



Бизнес-ценность low-code аналитики согласно теории ограничений

Арустамов Алексей, Loginom Company

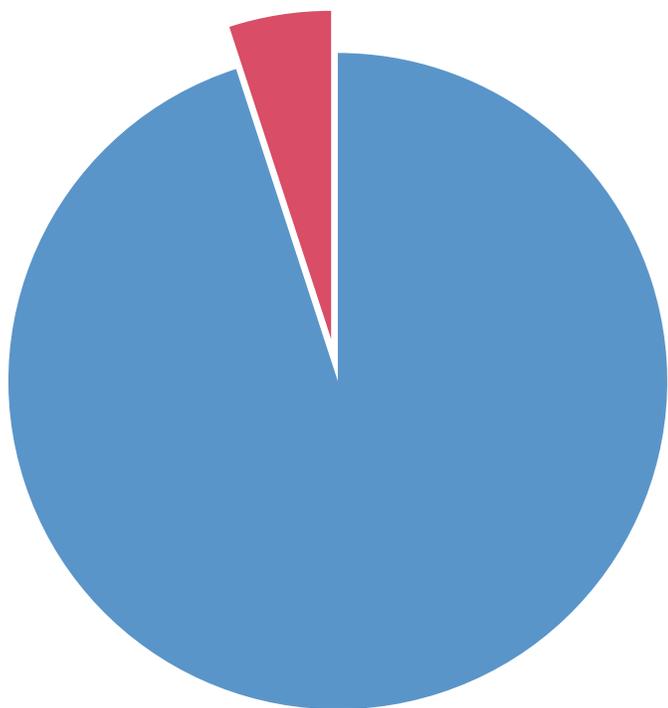
Специализация Loginom Company (ex. BaseGroup Labs):

- Data Mining
- Очистка данных
- Прогнозирование
- Консолидация

Компания в цифрах:

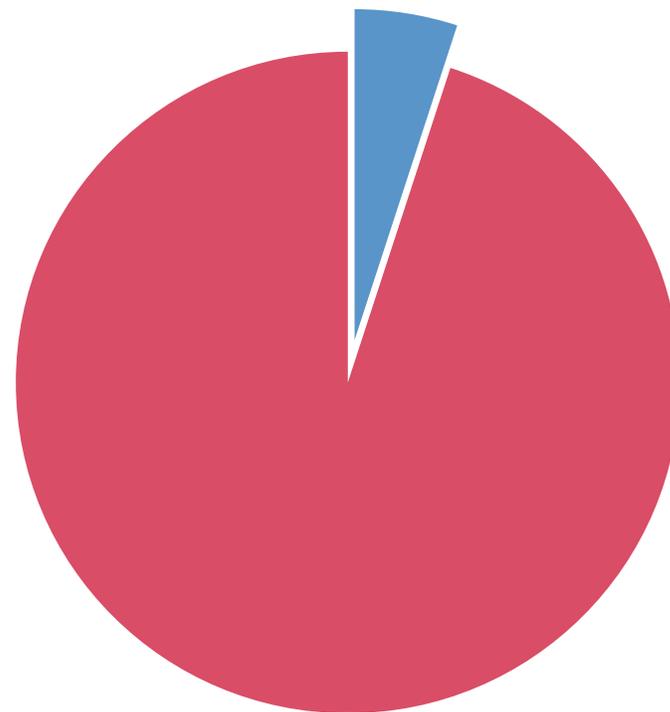
- Продвинутая аналитика с 1999 года
- 150+ проектов
- 3 поколения аналитических платформ

Технологии, о которых
рассказывают на
конференциях

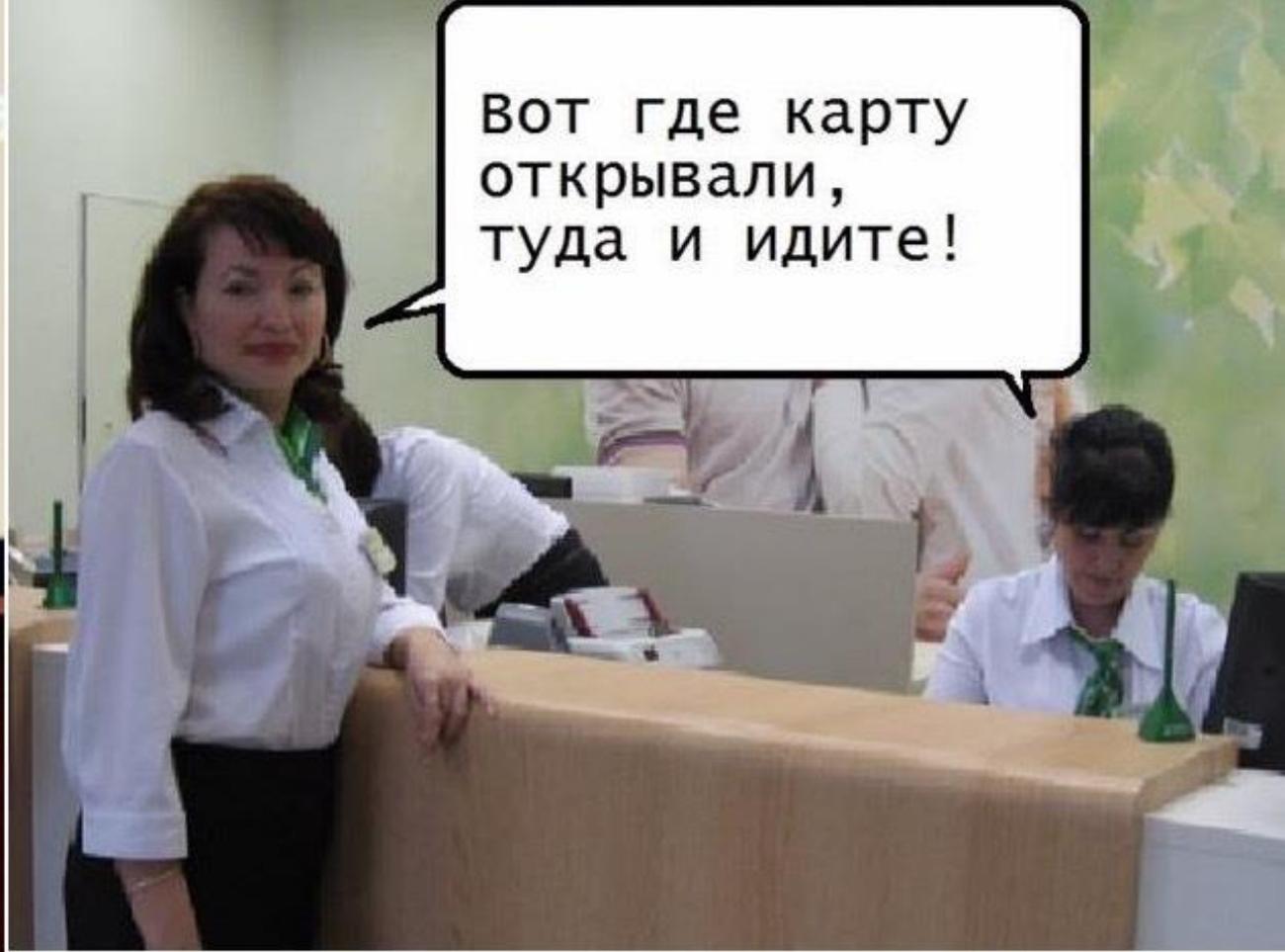


■ Data Science, AI, ML ■ Все остальное

Технологии,
используемые в
бизнес-процессах



■ Data Science, AI, ML ■ Все остальное



Знакомо? Хотя Сбербанк – не худший пример.



3D-печать – нишевая технология
с ограниченным применением



Смартфоны – технология,
изменившая образ жизни людей

example.ipynb

Managed: http://localhost:8888

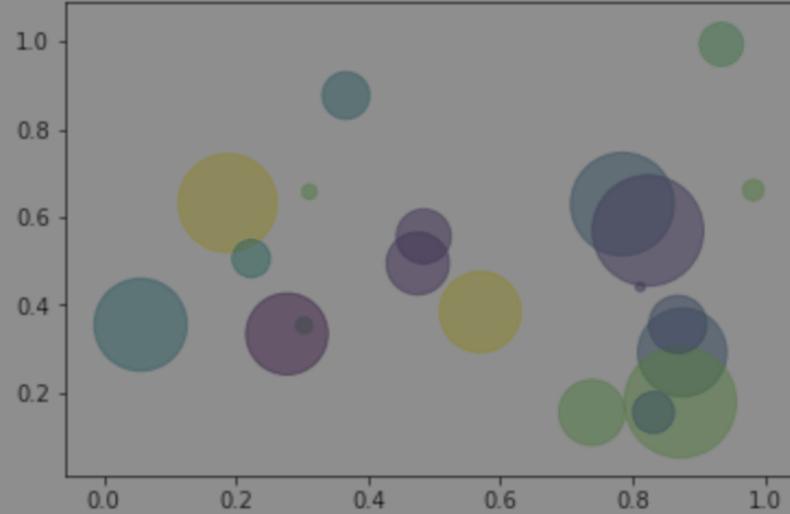
PyCharm (JupyterNotebookSample)

Trusted

```

1  #%% md
2
3  ## Write a formula
4
5  $$c = \sqrt{a^2 + b^2}$$
6
7  #%% md
8
9  ## Draw a scatter chart
10
11 #%%
12
13 import ...
14
15
16 N = 20
17 x = np.random.rand(N)  N: 20
18 print('X-axis values:\n', x)  x: {ndarray: (20,)}
19 y = np.random.rand(N)  N: 20
20 print('Y-axis values:\n', y)  y: {ndarray: (20,)}
21
22 #%% code

```



```

0.15546558 0.87679183 0.62963597 0.33413651 0.56913407 0.15536223
0.35613364 0.65744904 0.50569178 0.35502516 0.55612005 0.49454869
0.66140491 0.38411064]

```

month	average value
0	Jan 3.2

Бесплатные инструменты

– гарантия широкого применения. Неужели?

Jupyter: ...

```

In [ ]: N = (int) 20
In [ ]: area ...
In [ ]: colors = (ndarray: (20,)) [0.61642546 0.59052225 0.33529673 0.2031525 0.67022826 0.72979856, 0.3686584 0.44441121 0.3640378 0.08879216 0.21618187 0.71490...View
In [ ]: sample_data_frame = {DataFrame: (3, 2)} month average value [0: Jan 3.2], [1: Feb 5.2], [2: May 6.5] ...View as DataFrame
In [ ]: x = ...
In [ ]: y = (ndarray: (20,)) [0.35372327 0.63228352 0.29172777 0.44142282 0.99323923 0.17929492, 0.15546558 0.87679183 0.62963597 0.33413651 0.56913407 0.15536223...View
In [ ]: Special Variables

```

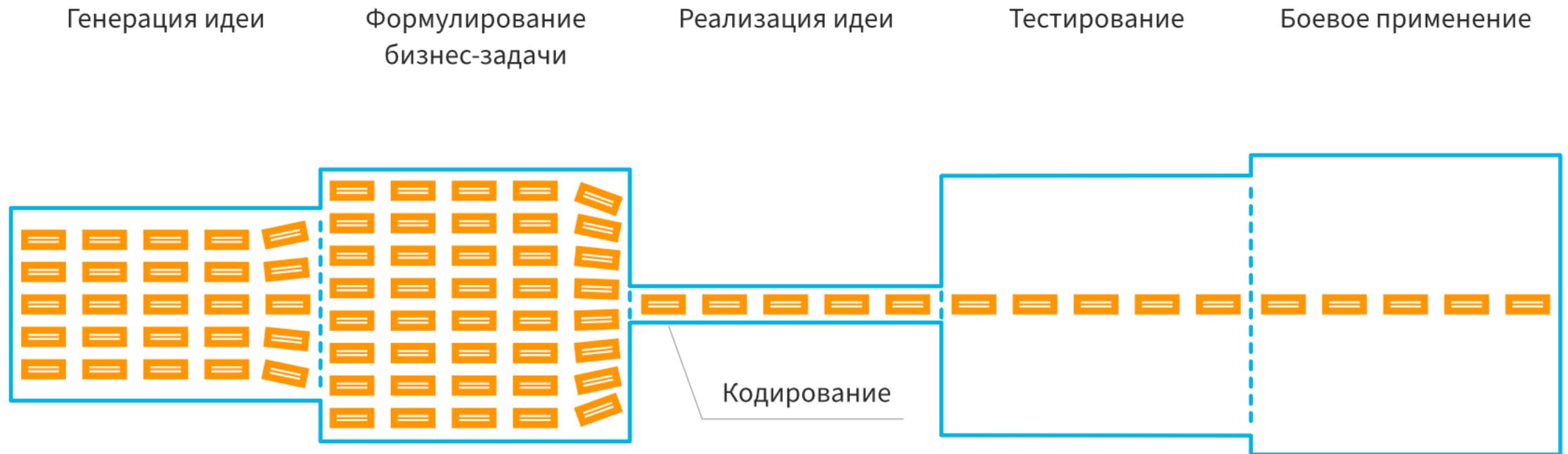


Теория ограничений – методология управления системами, базирующаяся на поиске узких мест, определяющих успех и эффективность всей системы в целом.

Эффект от управления узким местом намного превышает результат одновременного воздействия на все или большинство проблемных областей системы сразу или поочерёдно.

Процесс внедрения Data Science в бизнесе





В любом процессе есть одно узкое место, только оно ограничивает пропускную способность системы

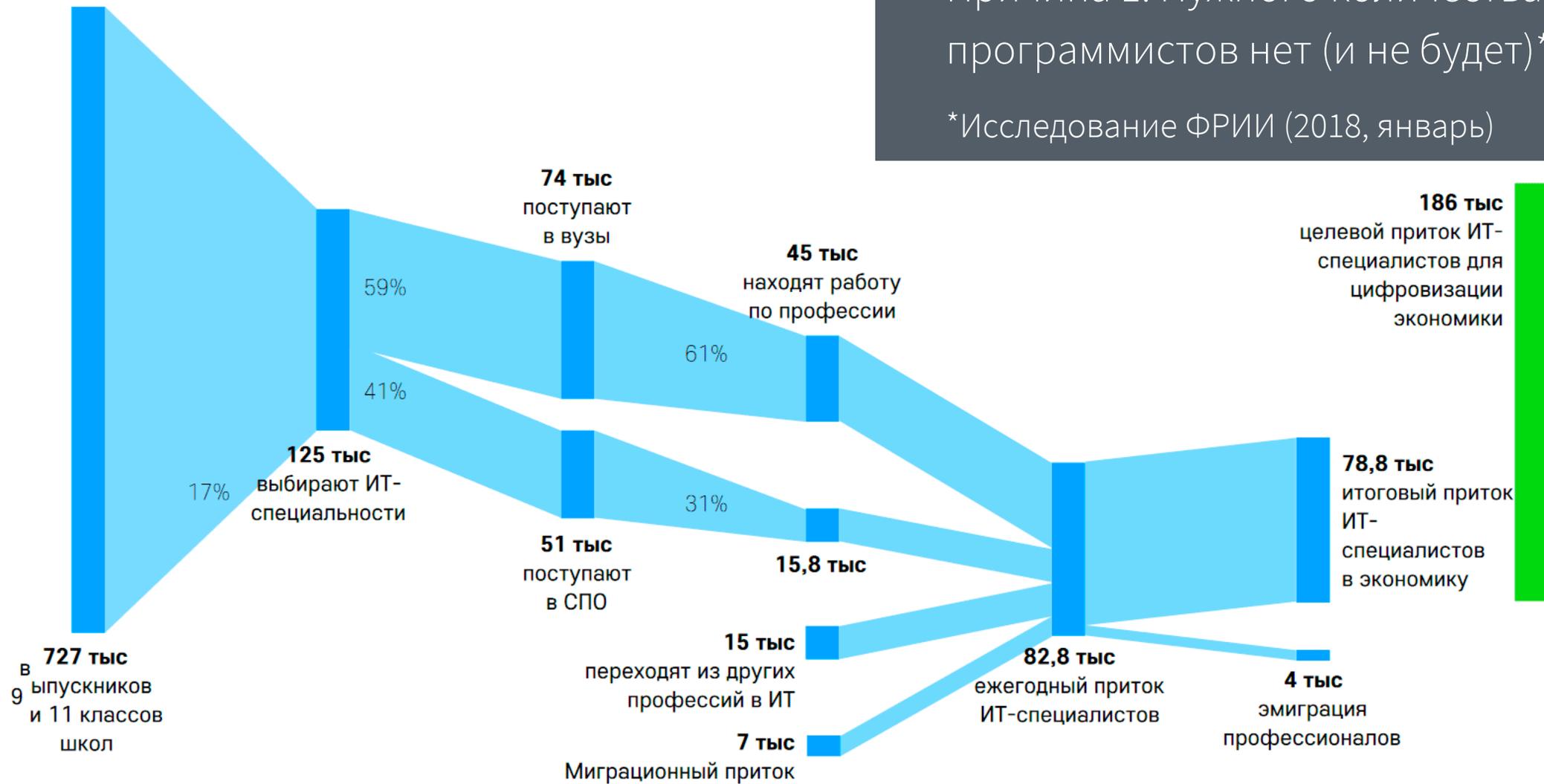
Пример Скрам-доски						
Бэклог Продукта	Бэклог Спринта	В процессе	Взаимная проверка	На тестировании	Готово	Заблокировано
10	2	1	1			1
10				2		
10	2	1	2			
10					2	
10	2					

Маркер «узкого места»

Согласно теории ограничений, перед узким местом «конвейера» больше всего необработанных задач – это типичная ситуация для IT департаментов 95% компаний.

Причина 1: Нужного количества программистов нет (и не будет)*

* Исследование ФРИИ (2018, январь)



Структура формирования ежегодного притока ИТ-специалистов в экономику России



5/

Валидация методики замера эффектов

93-95% доля неэффективных проектов в цифровой сфере в 2018-2019 гг.*



Методология оценки и ее результат **подтверждены** независимым ревью

Что подтверждаем:

- Дизайн эксперимента
- Сопоставимость контрольной и тестовой групп
- Контроль воздействия внешних факторов на результат
- Математическая достоверность теста
- Робастность/устойчивость замера эффекта
- Воспроизводимость результата



Методология оценки и ее результат **подтверждены с ограничениями**
Рекомендуется дополнительный контроль эффектов проекта

Возможные ограничения:

- Широкий доверительный интервал эффекта
- Не обеспечена полноценная сопоставимость контроля и теста
- Нет высокой устойчивости результата
- Отсутствие возможности полноценного учета сезонности
- Другое



Методология оценки и ее результат **не подтверждены**
Не рекомендуется руководствоваться данным замером эффекта при принятии решений

Причины результата:

- Замер эффекта **не корректен**: контроль и тест строго не сопоставимы
- Статистическая заявленного эффекта **не значимость** **подтверждается**
- Результат не воспроизводим: **ошибки** в коде, выводе результата
- Результат обеспечен внешними факторами: погода, инфляция, и др.

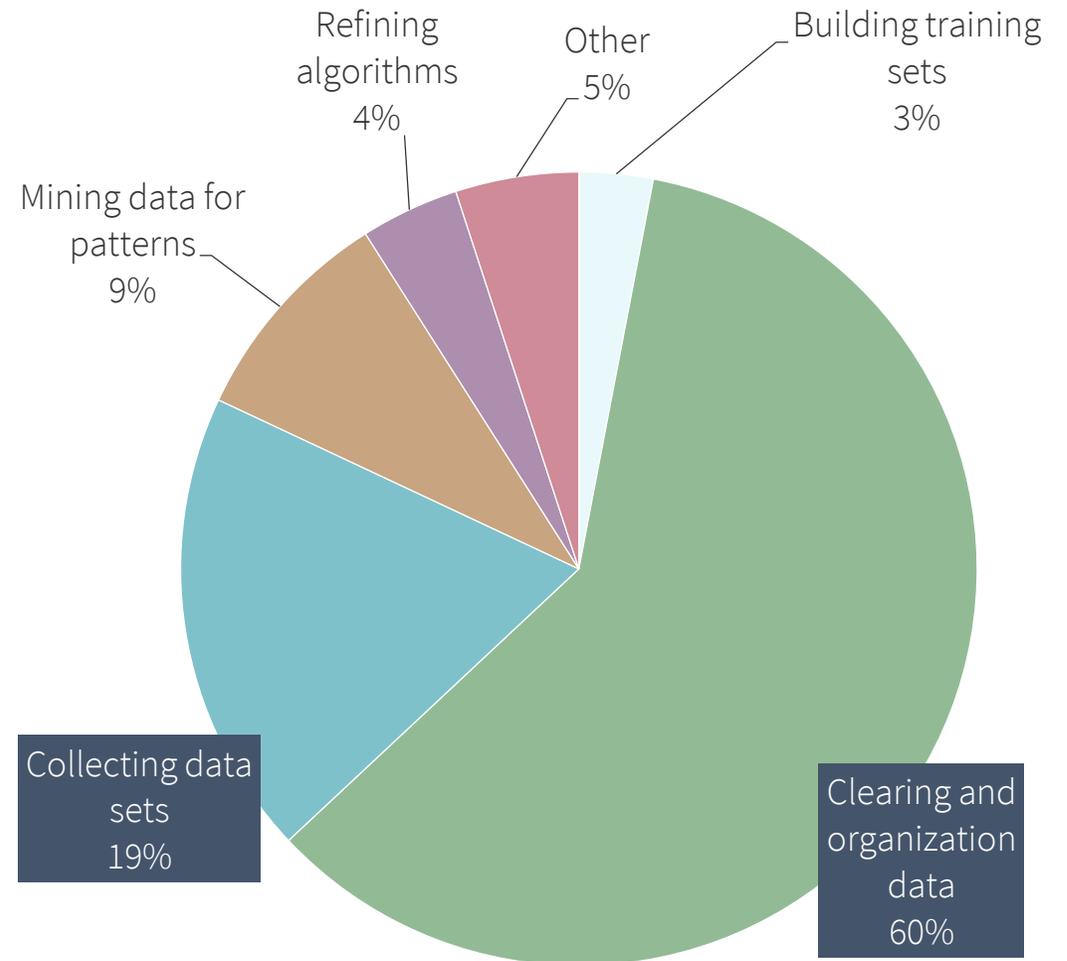


* По данным форума RAIF от октября 2019 г, «Инфосистемы Джет»



Причина 2: Более 90% идей попадет в корзину, но понять это можно после реализации.

Причина 3: 80%
времени проекта –
не анализ, а сбор,
организация и
очистка данных



Эффективное
время

Доля работы в
корзину

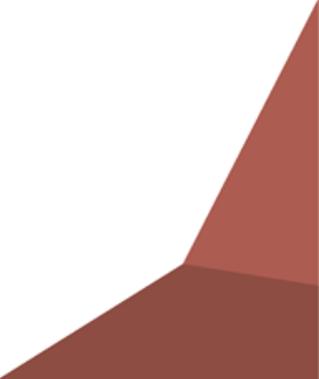
КПД работы
разработчика – 2%

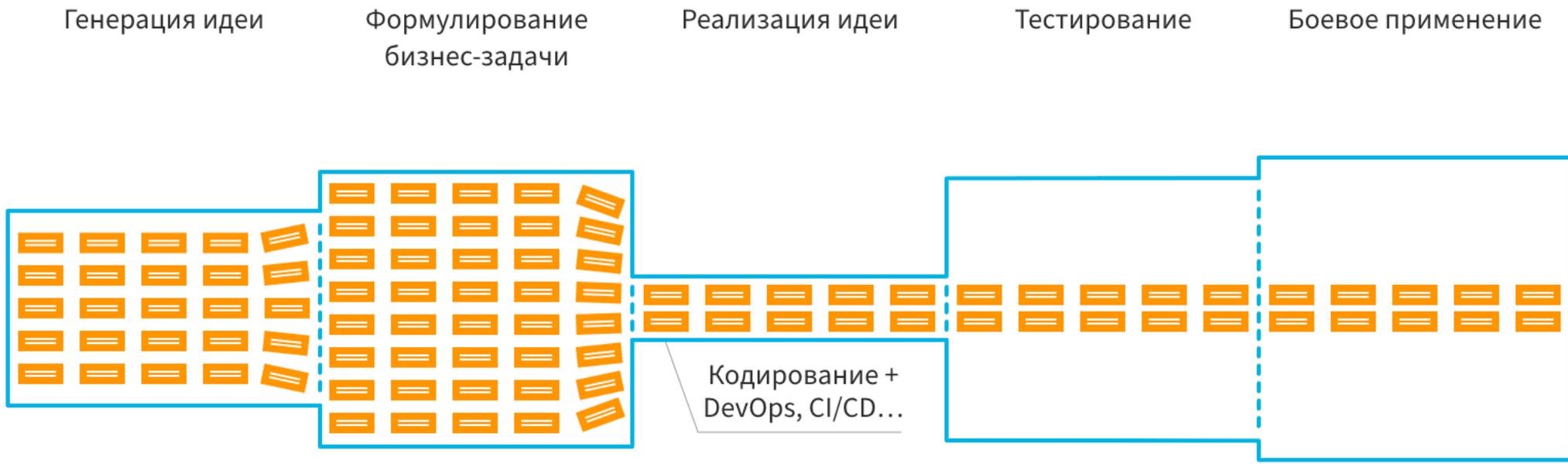
$$Y = X * (1 - 0.9) * (1 - 0.8) = 0.02 * X$$

Дефицитное
время
разработчика

Доля
подготовительной
работы

Решение 1: набор программистов и повышение эффективности их труда:

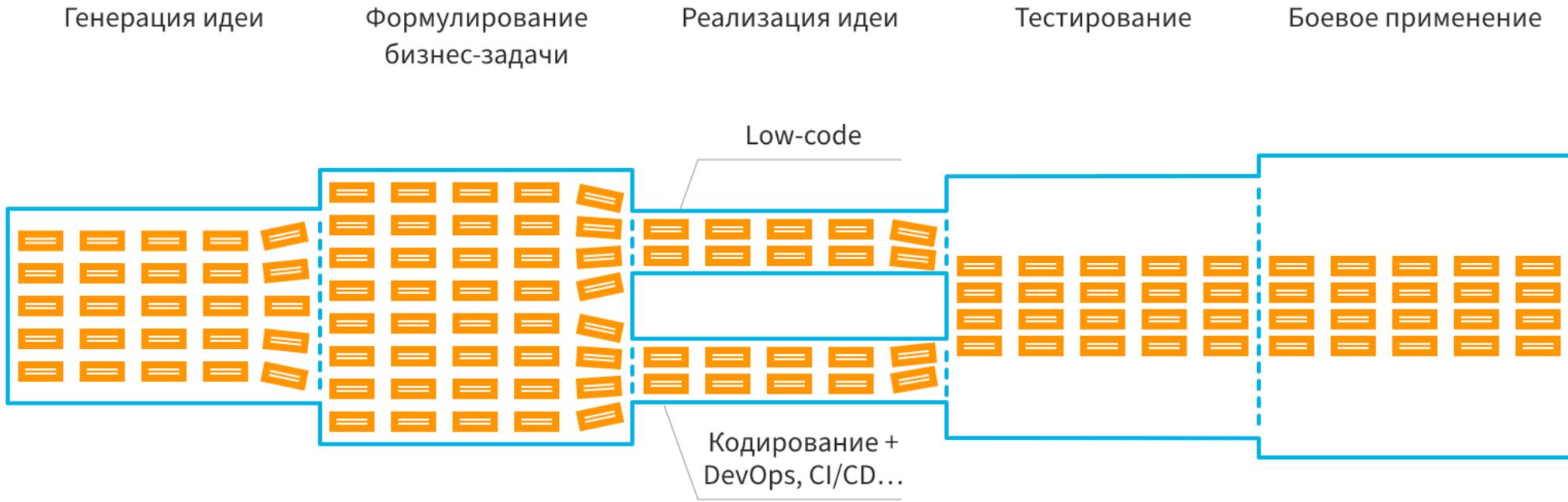
- DevOps, CI/CD, Автотесты
 - Готовые библиотеки
 - Agile, Scrum, Kanban
 - Облачные технологии
- 



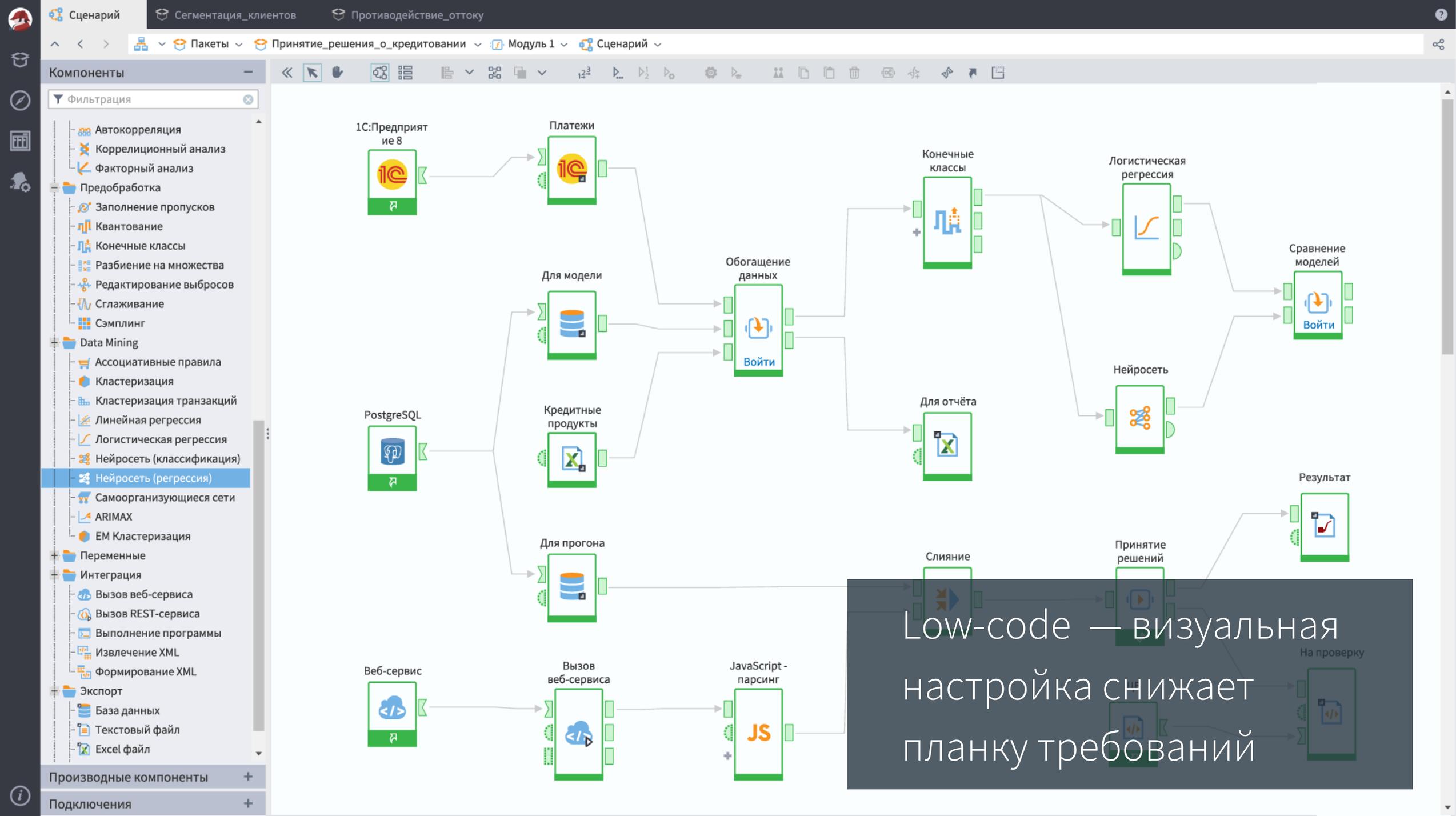
Набор разработчиков, DevOps, CI/CD, готовые библиотеки и прочее расширяют бутылочное горлышко

Решение 2: low-code – решение задач анализа непрограммистами. Подход позволяет бизнес-экспертам большую часть работ выполнять самим, разгружая дефицитный ресурс для другой работы.

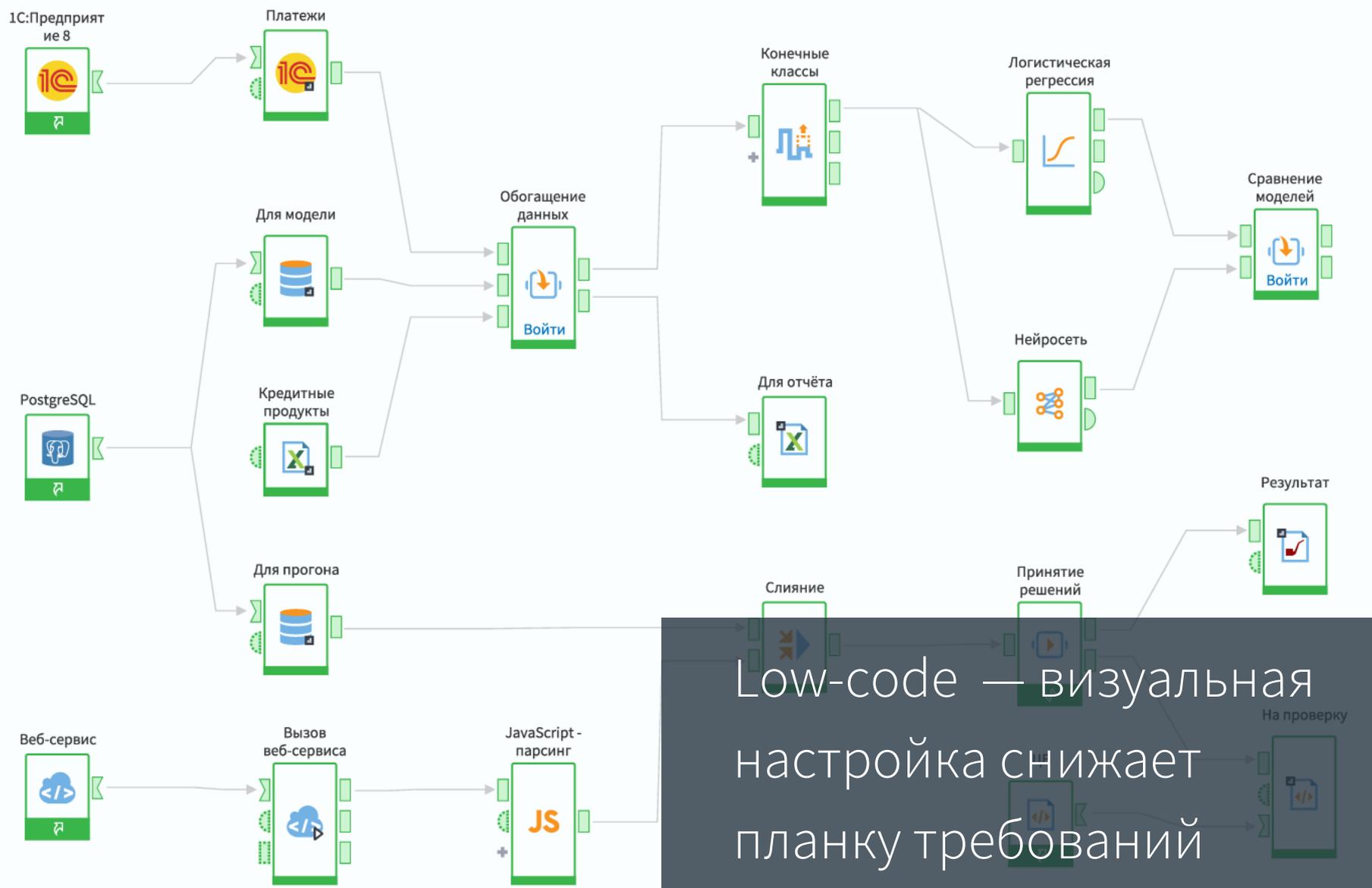




Low-code создает параллельный поток работ для решения задач аналитики



- Автокорреляция
- Корреляционный анализ
- Факторный анализ
- Предобработка
 - Заполнение пропусков
 - Квантование
 - Конечные классы
 - Разбиение на множества
 - Редактирование выбросов
 - Сглаживание
 - Сэмплинг
- Data Mining
 - Ассоциативные правила
 - Кластеризация
 - Кластеризация транзакций
 - Линейная регрессия
 - Логистическая регрессия
 - Нейросеть (классификация)
 - Нейросеть (регрессия)**
 - Самоорганизующиеся сети
 - ARIMAX
 - EM Кластеризация
- Переменные
- Интеграция
 - Вызов веб-сервиса
 - Вызов REST-сервиса
 - Выполнение программы
 - Извлечение XML
 - Формирование XML
- Экспорт
 - База данных
 - Текстовый файл
 - Excel файл



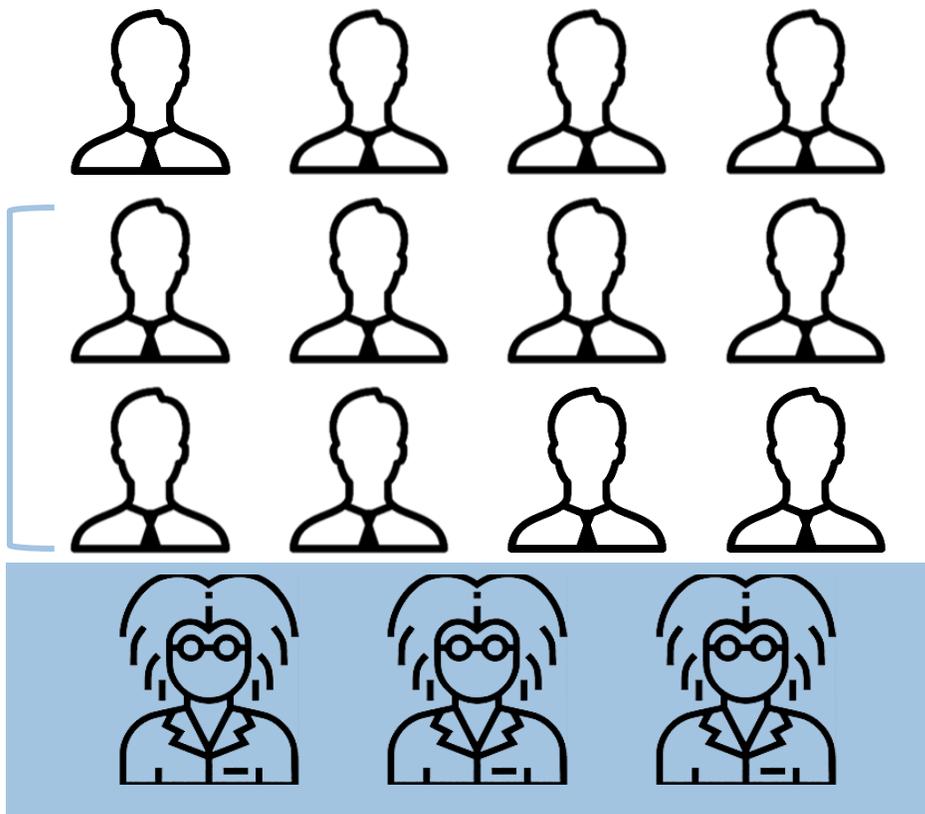
Low-code — визуальная настройка снижает планку требований



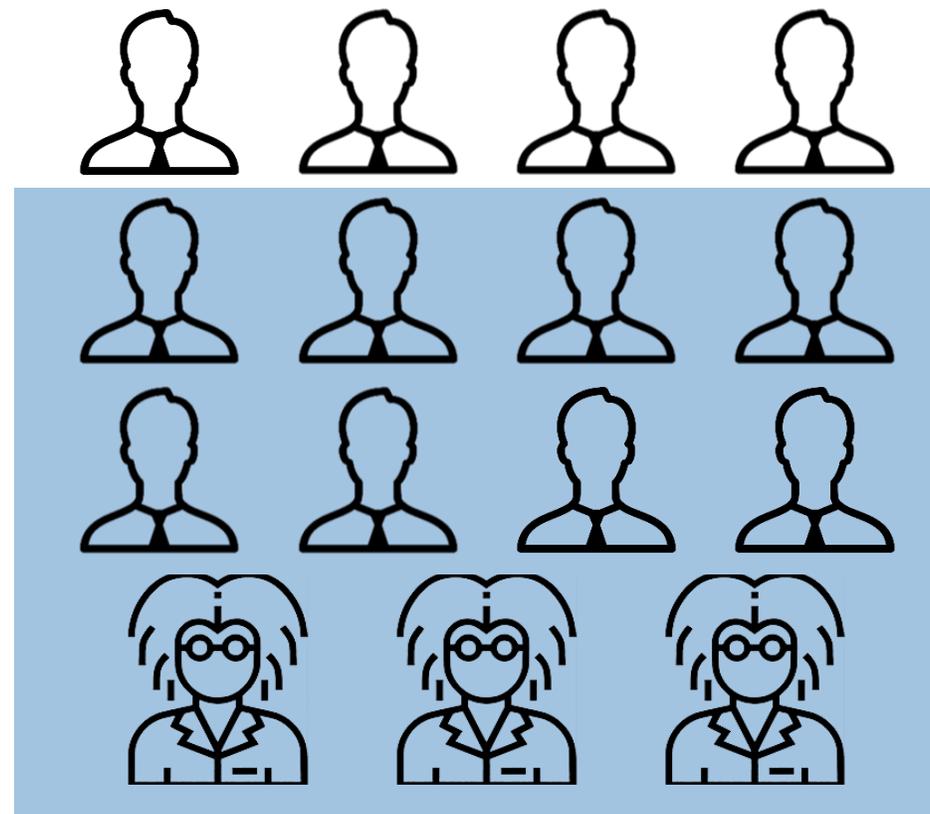
Low-code –
способ решения
сотен задач
продвинутой
аналитики

1. Кредитный скоринг
2. Анализ рисков
3. Антифрод
4. Прогнозирование спроса
5. Вероятность оттока клиентов
6. Очистка и дедупликация данных
7. Расчет сложных KPI
8. Кросс-продажи
9. Сегментация клиентов
10. ...

Потенциал



Data scientists



Data scientists +
Citizen Data scientists

Меньше работы «в урну»:

- Быстрая реализация идей без очередей в IT департамент
- Оценка адекватности идеи за счет знания предметной области
- Способность находить «зацепки» в данных и генерировать новые гипотезы

Экономия времени за счет
самостоятельной подготовки данных:

1. Интеграция с разнородными системами
2. Оценка качества данных с учетом бизнес-логики
3. Понимание бизнес-процессов, приводящих к таким данным

Low Code ≠ No Code

Low-code – не тотальный отказ от кодирования, а его минимизация.

Low-code – не замена разработке, а лучшая инвестиция с точки зрения программистов:

- Разработчики освобождаются от рутины, пользователи многое выполняют сами.
- Гипотезы тестируются без программистов. Они концентрируются на апробированных идеях.
- Программисты выполняют наиболее сложные задачи, которые кроме них никто не решит.
- Сложные блоки, реализованные при помощи кода, вызываются из low-code сценариев.

Low-code может совершить революцию на Data Science рынке за счет массовости:

- Визуальное проектирование увеличивает количество пользователей.
- Аналитики смогут применять знания без посредников в виде разработчиков.
- Самые сложные задачи можно закрыть кодированием.



Давайте обсудим, как организовать взаимодействие между программистами и экспертами, использующими Low-code, чтобы повысить реальное применение Data Science в бизнесе.



Арустамов Алексей Loginom Company

alexey.arustamov@loginom.ru

<https://www.linkedin.com/in/alexey-arustamov/>