

# Netrika.

## Импортозамещение СУБД: ОПЫТ ВЫПОЛНЕНИЯ И АВТОМАТИЗАЦИИ

---

Дмитрий Косарев, Антон Немцев,  
эксперты компании “Нетрика”

Декабрь 2023 г.

# Опыт миграций



Электронный дневник Санкт-Петербурга.  
Модернизация высоконагруженной системы 2012 года с полным изменением архитектуры. Миграция всех данных.



Создание системы «Развитие территории и недвижимости Санкт-Петербурга» с переносом в нее уже существующей системы электронного документооборота с сохранением последовательности транзакций.



Система учета административных правонарушений Санкт-Петербурга.  
Модернизация системы с полным изменением архитектуры. Сведение моделей данных в новом и старом форматах. Миграция данных в новый формат.



Электронное опубликование документов Администрации Ленинградской области.  
Нормализация и объединение 47 баз данных. Модернизация системы с полным изменением архитектуры.

# Реализованные проекты в 2022 году



## АО «Росагролизинг»

Замена Data Warehouse из MS SQL Server 2008 + MS SSIS 2008 на Postgres Pro Standard ver. 14 + Apache Airflow.

**131** процедура, до 12 000 строк кода в каждой

**6 Тб** данных



## Региональная система здравоохранения

Разработка скрипта для автоматического преобразования структуры базы данных из MS SQL Server в PostgreSQL с развертыванием новой базы данных и перекачкой данных.

**2 Тб** данных

# Угрозы использования зарубежных СУБД в компаниях и госструктурах

- **Снижение защищенности ПО и хранимых данных** из-за отсутствия обновлений зарубежных решений
- **Принятие регуляторных актов**, обязывающих организации переходить на отечественное ПО и предусматривающих наказание из-за срыва сроков
- **Риск штрафов и наложения ответственности** (в том числе уголовной) из-за потери данных
- **Риски утечки данных** и блокировки системы вендором
- Вынужденное **использование пиратского ПО**

# Обновления MS SQL Server за последние 2 года

**22 ноября 2021:** четырнадцатое накопительное обновление SQL Server 2019 CU14 — **32 исправления.**

**27 января 2022:** пятнадцатое накопительное обновление SQL Server 2019 CU15 — **36 обновлений.**

**18 апреля 2022:** шестнадцатое накопительное обновление SQL Server 2019 — **43 обновления.**

**14 февраля 2023:** 5 обновлений безопасности, относящихся к **критическим** для SQL Server 2022/ 2019/ 2017/ 2016/ 2014, в том числе уязвимости удаленного выполнения кода.

# Государственные меры по защите инфраструктуры

**С 31 марта 2022:** на значимых объектах КИИ запрещены закупки иностранного ПО и услуг, необходимых для использования этого ПО.

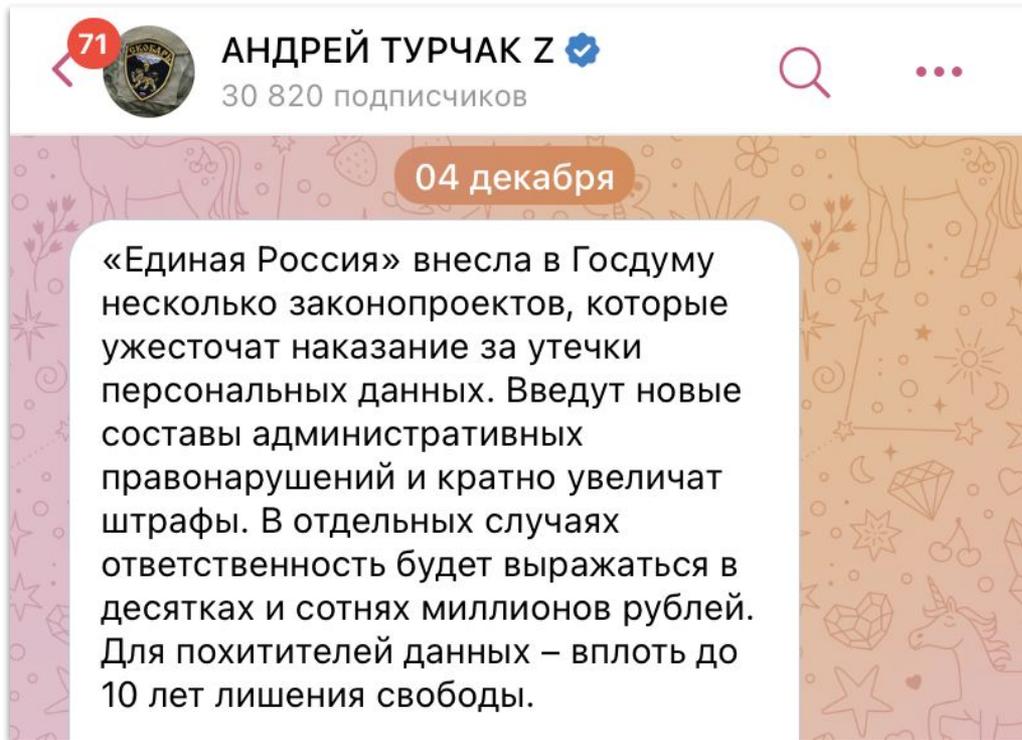
**С 1 января 2025:** на значимых объектах КИИ запрещается использовать иностранное ПО.

**К 1 января 2025:** всем государственным органам власти и госкомпаниям нужно обеспечить тотальное замещение операционных систем, офисного пакета, систем виртуализации, систем управления баз данных.

**Далее:** ужесточение требований будет расширяться для отраслевых системообразующих компаний и коммерческих компаний, обладающих чувствительной информацией и персональными данными пользователей:  
**медицинские, банковские, образовательные информационные системы.**

# Инициативы по ужесточению наказания за утечку персональных данных

Очередной этап ужесточения ответственности за неправомерный доступ к компьютерной информации и нарушение правил хранения и обработки или передачи данных



# Мотивация наших заказчиков



## АО «Росагролизинг»

- Выполнить требования законодательства.
- Отчитаться по соответствующему KPI.
- Обеспечить бесперебойную работу одной из своих ключевых систем.



## Региональная система здравоохранения

- Выполнить требования законодательства.
- Подготовиться к импортозамещению во всех регионах.

# Что сделать для минимизации рисков

1. Определить целевые сроки переезда, учитывающие требования законодательства по отношению к Вашей организации.
2. Понять масштаб работ (бюджет и сроки) и перечень потенциальных партнеров.
3. Определить, на что переезжаем (Тех.особенности + законодательство).
4. Провести организационную подготовку: определить тех. условия проекта и выделить куратора проекта

# Планирование проекта по миграции

## Подбор команды и оценка

1. Найти исполнителей
2. Оценить объем работ
3. Заключить договор

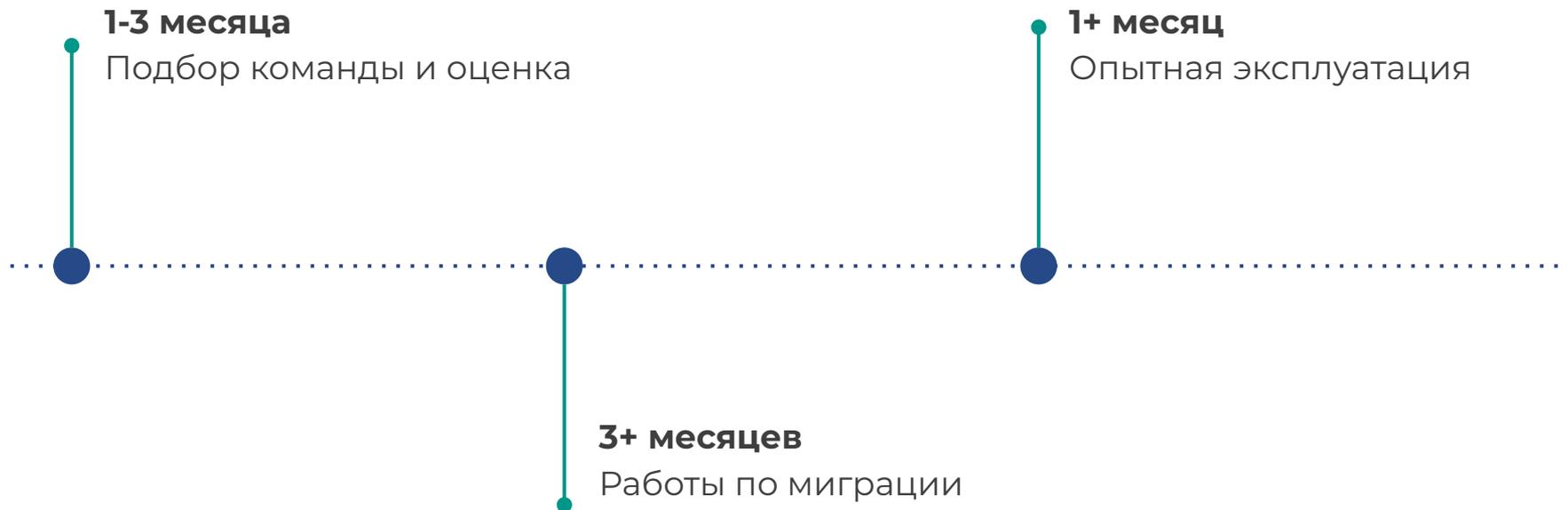
## Опытная эксплуатация

1. Проверка работы на реальных процессах
2. Поиск и исправление ошибок

## Работы по миграции

1. Перепись кода
2. Отладка кода
3. Отладка производительности

# Планирование проекта по миграции



# Кейсы: длительность



## АО «Росагролизинг»

**3 месяца** — подготовка, обследование и заключение договора

**4 месяца** — работы по миграции

**3 месяца** — опытная эксплуатация



## Региональная система здравоохранения

**1 месяц** — оценку и заключение договора

**2 месяца** — работы по миграции

**1 месяц** — опытная эксплуатация

# Как мы оцениваем трудоемкость миграции

1. Берем весь функциональный код в СУБД.
2. Прогоняем его через собственный анализатор кода, который на выходе сообщает количество строк кода со знакомыми и незнакомыми конструкциями.
3. Закладываем единицу времени на каждую строку со знакомой конструкцией.
4. Закладываем бОльшую единицу времени на каждую строку с незнакомой конструкцией.
5. Добавляем коэффициент сложности СУБД в целом.
6. Получаем что-то похожее на правду за короткий срок.

# Оценка трудоемкости миграции

## Определение степени глубины кода, расчет сложности кода

Объем скрипта структуры базы (байт): 6 637 457

Tables:	332	оценка: 192 653	глубина: 192 653 = 3%	мин: 55	макс: 4 929	сред: 580
Alter tables:	589	оценка: 84 408	глубина: 64 929 = 1%	мин: 38	макс: 277	сред: 110
Views:	3	оценка: 473	глубина: 473 = 0%	мин: 73	макс: 257	сред: 158
Procedures:	1 244	оценка: 26 308 955	глубина: 5 261 791 = 79%	мин: 66	макс: 69 511	сред: 4 230
Selects	2 988	оценка: 806 354	глубина: 806 354			
Inserts	162	оценка: 71 804	глубина: 71 804			
Updates	14	оценка: 897	глубина: 897			
Deletes	65	оценка: 10 687	глубина: 10 687			
Functions:	7	оценка: 6 575	глубина: 1 315 = 0%	мин: 155	макс: 213	сред: 188
Generators:	2	оценка: 32	глубина: 63 = 0%	мин: 28	макс: 35	сред: 32
Domains:	23	оценка: 553	глубина: 1 106 = 0%	мин: 37	макс: 72	сред: 48
Exceptions:	6	оценка: 228	глубина: 455 = 0%	мин: 59	макс: 95	сред: 76
Triggers:	131	оценка: 665 285	глубина: 133 057 = 2%	мин: 222	макс: 9 533	сред: 1 016
Selects	124	оценка: 20 997	глубина: 20 997			
Inserts	129	оценка: 13 252	глубина: 13 252			
Updates	29	оценка: 2 037	глубина: 2 037			
Deletes	3	оценка: 174	глубина: 174			
Indecies:	26	оценка: 1 569	глубина: 1 569 = 0%	мин: 43	макс: 77	сред: 60
Describe:	38	оценка: 1 925	глубина: 3 850 = 0%	мин: 41	макс: 252	сред: 101

# Оценка трудоемкости миграции

## Формирование HTML файла с цветовой индикацией кода

```
-- Показать оригинальный код
-- Procedures Оценка: 5 310(939 x 5) = часов:0.142, дней:0.02
CREATE PROCEDURE A1_ORDER_DEBTOR_IDDOC_AIUD0 ()
RETURNS ()
declare debtor_id BIGINT;
declare identity_doc VARCHAR(250);
declare flag INTEGER;
declare type_id INTEGER;
declare type_name VARCHAR(250);
declare series_str VARCHAR(25);
declare number_str VARCHAR(25);
declare note_str VARCHAR(250);
declare s VARCHAR(250);
as
begin
/* Документы, удостоверяющие личность */
for select i.type_id, c.name, i.series_str, i.number_str, i.note_str
from a1_order_debtor_identity doc i, cat_content c
where (i.debtor_id = debtor_id) and (c.id = i.type_id)
order by c.sort order
into type_id, type_name, series_str, number_str, note_str
do
begin
select result, flag
from sys_string_implode(identity_doc, 250, '|', '|', '...', flag)
into identity_doc, flag;
end

if (identity_doc = "") then identity_doc = null;

update a1_order_debtor p set
p.identity_doc = identity_doc
where p.id = debtor_id;
END^
```

comments

insert statements

update statement

# Автоматизации миграции: конвертация кода

## Тепловая карта результатов конверсии

### НАПРИМЕР:

Для переменных меняется синтаксис их декларирования и тип в соответствии с правилами PostgreSQL.

В HTML визуализации результаты этих действий выделяются желтым и зеленым цветами.

```
-- Показать оригинальный код
--Procedures Оценка: 5 310(939 x 5) = часов:0.142, дней:0.02
CREATE PROCEDURE A1_ORDER_DEBTOR_IDDOC_AIUD0 ()
RETURNS ()
declare debtor_id BIGINT;
declare identity_doc VARCHAR(250);
declare flag INTEGER;
declare type_id INTEGER;
declare type_name VARCHAR(250);
declare series_str VARCHAR(25);
declare number_str VARCHAR(25);
declare note_str VARCHAR(250);
declare s VARCHAR(250);
as
begin
/* Документ, удостоверяющие личность */
for select i.type_id, c.name, i.series_str, i.number_str, i.note_str
from a1_order_debtor_identity doc i, cat_content c
where (i.debtor_id = debtor_id) and (c.id = i.type_id)
order by c.sort order;
into type_id, type_name, series_str, number_str, note_str
do
begin
select result, flag
from sys_string_impls(identity_doc, 250, s, ', ', '...', flag)
into identity_doc, flag;
end

if (identity_doc = '') then identity_doc = null;

update a1_order_debtor p set
p.identity_doc = identity_doc
where p.id = debtor_id;
END^
;
```

# Автоматизации миграции: конвертация кода

## Сравнение исходного кода и кода в PostgreSQL

▼ -- Показать оригинальный код

```
CREATE PROCEDURE A1_ORDER_DEBTOR_IDDOC_AIUD0 ()
RETURNS ()
declare debtor_id domain_id64;
declare variable identity_doc domain_vch250;
declare variable flag domain_int;
declare variable type_id domain_catalog;
declare variable type_name domain_vch250;
declare variable series_str domain_vch25;
declare variable number_str domain_vch25;
declare variable note_str domain_vch250;
declare variable s domain_vch250;
as
begin
/* Документы, удостоверяющие личность */
for select i.type_id, c.name, i.series_str, i.number_str, i.note_str
from a1_order_debtor_identity_doc i, cat_content c
where (i.debtor_id = :debtor_id) and (c.id = i.type_id)
order by c.sort_order
into :type_id, :type_name, :series_str, :number_str, :note_str
do
begin
select result, flag
from sys_string_implode(:identity_doc, 250, :s, ',', '...', :flag)
into :identity_doc, :flag;
end

if (identity_doc = '') then identity_doc = null;

update a1_order_debtor p set
p.identity_doc = :identity_doc
where p.id = :debtor_id;
SUSPEND;
END^
```



▼ -- Показать оригинальный код

--Procedures Оценка: 5 310(939 x 5) = часо:0.142, дней:0.02

```
CREATE PROCEDURE A1_ORDER_DEBTOR_IDDOC_AIUD0 ()
RETURNS ()
declare debtor_id BIGINT;
declare identity_doc VARCHAR(250);
declare flag INTEGER;
declare type_id INTEGER;
declare type_name VARCHAR(250);
declare series_str VARCHAR(25);
declare number_str VARCHAR(25);
declare note_str VARCHAR(250);
declare s VARCHAR(250);
as
begin
/* Документы, удостоверяющие личность */
for select i.type_id, c.name, i.series_str, i.number_str, i.note_str
from a1_order_debtor_identity_doc i, cat_content c
where (i.debtor_id = :debtor_id) and (c.id = i.type_id)
order by c.sort_order
into :type_id, :type_name, :series_str, :number_str, :note_str
do
begin
select result, flag
from sys_string_implode(:identity_doc, 250, s, ',', '...', :flag)
into :identity_doc, :flag;
end

if (identity_doc = '') then identity_doc = null;

update a1_order_debtor p set
p.identity_doc = :identity_doc
where p.id = :debtor_id;
END^
```

# Автоматизации миграции: маппинг данных

	A	B	C	D	E	F	G
1	TargetTable	TargetColumn	TargetType	SourceTable	SourceColumn	Примечание	Тестирование
					<pre> CASE poi_type   WHEN 0 THEN 500   WHEN 1 THEN 100   WHEN 2 THEN 600   WHEN 3 THEN 300   WHEN 4 THEN 400   WHEN 5 THEN 200   ELSE 0 END                     </pre>	<p>Исправить. Базируется на легаси public enum PoiType {  STOP("Остановка"),  FUEL("АЗС"),  PARK("Парк"),  START_STATION("ДС"),  FINISH_STATION("КП"),  TEMPORARY("Временная");</p>	<p>В тестовых данных только три типа  Остановка  Парк  Временная  (NB 13.09: Исправлено)</p>
2	pois	type	integer	"bulk"."gppt_nsi_nsi_poi"	END		
3	pois	status	integer	"bulk"."gppt_nsi_nsi_poi"	<pre> CASE activity   WHEN 1 THEN 100   ELSE 200 END                     </pre>	<p>Исправить. Экземпляр класса PoiStatus, а не стандартного RecordStatus.</p>	активные POI
4	pois	location	geography	"bulk"."gppt_nsi_nsi_poi"	geo_data	Широта с долготой поменялись местами	(NB 13.09: Без изменений)
5	vehicle_capacity	name	citext	справочник	Значение из словаря: {"Новая сущность, не имеющая аналогов в легаси системе		
6	vehicle_carrier	binding_date	timestamp with time zone	GPPT_NSI_NSI_TRANSPORT	NULL		
7	vehicle_category	name	citext	-	-	Выглядит как кандидат на удаление	
8	vehicle_model	vehicle_floor_level	bigint	GPPT_NSI_NSI_TRANSPORT_M1		ИЗМЕНЕНИЕ. Задан вопрос заказчику. Ответ: "... проставить всем низкий уровень"	

# Автоматизации миграции: маппинг данных

## ПРИНЦИП РАБОТЫ:

1. Формирование Landing-области для миграции данных путем массового копирования (Bulk loading).
2. Конвертация схемы данных из ORACLE в PostgreSQL 1 в 1 с помощью oracle\_fdw и .NET утилиты.
3. Запуск ETL-скриптов, которые по маппингу переносят данные
4. из PostgreSQL с исходной схемой в PostgreSQL с конечной схемой.

*\* можно было бы обойтись без 2 шага, если бы в БД не было бинарных и геометрических данных*

# Кейс Росагролизинга: подводные камни

Этап	Наименование работ	30.09	07.10	14.10	21.10	28.10	04.11	11.11	18.11	25.11	02.12	09.12	16.12	23.12	30.12
Миграция пакета №1	Миграция общей структуры пакета и разработка в рамках DAG файла			■	■										
	Тестирование пакета и SQL скриптов				■	■	■	■	■	■	■				
	Оптимизация производительности процедур						■	■	■	■	■	■			
Миграция пакета №2 (идентичен №1)	Копирование структуры и тестирование пакета и SQL скриптов											■			
	Миграция общей структуры пакета и разработка в рамках DAG файла				■	■									
	Тестирование пакета и SQL скриптов									■	■				
Миграция пакета №3	Оптимизация производительности процедур											■			
	Миграция общей структуры пакета и разработка в рамках DAG файла.			■	■										
	Тестирование пакета и SQL скриптов				■	■	■	■	■	■	■	■	■	■	■
Миграция пакета №4	Оптимизация производительности процедур						■	■	■	■	■	■	■	■	■
	Миграция общей структуры пакета и разработка в рамках DAG файла.			■	■										
	Тестирование пакета и SQL скриптов				■	■	■	■	■	■	■	■	■	■	■
Миграция пакета №5	Оптимизация производительности процедур											■	■	■	
	Миграция общей структуры пакета и разработка в рамках DAG файла.				■										
	Тестирование пакета и SQL скриптов						■	■	■	■					
Миграция пакета №6	Оптимизация производительности процедур														
	Миграция общей структуры пакета и разработка в рамках DAG файла.				■	■									
	Тестирование пакета и SQL скриптов								■						
Миграция пакета №7	Оптимизация производительности процедур														
	Миграция общей структуры пакета и разработка в рамках DAG файла.				■	■	■								
	Тестирование SQL скриптов и пакета									■					
Миграция пакета №8	Оптимизация производительности процедур														
	Миграция общей структуры пакета и разработка в рамках DAG файла.					■	■								
	Тестирование скриптов и пакета								■						
Миграция пакета №9	Оптимизация производительности процедур														
	Миграция общей структуры пакета и разработка в рамках DAG файла.			■											
	Тестирование пакета и SQL скриптов								■						
Финальное тестирование											■	■	■	■	

# Кейс Росагролизинга: отладка

1. Имена объектов длинные + на русском языке.
2. Функциональные баги в исходном коде.
3. Скрытые баги в исходном коде, например, недетерминированность.
4. Различия приведения типов данных (float -> decimal) и в операции округления.
5. Отсутствие возможностей в Postgres 14: возврат нескольких наборов данных из процедур, merge и др.

Итерационная работа по исправлению кода и в MS SQL, и в Postgres.

# Кейс Росагролизинга: оптимизация

1. Изучали планы выполнения по шагам и выявляли проседания.
2. Заменяли непроизводительные запросы. Outer apply -> оконная функция.
3. Убирали лишние индексы, добавляли нужные.
4. Вводили дополнительную статистику для планировщика запросов.
5. Тюнили сервера Linux и Postgres. Например, параметр read-ahead-kb значительно увеличивает производительность последовательного чтения.

(!) Следите за актуальностью visibility map в Postgres. Иначе потеряете эффект от покрывающих индексов.

**Итог: оптимизировали один запрос с 16 часов до 30 секунд.**

# Кейс Росагролизинга: автоматизация

1. Утилита переписи SQL кода процедур и базы данных in-place.
2. Утилита миграции SSIS пакетов путем генерации DAG файла с тем же наборов step'ов и task'ов.
3. Утилита копирования данных для одинаковой структуры БД, применяется бинарный BULK copy.
4. Утилита сравнения 100% соответствия данных в БД MS SQL и Postgres, а также проверки на 20 контрольных примерах.

Всё самописное. Найденные готовые решения теряют комментарии и форматирование кода.

# Кейс Росагролизинга: опытная эксплуатация

1. Сначала 2 дня переносили текущие данные в новую систему + 1 день строили индексы (не копируйте с включенными индексами - МЕДЛЕННО).
2. На выходных:
  - остановили систему,
  - дозалили новые данные,
  - обновили старые данные,
  - переключили взаимосвязи с внешними системами.
3. Включение обоих инстансов системы:
  - PG работает полноценно, с ним работают внешние системы и аналитика,
  - MS работает на тех же самых входных данных как эталон.
4. Отключение MS по завершению опытной эксплуатации.

# Как упростить себе миграцию

1. Начать делать первые этапы как можно раньше.
2. Использовать инструменты автоматизации, например, от коллег из Postgres Pro.
3. Подключить коллег с опытом миграции. Можно сделать это только на начальных этапах проекта, для автоматизации и ускорения решения технических рисков, а затем дорабатывать результат самостоятельно.
4. Если нужно сделать быстрее - на определенном этапе работу по отладке кода можно спокойно параллелить между разработчиками.

Netrika.

# Есть вопросы?

---

Дмитрий Косарев | +7 (921) 217-79-67

[migrations@netrika.ru](mailto:migrations@netrika.ru)

