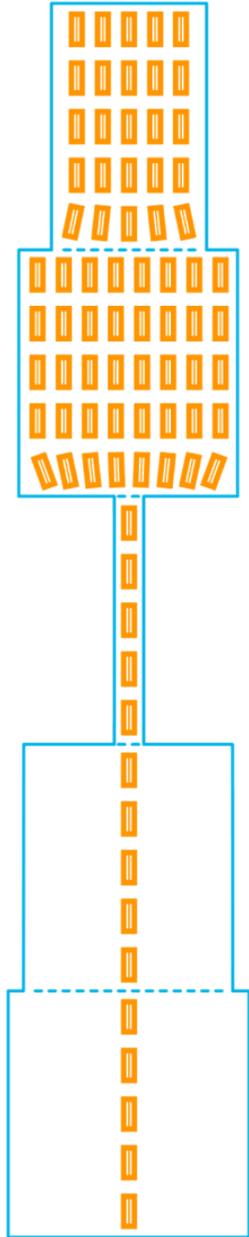


Бизнес-ценность low-code аналитики

Арустамов Алексей, Loginom Company

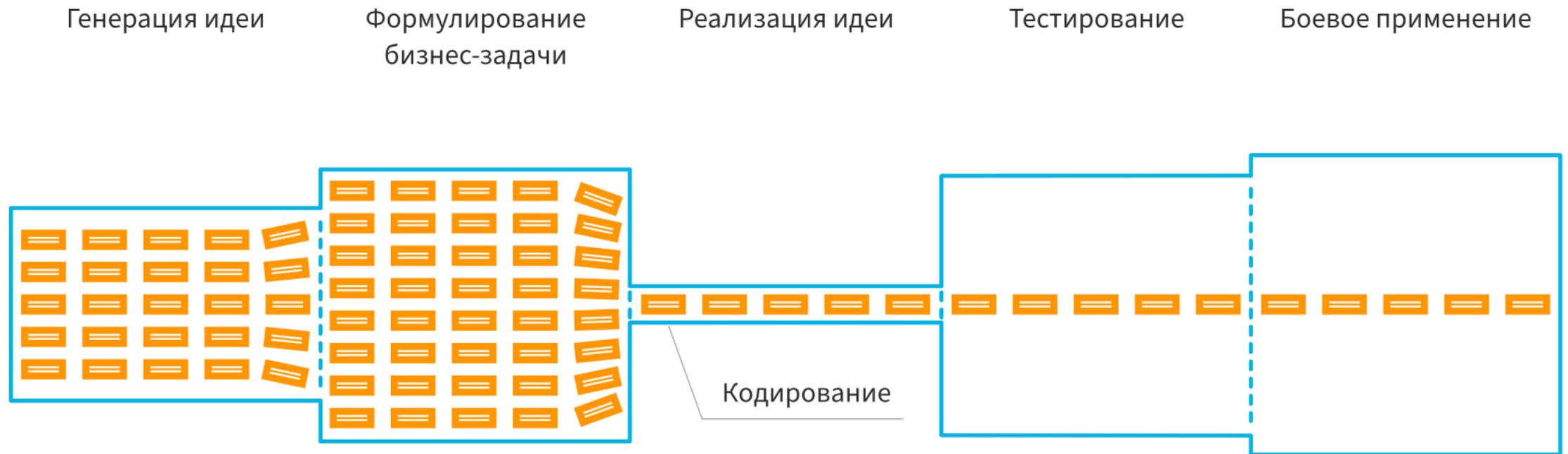


Теория ограничений – методология управления системами, базирующаяся на поиске узких мест, определяющих успех и эффективность всей системы в целом.

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

Бизнес-процесс аналитики данных





При ставке на кодирование дефицит ресурсов в разработке является «узким местом»

Пример Скрам-доски						
Бэклог Продукта	Бэклог Спринта	В процессе	Взаимная проверка	На тестировании	Готово	Заблокировано
10	3	1	1			1
				2		
	2	1	2			
					2	
	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: Дефицит IT-кадров в России составляет от 500 тыс. до 1 млн. человек. В ближайшие 5 лет тренд на цифровизацию потребует еще примерно 2 млн. специалистов.



Минцифры
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



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



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

Что подтверждаем:

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



Возможные ограничения:

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

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.

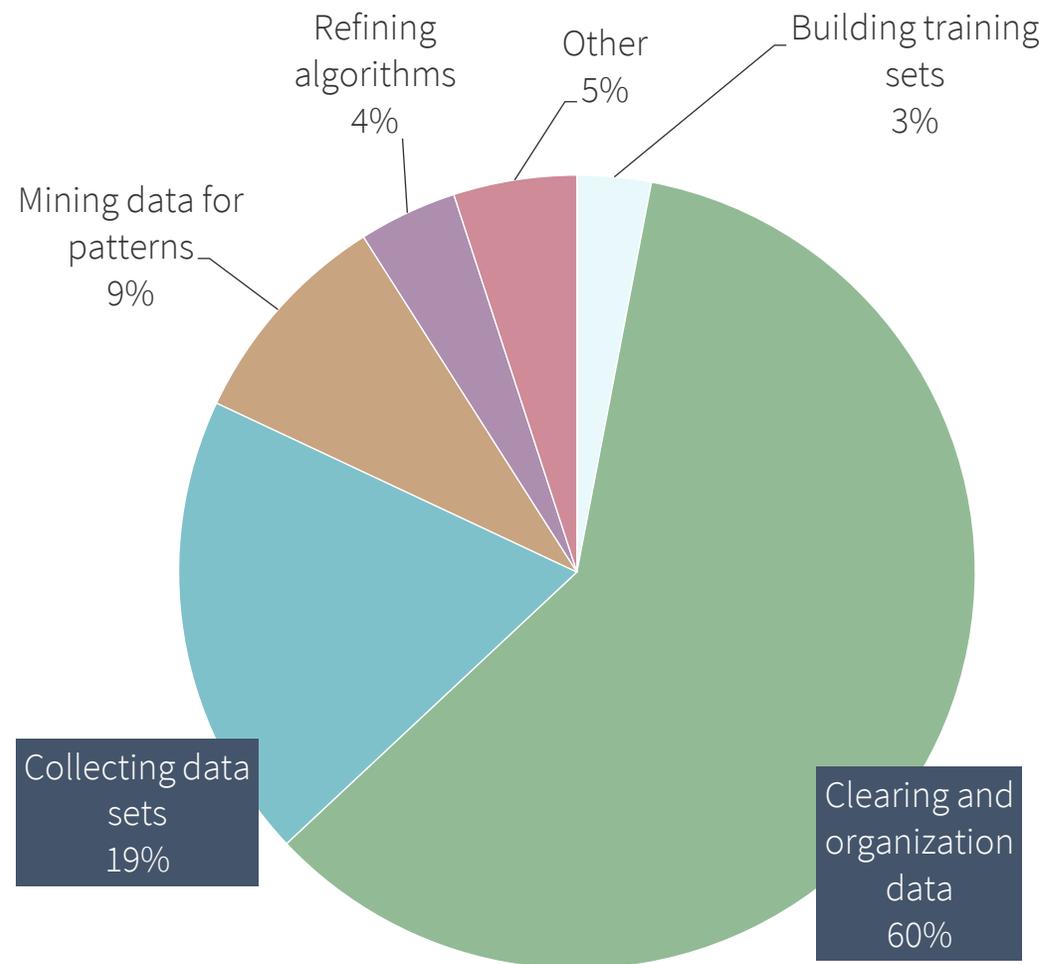
некорректного

* 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

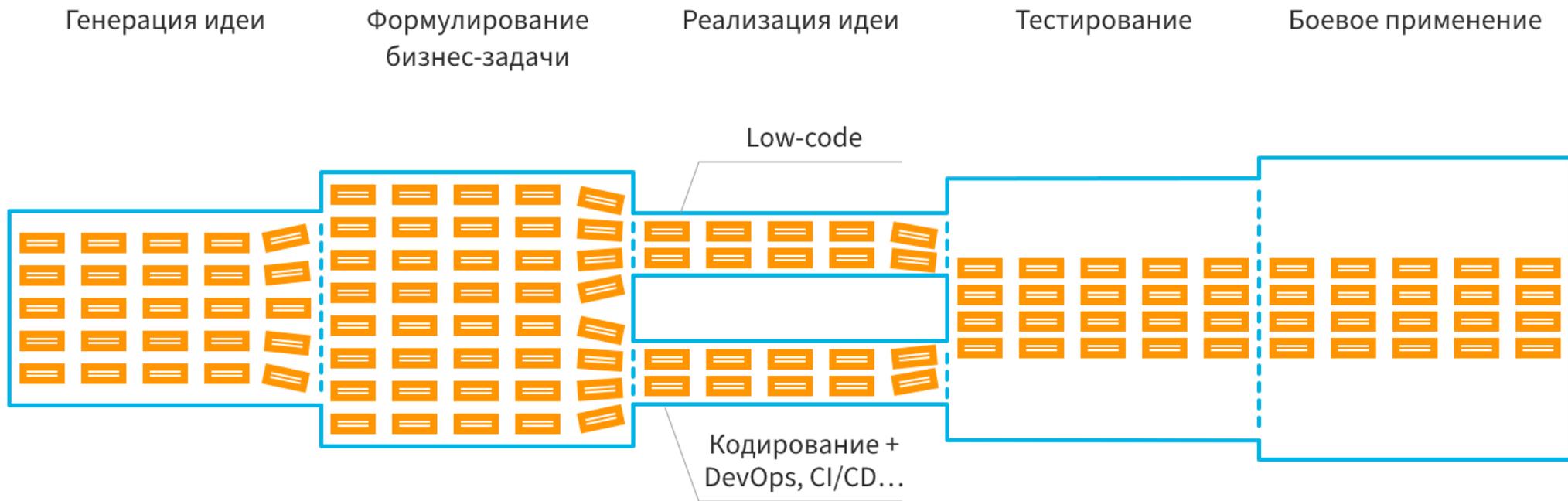
Доля работы в
корзину

КПД работы
разработчика – 2%

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

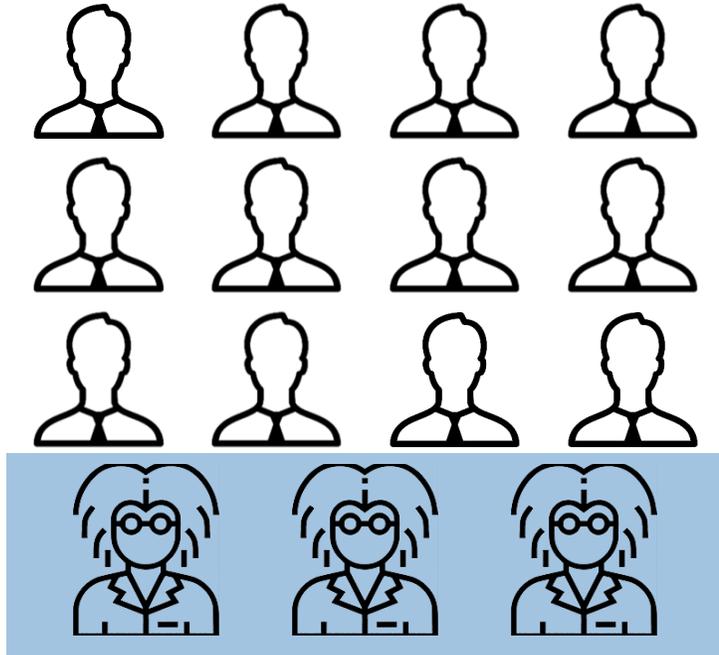
Дефицитное
время
разработчика

Доля
подготовительной
работы

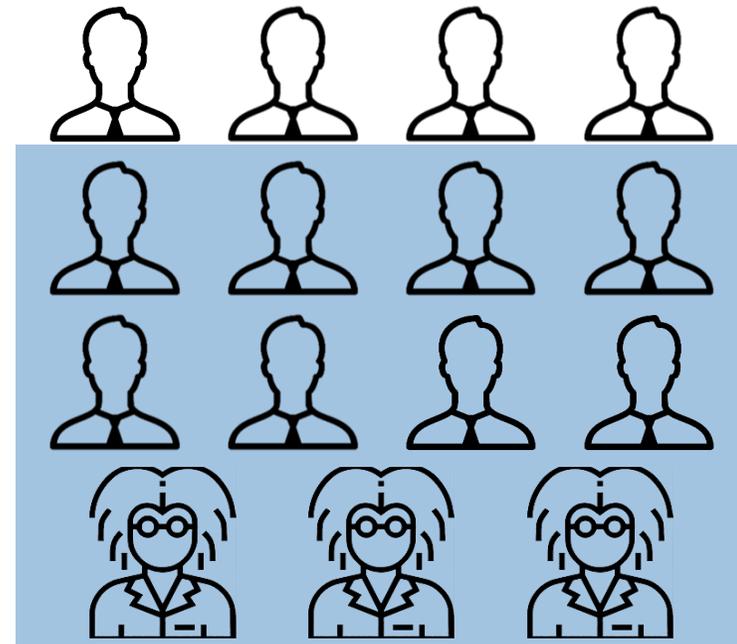


Low-code позволяет бизнес-экспертам часть работ выполнять самим, разгружая дефицитный ресурс и расширяя узкое место.

Проще найти/обучить специалистов



Разработчики



Разработчики +
Гражданские аналитики

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



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. ...

Low-code – не замена разработке, а лучшая инвестиция в цифровизацию:

1. Разработчики освобождаются от рутины, пользователи многое выполняют сами.
2. Гипотезы тестируют без программистов. Они концентрируются на апробированных идеях.
3. Программисты выполняют задачи, которые без них не решить.
4. Сложные блоки, реализованные при помощи кода, вызываются из low-code сценариев.

loginom.ru