# Стратегия процесса ALCM

## Введение

Данный документ описывает стратегию управления жизненным циклом изменений (**Application Lifecycle Management**, ALCM).  
Цель стратегии — установить единообразный и контролируемый процесс разработки, тестирования и внедрения изменений в модель, минимизируя риски для работающих бизнес-процессов. Данная стратегия применима как к процессу полной, так и частичной миграции.

## Цели стратегии

* Обеспечение бесперебойной работы пользователей в продуктивной среде.
* Управляемость и отслеживаемость всех изменений модели.
* Снижение риска внесения ошибок в продуктивную модель.
* Согласованность действий между разработчиками и пользователями.

## Участники процесса

* **Разработчики моделей** — отвечают за подготовку и внесение изменений в структуру модели.
* **Команда тестирования** — проводят проверку корректности изменений и их влияния на данные.
* **Пользователи** — используют модель в продуктивной среде для операционной деятельности, изменяют только Production объекты.

## Описание сред

## Development (DEV)

Среда разработки, предназначенная для создания и модификации структуры модели.  
Здесь проводятся все работы по изменению модели и первичное тестирование функционала разработчиками.

**Основные характеристики:**

* Используются тестовые данные или ограниченные наборы данных.
* Доступ только для разработчиков.

## Testing (TEST)

Среда тестирования, предназначенная для проверки корректности разработанных изменений на данных, приближенных к продуктивным.

**Основные характеристики:**

* Данные — актуальные или маскированные копии Production.
* Проверяются:
  + расчетные показатели,
  + корректность бизнес-логики.
* Доступ для разработчиков и команды тестирования.

## Production (PROD)

Продуктивная среда, в которой модель используется конечными пользователями.

**Основные характеристики:**

* Работает на актуальных данных.
* Пользователи имеют возможность создавать и изменять только Production-объекты.
* Изменения модели вносятся только путем ALCM миграции после успешного тестирования на TEST.

## Процесс управления изменениями

1. **Разработка изменений**
   * В DEV-модели вносятся необходимые изменения и выполняется локальное тестирование разработчиком.
2. **Миграция изменений в TEST**
   * После завершения разработки изменения переносятся в TEST для проверки на данных, приближенных к боевым.
3. **Тестирование**
   * На TEST проверяется корректность и влияние изменений.
   * В случае выявления ошибок изменения возвращаются в DEV на доработку.
4. **Внедрение и перенос изменений в продуктив**
   * После успешного тестирования изменения переносятся в PROD.
   * Пользователи продолжают работу без потери собственных аналитических объектов.

## Схема процесса

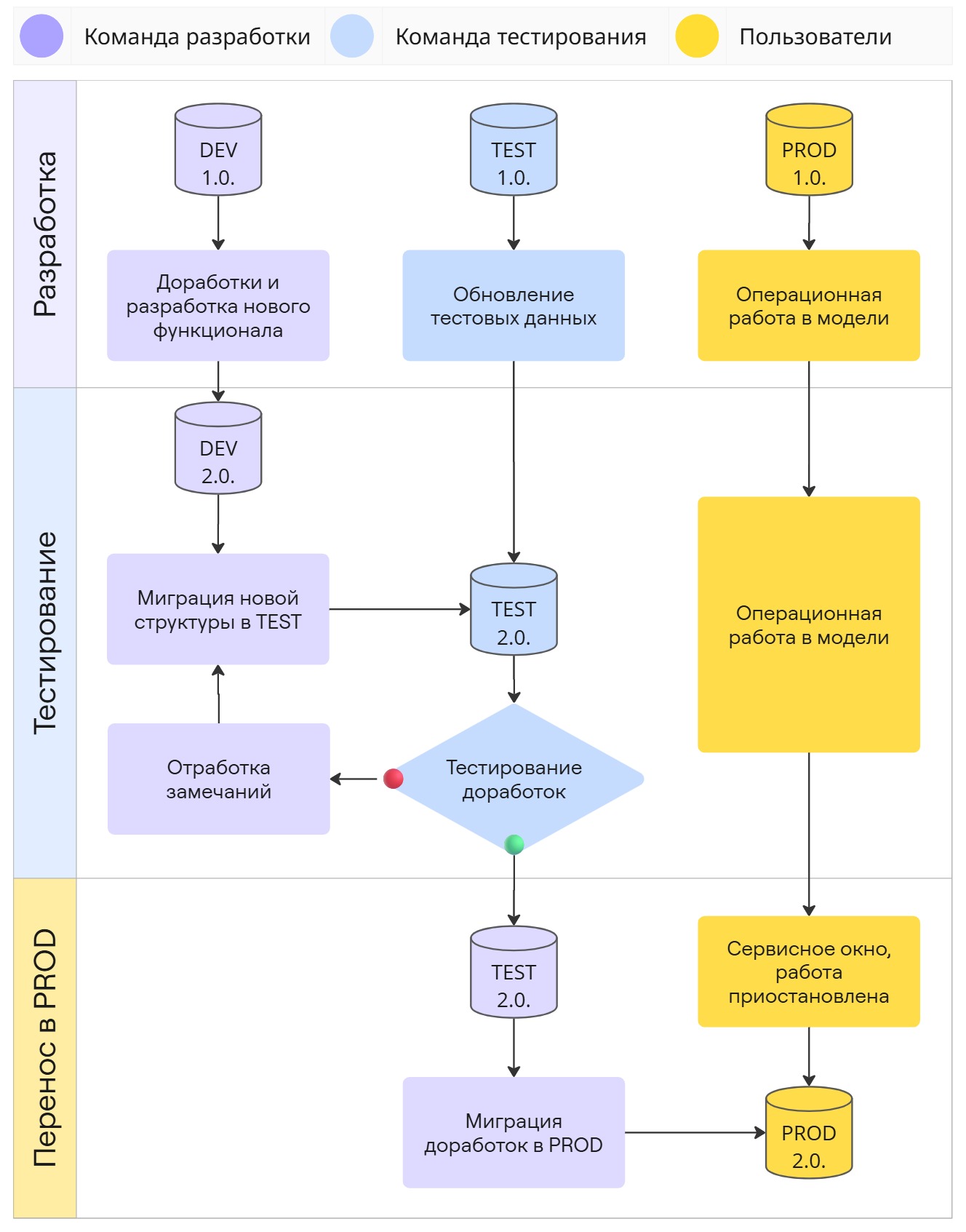


Figure 1 - Схема процесса ALCM

## Ограничения и контроль качества

## Ограничения процесса

Необходимо организационными мерами обеспечить поэтапный процесс доработки моделей. Каждый этап содержит список доработок, сопровождается документированием изменений и согласованием ответственных сторон. В случае если доработка не прошла тестирования или не готова к переносу, она вручную исключается из миграции и переносится на следующий этап.  
Перед переносом в PROD все изменения должны быть утверждены заказчиком.

Изменения в логику расчетов или алгоритмы не могут быть внесены на PROD сервер вне описанного выше процесса.

## Перенос срочных доработок вне цикла миграции

В случае возникновения острой необходимости в доработках PROD-модели вне цикла ALCM миграции, необходимо проделать это доработки в DEV-модели, отметить измененные в рамках внеплановой доработки сущности флагом **Dev complete** (аналогично флагу Production) и произвести частичную миграцию в TEST среду, проверить корректность изменений и затем произвести частичную миграцию в PROD-модель.

**Важно!**

Внесение изменений в PROD-модели параллельно циклу доработок в DEV среде с очень большой вероятностью приведет к невозможности переноса изменений из DEV или TEST модели путем ALCM миграции. Возможные пути выхода из ситуации после внесений изменений в PROD-модели см. [таблице](#_Ошибки_в_ходе) (п.5)

## Rollback план, действия при возникновении ошибок

## Ошибки в ходе миграции

В случае обнаружения ошибок в ходе совершения миграции необходимо проверить известные причины ошибок и убедиться в их отсутствии:

Таблица 1 - Возможные причины ошибок миграции

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Возможная причина** | **Пример** | **Действие** |
| 1 | Падение модели из-за превышения ОЗУ (см. Server Status) | Счетчик оперативной памяти показывает, что воркспейс практически заполнен, в ходе миграции происходит падение модели. | Обеспечить необходимо количество ОЗУ (зеленая зона) |
| 2 | Использование некорректного бэкапа DEV модели | При добавлении бэкапа DEV модели случайно был добавлен некорректный файл (другой бэкап, бэкап другой модели и т.п.) | Загрузить корректный бэкап и повторить миграцию |
| 3 | Несовпадение версий DEV и PROD моделей | Версия DEV среды отличается от PROD | Выровнять версии обеих сред, взять новый DEV бэкап и повторить миграцию |
| 4 | Некорректные параметры настройки миграции | Конфигурации времени мигрируют из DEV, настройки SwitchOver – из PROD. При этом в PROD-модели в SwitchOver используется удаляемая дата. | Указать корректные настройки в соответствии с выполненными изменениями DEV и PROD моделей. |
| 5 | Изменение метаданных в PROD-модели | В PROD-модели были созданы новые сущности (справочники, мультикубы и т.п.) | Способ 1. Откатить PROD-модель на момент ДО изменения метаданных, совершить миграцию, восстановить продуктивные данные вручную.  Способ 2. Перенести доработки с помощью механизма частичной миграции.  Способ 3. Создать новую DEV модель (скопировать PROD) и перенести изменения вручную из старого DEV бэкапа. |

Для выявления причины ошибки необходимо руководствоваться текстом самой ошибки.

В случае падения модели не по причине превышения ОЗУ в ходе совершения миграции, необходимо передать оба бэкапа (DEV и PROD), а также лог падения (Error log) в техническую поддержку вендора.

## Ошибки на этапе тестирования

В случае обнаружения ошибок в данных или логике модели при тестировании и приемке изменений, необходимо откатить модель на бэкап до совершения миграции и передать оба бэкапа (DEV и PROD) на тестирование команде разработки / поддержки для выявления причины возникновения ошибок и их исправления.