的合规性、规范性和合理性进行审核，具体审批目的包括但不限于以下几点：
   - 发布内容是否与原始需求一致。
   - 发布内容是否与数据安全、合规要求有冲突。
   - 发布内容是否会造成任务报错、脏数据写入等情况。
   - 发布内容的发布时间段是否合理或需要调整。
   - 紧急发布的必要性。

   建议安排对业务逻辑、代码较为熟悉的人员把控审批流程。审批通过后即进入发布执行阶段。如果不通过，则发布立即终止，或驳回申请进行调整后重新申请。

   审批环节是一个非常重要且不可或缺的环节，它关系到数据生产环境的稳定性和数据的可靠性、安全性。建议企业根据自身情况，安排经验丰富的相关人士来承担此项工作。
3. 发布执行：审批通过后，由运维人员执行发布。

为保证将程序正确、完整地发布到线上，发布时应严格按照开发人员的发布操作步骤执行，且可以查询操作日志记录。

发布完成后，发布人员需要启动关联通知工作。

4. 关联通知：发布人员需将发布变更信息及时通知包括但不限于以下关联方：
   - 该代码所在节点的一级子节点责任人。
   - 任务关联产出基线责任人。
5. 数据质量监控与冒烟测试：发布完成后，开发人员根据数据与业务特点配置数据质量监控规则，并进行冒烟测试。

   冒烟测试必须完成至少一个调度周期的运行，以验证新发布或者变更的任务节点可行性。如果冒烟测试不通过，则发布执行人员需根据情况，执行代码回滚或者通知开发人员进行紧急线上发布。

# 7.附录

## 7.1. 数据仓库需求模板

本文将为您介绍数据仓库需求模板、常规需求申请单和迭代需求申请单。

填写说明：

- *为必填项目，其它可以选择性进行填写。
- 指标逻辑可以引用指标和术语（或指标库）中的定义。
- 如果数据范围、更新频率、时间窗口、数据提供形式和表头信息不一致，可以针对指标项单独说明。
- 如果涉及到数据提供或数据交互，数据验收人、待验收数据样本和数据验收方式为必填项，其它项并非强制需求。

### 数据仓库业务需求模板

| 数据仓库业务需求模板 | | | |
|---|---|---|---|
| 需求申请 | 需求申请人* | | |
| | 需求使用方* | | |
| | 期望完成日期* | | |
| | 需求类型* | | |
| 需求目的 | 需求背景* | | |
| | 期望目标* | | |
| | 应用系统名 | | |
| | 应用系统联系人 | | |
| 需求内容 | 需求概览 | 需求范围* | 描述此次需求涉及的范围（可以从人群特征，业务场景等维度定义数据范围、改造哪些表等）。 |
| | | 包含的指标 | 多个指标以逗号分隔。如果指标较多，可以在日常业务需求附表中的指标名称一栏填写。 |
| | | 数据交互方式 | 涉及到数据输出的，需要描述数据的交互方式、格式等。 |
| | | 附件说明 | 如果有附件需要补充的，请在此说明，并同步附加附件。 |

| 数据仓库业务需求模板 | | | |
|---|---|---|---|
| 项目涉众 | 数据产品经理 | | |
| | 设计人员 | | |
| | 开发人员 | | |
| | 测试人员 | | |
| | 数据安全与合规人员 | | |
| 需求版本变更历史 | | | |
| 版本号 | 版本确认日期 | 版本变更点 | 提交人 |
| | | | |

## 常规需求申请单

指标需求中通常会涉及到下表中的约定项，如果需要自定义约定项，可以在自定义格式列进行填写。

| 约定项 | 默认格式 | 自定义格式 |
|---|---|---|
| 日期 | yyyymmdd | |
| 比率值 | 4位小数点 | |
| 时间戳 | yyyy-mm-dd hh24:mi:ss，格林尼治时间。 | |
| 金额 | 单位为分。 | |
| 时间粒度 | 日：T-1日的00:00~24:00。 | |
| | 周：周一到周日，对应指标仅周日有值。 | |
| | 月：自然月，对应指标仅月末最后一天有值。 | |
| | 年累计：自然年，1月1日到T-1。 | |
| | 财年累计：财年4月1日到T-1。 | |

| 约定项 | 填写内容 | 约定项 | 填写内容 |
|---|---|---|---|
| 时间窗口（历史数据要求）* | | 存储周期* | |
| 更新频率（日、周、月、小时、分钟、其它）* | | 期望数据更新时间* | |
| 数据验收人 | | 待验收数据样本 | |

| 约定项 | 填写内容 | 约定项 | 填写内容 |
| --- | --- | --- | --- |
| 数据验收方式 | | 数据提供形式 | • 物理表<br>• 数据文件<br>• 数据查询服务或接口 |
| 备注 | | | |

| NO. | 粒度 | 目录 | 接口表 | 指标名称* | 指标逻辑* | 空值/异常值处理* | 监控项 | 值是否唯一* | 数据来源* | 安全等级* | 备注 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

### 迭代需求申请单

| 数据仓库需求变更申请单 | | | |
| --- | --- | --- | --- |
| 需求变更申请 | 原始需求ID* | | |
| | 需求申请人* | | |
| | 需求使用方* | | |
| | 期望完成日期* | | |
| 需求变更原因 | 需求变更背景* | | |
| | 是否可以在需求评审前预知* | | |
| | 如何避免此类变更发生* | | |
| 需求变更内容 | 原始需求（对于新增的需求，填无）* | 变更内容* | 变更类型* |
| | | | |

# 7.2. 数据探查报告

数据探查报告模板，如下表所示。

| 字段顺序 | 字段名 | 字段注释 | 字段类型 | 总行数 | 空值个数 |
| --- | --- | --- | --- | --- | --- |
| | | | | | |
| | | | | | |
| | | | | | |

| 空值比例 | 唯一个数 | 均值（number）：：TOP1（string） | 最小值：：TOP2 | 1%分位数：：TOP3 | 5%分位数：：TOP4 |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |

| 25%分位数：：TOP5 | 中位数：：BOT5 | 75%分位数：：BOT4 | 95%分位数：：BOT3 | 99%分位数：：BOT2 | 最大值：：BOT1 |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |

# 7.3. ETL文档

## 表总览

| 表名 | 说明 |
|---|---|
| ods_raw_log_d | 离源ODS层最近的数据 |
| dwd_user_info_d | 用户公共明细表 |
| dws_user_info_d | 用户公共汇总表 |
| dm_user_info_d | 用户数据集市表 |
| rpt_user_info_d | 用户分析汇总表 |

## 节点dwd_user_info_d

| 任务（节点）名称 dwd_user_info_d | | | | | | |
|---|---|---|---|---|---|---|
| 字段名称 | 目标表字段 | 字段说明 | 源表 | 涉及源表字段 | 算法说明 | 备注 |
| uid | 用户ID | 用户ID | ods_log_info_d | uid | 抽取汇总 | |
| gender | 性别 | 性别 | ods_log_info_d | time | 抽取 | |
| age_range | 年龄段 | 年龄段 | ods_log_info_d | status | 抽取 | |
| zodiac | 星座 | 星座 | ods_log_info_d | bytes | 抽取 | |

| 任务（节点）名称 dwd_user_info_d | | | | | | |
|---|---|---|---|---|---|---|
| 字段名称 | 目标表字段 | 字段说明 | 源表 | 涉及源表字段 | 算法说明 | 备注 |
| region | 地域，根据IP获取 | 地域，根据IP | ods_log_info_d | region | 转换 | |
| device | 终端类型 | 终端类型 | ods_log_info_d | method | 抽取 | |
| identity | 访问类型 crawler feed user unknown | 访问类型 crawler feed user unknown | ods_log_info_d | URL | 抽取 | |
| method | HTTP请求类型 | HTTP请求类型 | ods_log_info_d | protocol | 抽取 | |
| URL | URL | URL | ods_log_info_d | referer | 截取引用IP | |
| referer | 来源URL | 来源URL | ods_log_info_d | device | 截取设备名称 | |
| time | 时间 yyyymmddhh:mi:ss | 时间 yyyymmddhh:mi:ss | ods_log_info_d | identity | | |

# 7.4. 调度设计文档

| 节点ID | 节点名称 | 用途 | 数据输入表 | 数据产出表 | 调度周期 |
|---|---|---|---|---|---|
| 320170257 | workshop_start | 虚拟节点，用于管理下游节点 | Null | Null | 日 |
| 320170260 | MySQL数据同步 | 拉取MySQL数据源数据 | ods_user_info_d | ods_user_info_d | 日 |
| 320170260 | FTP数据同步 | 拉取FTP数据源数据 | Null | ods_raw_log_d | 日 |
| 320170261 | ods_log_info_d | 原始数据脏数据清理 | ods_raw_log_d | ods_log_info_d320170259 | 日 |
| 320170262 | dw_user_info_all_d | 轻度汇总数据 | ods_log_info_d | dw_user_info_all_d | 日 |
| 320170263 | rpt_user_info_d | 统计汇总报表数据 | dw_user_info_all_d | rpt_user_info_d | 日 |

| 定时时间 | 预计运行时间 | 上游节点ID | 上游节点名称 | 基线时间 | 优先级 |
| --- | --- | --- | --- | --- | --- |
| 00：01 | 5s | Null | Null | Null | 1 |
| 00：03 | 1mins | 320170257 | workshop_start | Null | 1 |
| 00：03 | 1mins | 320170257 | workshop_start | Null | 1 |
| 00：05 | 10mins | • 320170260<br>• 320170259 | • MySQL数据同步<br>• OSS数据同步 | Null | 1 |
| 00：20 | 5mins | 320170261 | ods_log_info_d | Null | 1 |
| 00：30 | 30s | 320170262 | dw_user_info_all_d | 0:40:00 | 1 |

# 7.5. 单元测试报告

## 单元测试要求

| 用例小类 | 测试要点 | 说明 | 是否已检查（Y/N） |
| --- | --- | --- | --- |
| 规范性 | 命名规范检查（表、视图、工作流、字段） | 是否符合MaxCompute数仓建设规范管理指南中命名规范的表命名规范。 | |
| | 代码格式和注释规范性 | 是否符合MaxCompute数仓建设规范管理指南中的编码规范。 | |
| | 表引用规范性 | 数据不允许跨层引用。 | |
| | 表更新策略规范 | 建议临时表均为非分区表，正式表均为分区表。 | |
| | 是否支持重跑 | 代码必须支持重跑。 | |
| 源数据质量 | 非空值检查 | 检查所用字段是否存在空值，以及代码对空值处理的策略是否正确。 | |
| | 字段枚举值检查 | 字段的枚举值是否都在代码考虑范围内，是否有可能会出现新值。 | |
| | 主键检查 | 物理主键或逻辑主键是否成立。 | |
| | 数据完整性检查 | 代码中引用的数据能否支撑实际需求。 | |

| 用例小类 | 测试要点 | 说明 | 是否已检查（Y/N） |
| --- | --- | --- | --- |
| | 字段间逻辑检查 | 字段间的业务逻辑关系是否在数据上成立，例如余额=总的发放-总的回收。 | |
| 代码质量/BUG检查 | 历史拉链表检查断链/交叉链 | 使用标准SQL进行检验。 | |
| | 数据倾斜检查 | 是否存在倾斜的情况，是否有大表join小表未用mapjoin等。 | |
| | 表分区选择检查 | 代码对表分区的选择是否正确。 | |
| | 关联条件检查 | 关联条件是否正确，是否会产生意料外的结果，例如多对多关联、笛卡尔积。 | |
| | 字段类型检查 | 字段类型是否正确，例如：金额字段必须为X数据类型，编号字段必须为X数据类型。 | |
| | 执行效率检查 | 单条SQL执行时间不超过30分钟，单个脚本执行时间不超过60分钟。 | |
| 数仓特殊需求 | 脏数据检查 | 检查是否有脏数据。 | |
| | 增量/全量数据抽取规范 | 抽取时间大于X分钟的，则考虑更改为增量抽取。 | |
| | 数仓抽取时间点检查 | 数仓抽取时业务系统是否ready，抽取的数据是否完整。 | |
| 指标特性检查 | 细分指标趋势检查 | 例如会员拉链表记录数相比前一天必须是正增长、当日累计值-上日累计值必须大于0。 | |
| | 不同粒度数据转换正确性 | 例如细粒度向粗粒度汇总，通常使用最大/最高/最小/最低等过滤条件，如：支用层逾期天数转换到客户层指标（最高逾期天数）。最高逾期天数 = Max（支用层逾期天数）。 | |
| | 值域范围检查 | 检查字段值的范围是否正确，如：金额>=0，比率<=1，天数<=业务起始日期至今，还款日期>=放款日期。 | |
| | 代码值分布检查 | 从业务逻辑考量字段值的分布情况是否合理。 | |
| | | | |

| 用例小类 | 测试要点 | 说明 | 是否已检查（Y/N） |
| --- | --- | --- | --- |
| | 可累加值与不可累加值检查 | 检查可累加值和不可累加值的处理逻辑正确性，如：计算客户数总计时需要做去重处理，金额则可以累加。 | |

## 单元测试用例记录

| 序号 | 用例大类 | 测试要点 | 表 | 字段 | 自定义表达式 | 备注 |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 规范性 | 命名规范检查（表、视图、工作流、字段） | jrcdm_agt_ovd_ins_detail_fact_dd | | | |
| 2 | 规范性 | 是否支持重跑 | jrcdm_agt_ovd_ins_detail_fact_dd | | | |
| 3 | 源数据质量 | 主键检查 | afclms_clms_loan_contract | contract_no | | |
| 4 | 指标特性检查 | 值域范围检查 | jrcdm_cust_drawndn_fact_ds | prin_max_ovd_days，inte_max_ovd_days | prin_max_ovd_days>=inte_max_ovd_days | 检验逾期天数的业务逻辑。 |
| 5 | 指标特性检查 | 值域范围检查 | x_jredw_da_drawndn_ovd_date_info | Prin_Ovd_Start_Dt | Prin_Ovd_Start_Dt<=Prin_Ovd_End_Dt，Inte_Ovd_Start_Dt<=Inte_Ovd_End_Dt | 检查业务逻辑正确性。 |

| 测试结果 | 测试结果备注 | 是否转化监控 | 监控阈值 | 创建日期 | 创建人 | 所属项目名称 |
| --- | --- | --- | --- | --- | --- | --- |
| 通过 | | | | 2013/7/16 | XXX | 某项目 |
| 通过 | | | | 2013/7/16 | XXX | 某项目 |
| 通过 | | | | 2013/7/16 | XXX | 某项目 |
| 通过 | | 是 | <1 | 2013/7/16 | XXX | 某项目 |

| 测试结果 | 测试结果备注 | 是否转化监控 | 监控阈值 | 创建日期 | 创建人 | 所属项目名称 |
| --- | --- | --- | --- | --- | --- | --- |
| 未通过 | 开发代码中存在以下两个问题：<br>• 未对期次还款日大于当前日期的记录进行过滤，这部分为未到期记录，需要排除。<br>• 未对记录中创建时间小于期次还款日的、未结清的期次记录的逾期结束时间，赋予与逾期开始时间一致的处理。 | 是 | <1 | 2013/7/16 | XXX | 某项目 |

# 7.6. 发布操作文档

| 序号 | 节点ID | 文件名 | 发布次序 | 是否需要生产冒烟 | 是否需要重跑历史数据 | 重跑历史时间段 | 发布验证是否通过 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | xxxxx | dw_user_log_info_d.sql | 1 | Y | Y | 20190326-20190426 | Y |

# 7.7. 代码评审报告

## 代码评审要求

| 用例小类 | 测试要点 | 说明 | 是否已检查 |
| --- | --- | --- | --- |
| 数据一致性测试 | 主键唯一性 | 产出表必须有物理主键或逻辑主键，且在数据上主键成立。 | 是 |
| | 主键和外键逻辑关系 | 检查设计文档里关于主外键的设计是否在开发阶段得以实现，且在数据上成立，例如是否存在外键丢失。 | 是 |
| | 系统/业务间格式和类型一致性检查 | 检查设计文档描述的字段定义是否与实际值一致。例如日期是否包含时分秒，金额字段是否为Double，单位为元/分，保留小数位数。 | 是 |
| | 业务来源一致性检查 | 从同样业务来源的指标是否在数据上一致。例如同样是余额指标，数据来源是否一致或来自同一加工链路，如果不是，则结果是否一致。 | 是 |

| 用例小类 | 测试要点 | 说明 | 是否已检查 |
| --- | --- | --- | --- |
| | 同名逻辑定义检查 | 字段或逻辑定义相同，是否存在值不一样的情况。例如同样是贷款发放额，不同的表之间数据是否一致。 | 是 |
| 数据完整性 | 数据获取是否完整 | 代码中的数据获取逻辑是否完整。例如累计客户数，是否完整包含了历史上有效存在，但当前不存在的客户。 | 是 |
| | 边界值检查 | 代码中对于边界值的处理是否正确。例如最近30天包含今天但不包含第前30天的。例如日期筛选是否为双闭区间。 | 是 |
| | 过滤条件完整性 | 过滤条件是否完整。例如筛选当前有效会员需要加上会员状态的限制。 | 是 |
| 指标间逻辑检查 | 同表字段间逻辑检查 | 同表不同字段间在业务上存在的逻辑是否在数据上成立。例如贷款为结清状态，则结清日期一定非空；状态为逾期，则逾期金额一定大于0。 | 是 |
| | 跨表/跨系统逻辑检查 | 跨表/跨系统间在业务上存在的逻辑是否在数据上成立。例如不良贷款余额>0，则该账户三级分类应为次级、可疑和损失。 | 是 |

### 代码评审测试用例记录

| 备注 | 测试结果 | 测试结果备注 | 是否转化监控 | 监控阈值 | 创建日期 | 创建人 | 所属项目名称 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 检查主键的唯一性 | 通过 | | 是 | <1 | 2019/3/16 | XXX | 订单主题分析 |

# 7.8. 测试分析方案报告

## 产品概述

- 产品背景

  描述该数据产品的业务背景，以便测试小组成员了解业务背景，划分测试场景，并站在用户的立场进行测试。

- 开发背景

  描述该项目采用的技术背景。

- 产品目标

  描述产品所需达到的预期目标，基于此可以评估当前架构设计是否能够支持该目标的实现。

## 项目整体分析

- 功能性需求测试分析
  - 术语表

    下表将为您介绍产品需求文档中的术语并给出定义，避免由于对术语理解不一致而导致漏测或错误。

    | 名称 | 说明 |
    | --- | --- |
    | | |
    | | |

  - PRD、指标需求清单与测试功能对应列表

    详细描述数据测试指标需求。

    | 指标名称 | 字段来源 | 业务规则 |
    | --- | --- | --- |
    | | | |
    | | | |

- 系统架构分析

  概括当前项目数据开发总体的流程和范围。

## 测试过程管理

- 测试版本控制

  代码从测试环境发布至开发环境后，需描述此部分。

  项目交付测试通过后，每天上午9点、下午3点接受开发提交的新版本，其他时间测试环境不接受变更。

  | 版本号 | 更新日期 | 触发情况 |
  | --- | --- | --- |
  | | | |
  | | | |

- 测试环境描述

  对测试环境给出逻辑图描述，分析问题和风险。

  例如测试环境和线上环境不一致，可能导致的测试风险。测试环境在一些可能和开发公用的系统，存在的消息分发问题等。

- 测试进入退出准则

  测试进入准则，下表仅描述项目个性化的准则。

  | 任务 | 角色 | 验收标准 |
  | --- | --- | --- |
  | | | |
  | | | |

  测试退出准则，下表仅描述项目个性化的准则。

  | 任务 | 角色 | 验收标准 |
  | --- | --- | --- |
  | | | |
  | | | |

- 测试策略

- 测试设计策略

  描述需要进行的测试，例如功能测试、接口测试等，并分别描述原因。

- 测试执行策略

  描述测试执行需要进行多少轮、每轮的测试重点、每轮测试的优先级，并分别描述原因。

- 回归测试策略

  重点描述整个项目的回归测试策略，不仅包含项目本身，还需要包含其它关联产品线的配合方式等。

- 难点测试方案

- 缺陷管理

  与项目组成员在缺陷处理问题上达成一致，避免测试执行时项目状态过于无序。

  例如XX缺陷必须在两天内修复，如果有拖延，则整个测试依次顺延。

### 困难及风险

基于以上分析，判断项目内存在的风险与困难，并对这些风险和困难进行跟踪直到项目结束，可以参照如下表格：

| 风险描述 | 提出人 | 建议规避措施 | 备注 |
| --- | --- | --- | --- |
| | | | |
| | | | |

## 7.9. 交付测试报告

### 代码交付情况

关键指标包括BUG（每轮测试发现的缺陷总数）、执行率和通过率。

| XX 项目 | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 总用例 | 400 | 交付测试用例 | | 120 | 交付测试周期 | | 2019.03.16-2019.04.16 | | | |
| | | | | | | | | | | |
| 轮次 | 日期 | PASS | FAILED | NO RUN | BUG | 执行率 | 通过率 | 备注 | 交付结果 | 缺陷总数 |
| 1 | 2019.03.16 | 70 | 10 | 40 | 10 | 70% | 58.33% | ** | 不通过 | 12 |
| 2 | 2019.03.29 | 118 | 2 | 0 | 2 | 100% | 98.33% | ** | 通过 | |

### 文档交付情况

| 文档交付情况 | | |
|---|---|---|
| 文档名称 | 交付结果 | 备注 |
| Xxx 项目&升级包产品需求文档 | 不通过 | 项目产品需求文档多处口径不明确 |
| Xxx 项目&升级包系分文档 | 不通过 | 未提交项目系分文档 |
| Xxx 项目&升级包 ADI 设计文档 | 通过 | |
| | | |

### 文档测试准入条件

| 交付测试准入条件报告 | | |
|---|---|---|
| 名称 | 提交结果 | 备注 |
| 测试数据提交 | 未提交 | 开发人员提交测试数据作为测试基础，同时需开发人员先自测。 |
| 计算引擎扫描报告 | 已提交 | 检查开发人员提交代码的结果，不允许有计算引擎无法编译通过的情况。 |

### 交付测试遗留问题

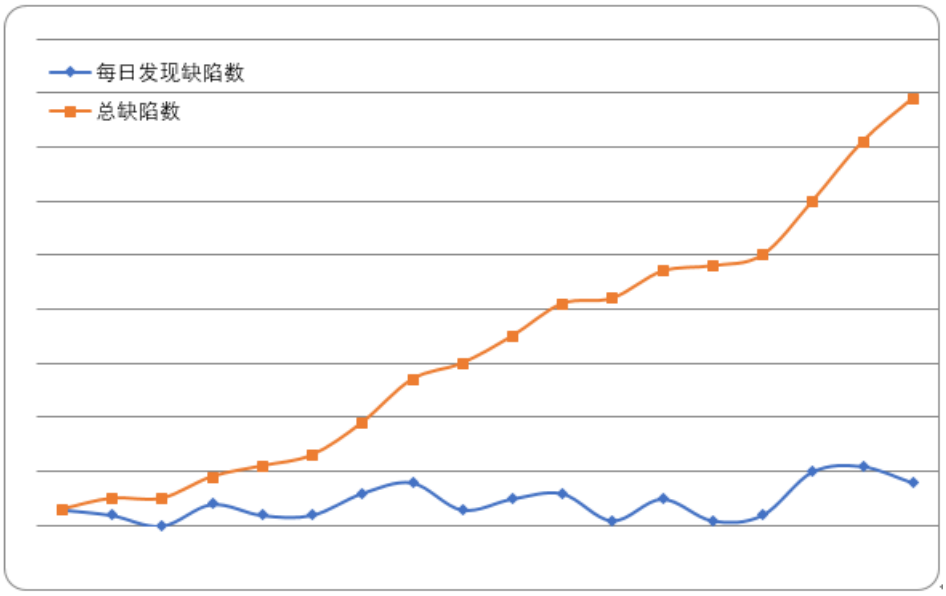
记录交付测试通过后，遗留在功能测试阶段未解决的问题。

| 交付测试遗留问题 | | |
|---|---|---|
| 问题描述 | 处理意见 | 备注 |
| 1. Xxx 口径计算错误 | 由于改动影响面很小，可遗留到项目第一迭代解决 | |
| | | |
| | | |
| | | |
| | | |
| 遗留问题总数 | | 1 |

# 7.10. 质量评估报告模板

### 测试情况说明

- 测试用例执行通过率：0%~100%。
- 每日发现故障趋势图。




- 线下缺陷严重程度分类。

## 需求实现说明

- 需求覆盖率（在测分文档中，需求与功能对应列表为准）：0%~100%。
- 需求变更情况：包括已走正式流程的需求变更，邮件通告的需求变更，以及当前功能改动了原有需求的说明。

| 阶段 | 说明 | 分类 |
| --- | --- | --- |
| 测分阶段 | 增加老会员模式下添加银行卡的出错情况提示。 | 需求变更 |
| | 老会员添加卡的流程中，增加生僻字用户的判断。 | 需求变更 |
| | 增加推荐规则模板：推荐规则为空时的展示方式。 | 需求变更 |

- 未实现需求：请说明需求未实现的原因。

## 遗留问题列表

| 序号 | 问题描述 | 风险影响分析 | 风险等级 | 建议跟进负责人 |
| --- | --- | --- | --- | --- |
| Delay_1 | 由于XX API回参格式限制，XX字段返回结果无法适配计算引擎字段类型。 | 接口改造需花费X天，导致项目整体进度Delay X天。 | 高 | XXX |

## 质量评估结果

- 测试是否通过
- 保留建议

  遗留的问题在本项目中可以接受，但Delay_1缺陷必须在XXX年X月X日之前启动升级包修复。

# 7.11. 验收报告模板

## 测试验收点

| 序号 | 测试验证点（按实际情况增减） | 是否通过 |
| --- | --- | --- |
| 1 | 数据主键是否重复。 | |
| 2 | 结果数据的明细分布，包括数据量、空值、均值及其他相关业务指标的分布。 | |
| 3 | 抽样检查：与需求设定时的抽样样本进行对比，查看是否存在差异。 | |
| 4 | 如果是迭代需求，需要与一期的结果进行对比，查看数据量差异、明细差异等。 | |
| 5 | 某些数值型结果进行同比、环比，获得大概增长率和变化范围，判断数据的正确性。 | |

## 需求实现情况

- 已实现内容。
- 未实现内容：需要说明未实现的原因。

## 发现问题列表

| 序号 | 问题描述 | 风险影响分析 | 风险等级 | 建议跟进负责人 |
| --- | --- | --- | --- | --- |
| Delay_1 | 由于XX API回参格式限制，XX字段返回结果无法适配计算引擎字段类型。 | 接口改造需花费X天，导致项目整体进度Delay X天。 | 高 | 张三 |

## 验收评估结果

业务方（数据产品经理）：通过/不通过。

验收通过。遗留的问题在本项目中可以接受，但Delay_1缺陷必须在xxxx年x月x日之前启动升级包修复。

# 8.运维阶段

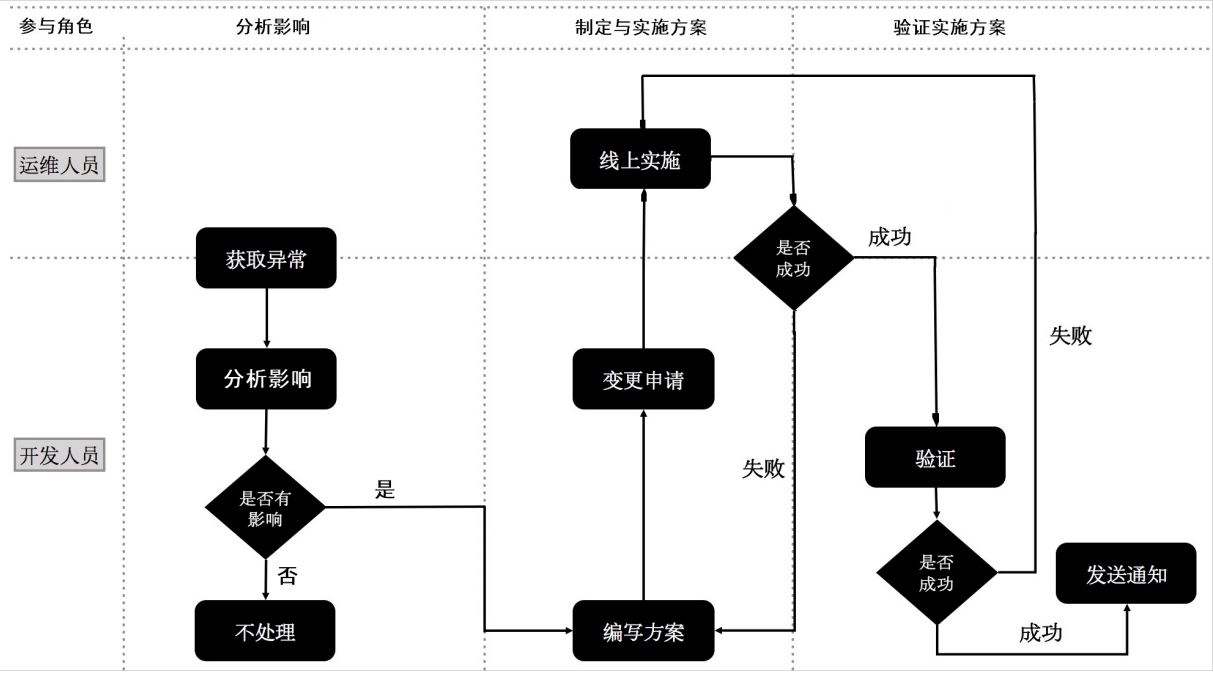
开发人员根据需求将代码发布上线后，还需要及时处理数据、程序、调度、监控告警等的异常事件，保障数据产出时效、程序高效运行和生产稳定性。

## 背景信息

数据开发人员主要需要处理以下事项：

- 程序异常处理、性能优化。
- 调度异常处理。
- 数据质量监控规则异常分析、规则优化。
- 数据异常的核查。

运维阶段的流程包括分析影响、制定与实施方案和验证实施方案。




## 操作步骤

1. 分析影响。

   运维人员或开发人员通过监控规则捕获、自主发现或其它方法获取关于数据产出时效性、数据准确性等指标的异常情况，并进行影响分析。异常情况包括但不限于：

   - 任务运行失败。
   - 任务运行时间过长。
   - 产出表中出现脏数据。

   开发人员根据影响分析的结果判断是否对线上的数据应用有影响。

   - 如果有影响，需要开发人员及时推送告警信息至任务责任人，并判断原因、确定可行性解决方案。
   - 如果无影响，则无需处理。

2. 制定与实施方案。

   i. 开发人员提交线上变更申请。

ii. 审批人员（建议安排为对业务逻辑、代码较为熟悉的人员）审批允许发布变更。

iii. 运维人员按照步骤实施发布，完成后通知数据开发人员进行验证。如果验证失败，则运维人员按照修改脚本的回滚方法进行回滚，并反馈结果至开发人员。

3. 验证实施方案。

开发人员在收到运维人员实施成功的通知后，开始验证变更结果是否符合预期。

- 如果符合预期，则开发人员需要将此次变更的原因、内容及生效时间通知直接下游及关联方的人员。
- 如果未符合预期，则开发人员需要反馈给运维人员执行回滚。