

**Производные узлы:
заменяем десятки страниц мануалов
готовыми шаблонами**

Любой разработчик знает

Прекраснее написания
первичной
документации может
быть только
ее актуализация

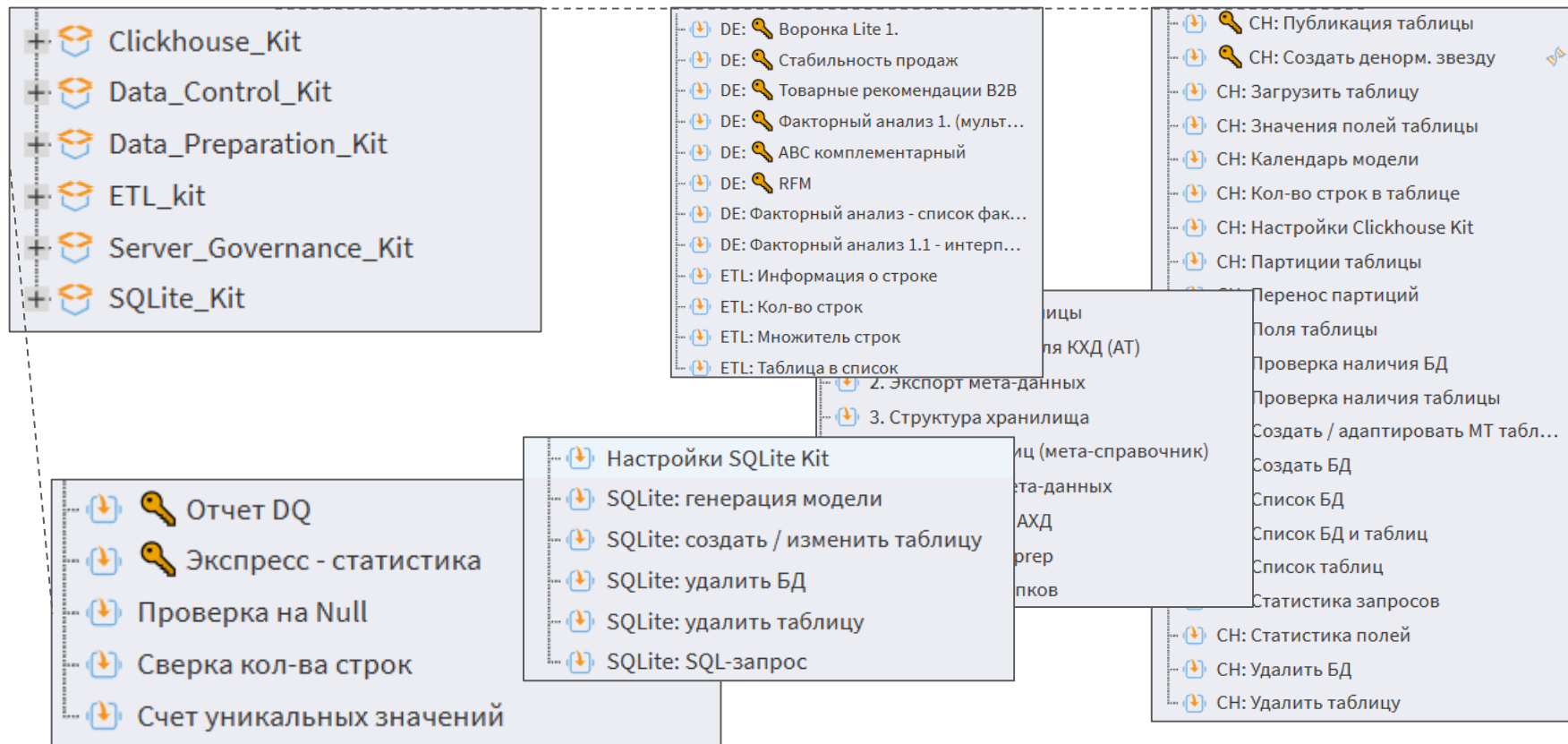


Любой пользователь знает

**Прекраснее инструкции на
100 листов может быть
только видео-инструкция на 3
часа**



Но как научить пользователя работать с десятками компонентов?



Не все комбинации узлов можно привести к адекватным производным компонентам

Сценарий может требовать большого числа наборов данных и настроек. Работать с таким компонентом неудобно.

Плюс, не все параметры узлов могут сценария могут быть доступными через переменные.

Пульт управления миром



Производные узлы (ПУ) позволяют решить эти проблемы

Новый производный компонент

Метка:

Описание:

Область видимости

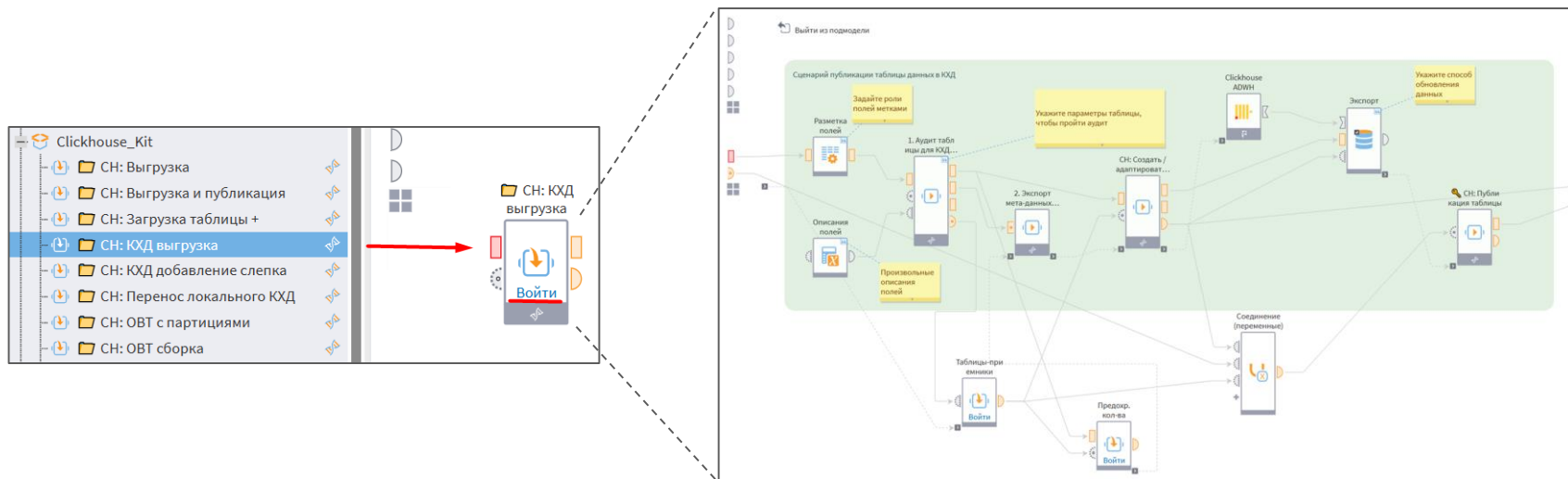
- ☐ Закрытый (доступен только внутри текущего модуля)
- ☐ Защищенный (доступен только внутри текущего модуля и его подмоделей)
- ☐ Внутренний (доступен только внутри текущего пакета)
- ☒ Открытый (доступен во всех пакетах)

Целевые типы узлов

- ☐ Выполнение узла и Цикл
- ☒ Производные узлы

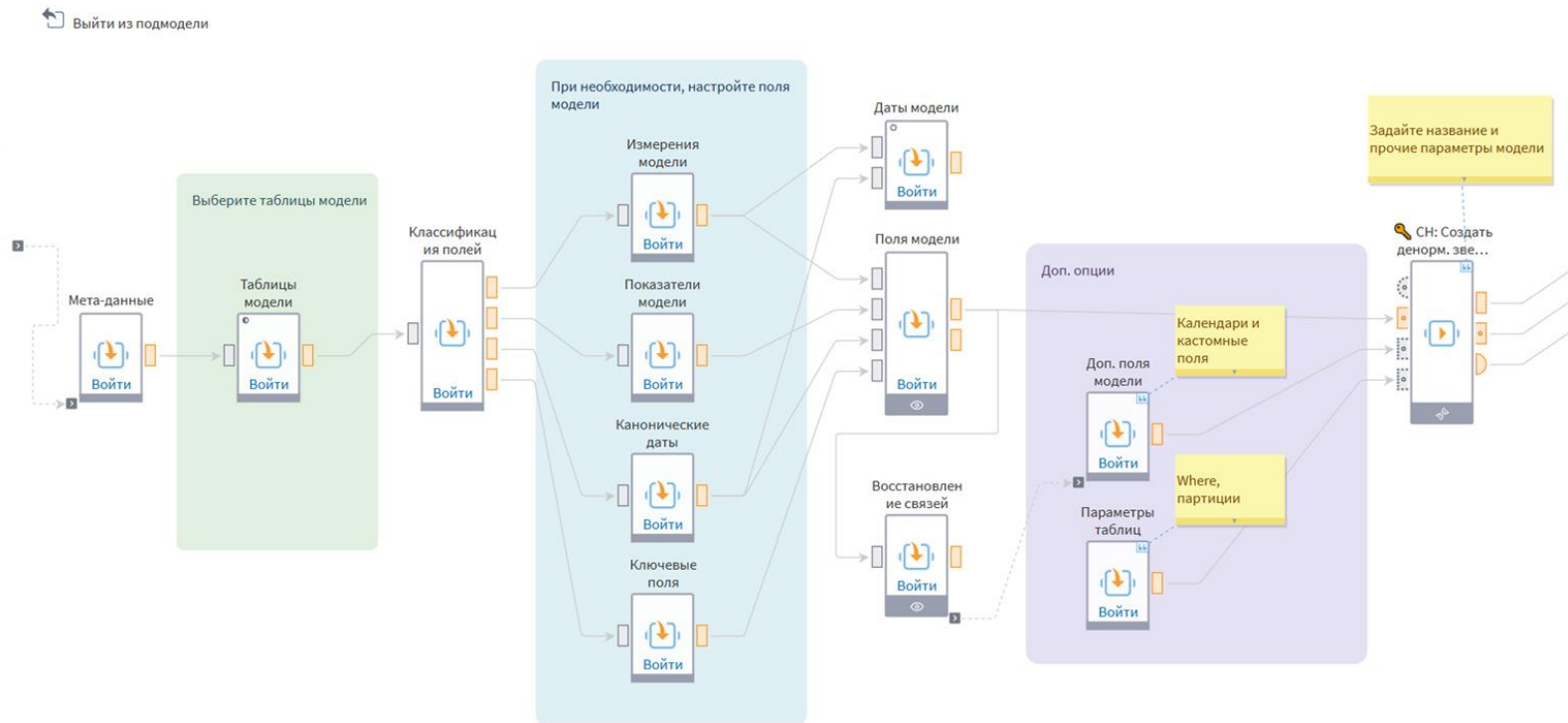
Добавить [Отмена](#)

Пользователь может менять в ПУ параметры компонентов, добавлять свои компоненты, проводить отладку.

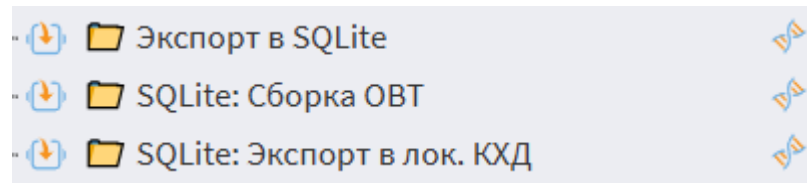
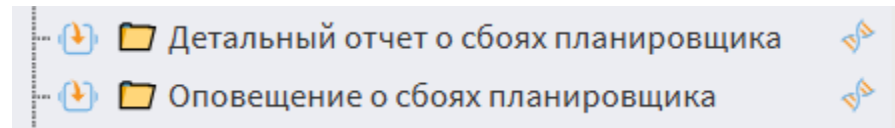
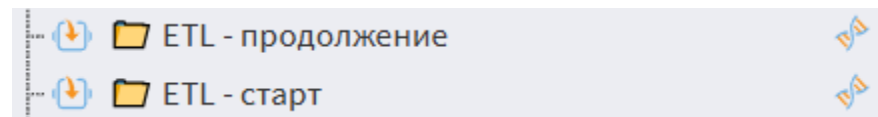
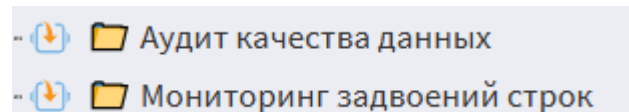
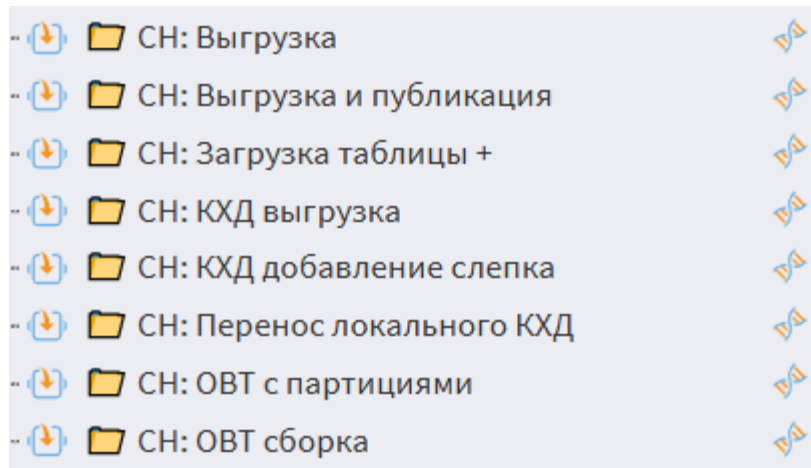


**При этом связь с родительским компонентом остается.
Внесение изменений в него отразится и в ПУ, с сохранением изменений пользователя.**

В ПУ удобно делать “обвес” для упрощения работы с компонентами



Шаблоны используются для решения 90% сценариев



Это не только разные наборы компонентов,
но и одинаковые наборы с разными настройками

Loginom FAQ - библиотека базовых ETL кейсов

Всегда под рукой
Наглядно
Интерактивно



Loginom_FAQ	
1. Построение сценария - основы	
2. Управление набором полей	
3. Создание вычисляемых полей	
4. Упорядочивание сценария и переиспользование	
5. Переменные: создание и использование	
6. Управление порядком активации узлов	
1. Пример ETL-сценария	
2. Склейка одинаковых таблиц БД	
3. Автоматизация работы с БД (пишем свой OUTER UNION)	
Даты: парсинг и трансформация	
Загрузка однотипных файлов	
Замена значений	
Поиск и устранение дубликатов	
Протяжка значений поля	
Сумма накопительным итогом	
Транспонирование таблицы	
Условия (ветки сценария)	
Фильтрация: динамическая, сложные условия	
Цикл 1: For Each	
Цикл 2: Do While	
JOIN / Слияние	
UNION / Объединение	

Что в итоге?

Ускорение разработки (на примере библиотеки DMP) - от 3-х раз;

Порог входа - простейшие инструкции;

Ошибок меньше - меньше нагрузка на ТП;

в 4 раза меньше компонентов;

в 3 раза меньше документация;

Предсказуемое качество финального результата;

Ваши вопросы? :)

Loginom FAQ

