



Low-code ECM

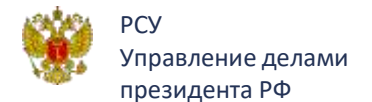
Новая парадигма автоматизации
документооборота



О компании ELMA



Более **2000** компаний
используют ELMA



Технологии ELMA365



Современные технологии
легкие и масштабируемые
решения от глобальных
ИТ-лидеров

 ELMA 365



Golang

ubuntu 



PostgreSQL



kubernetes

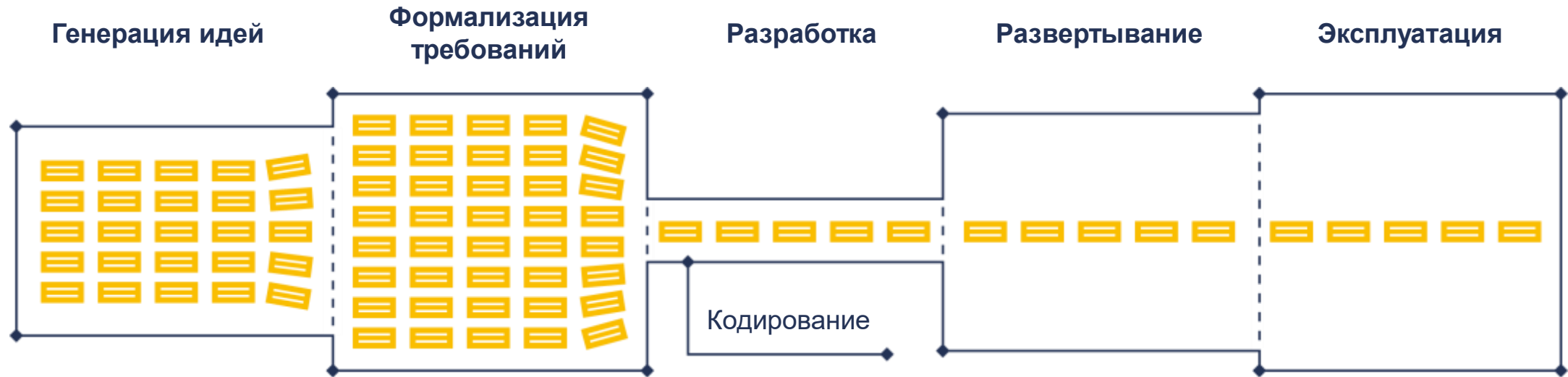
 **RabbitMQ**

 **ANGULAR**

 **Yandex Cloud**

Ценность Low-code для бизнеса





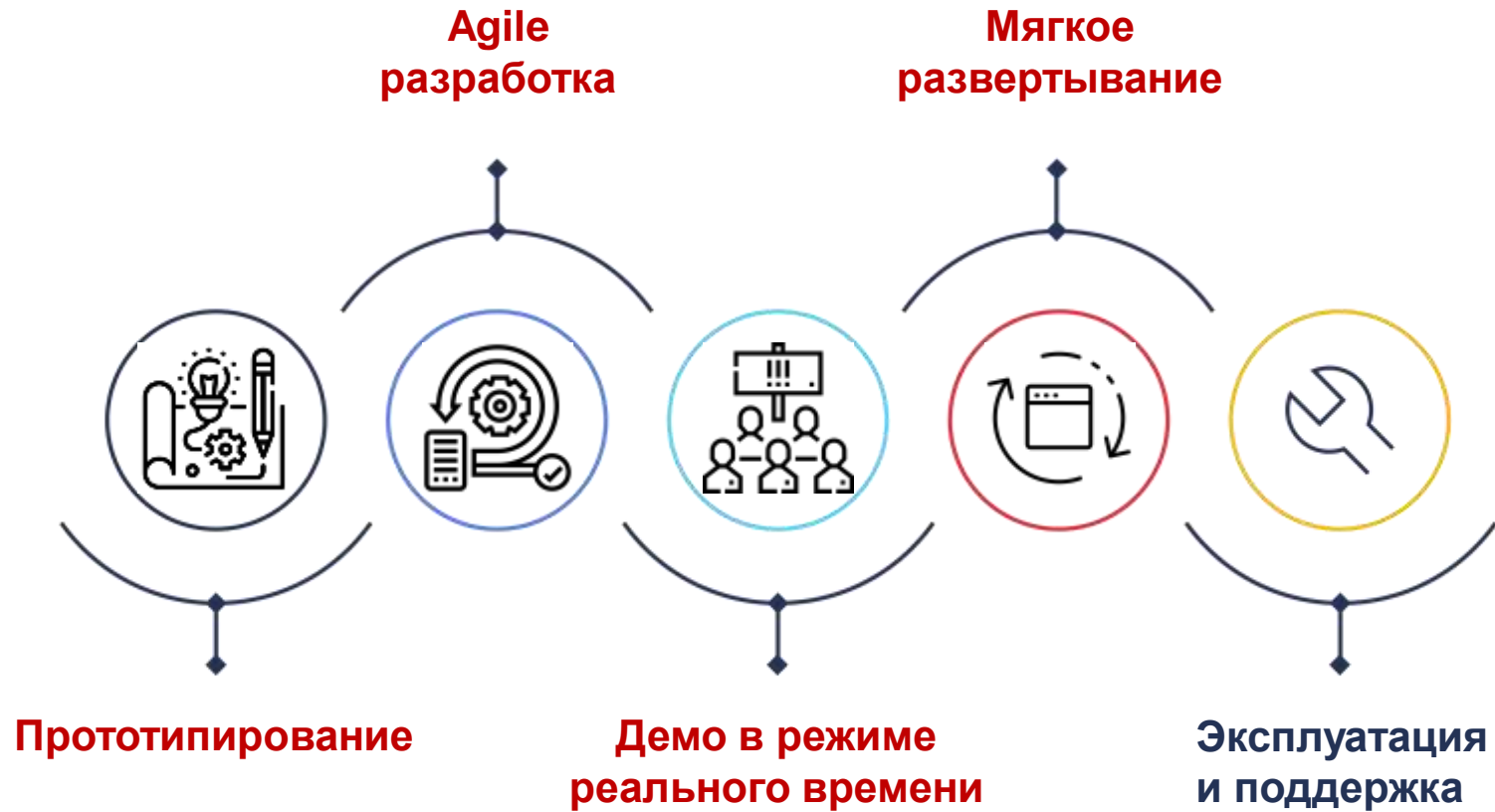
ТРАДИЦИОННАЯ РАЗРАБОТКА КОРПОРАТИВНЫХ ПРИЛОЖЕНИЙ



РАЗРАБОТКА LOW-CODE ПРИЛОЖЕНИЙ

Снижение бэклога ИТ создает новое качество разработки, теперь бизнес получает возможность тестировать свои гипотезы в режиме реального времени.

Трансформация процесса разработки



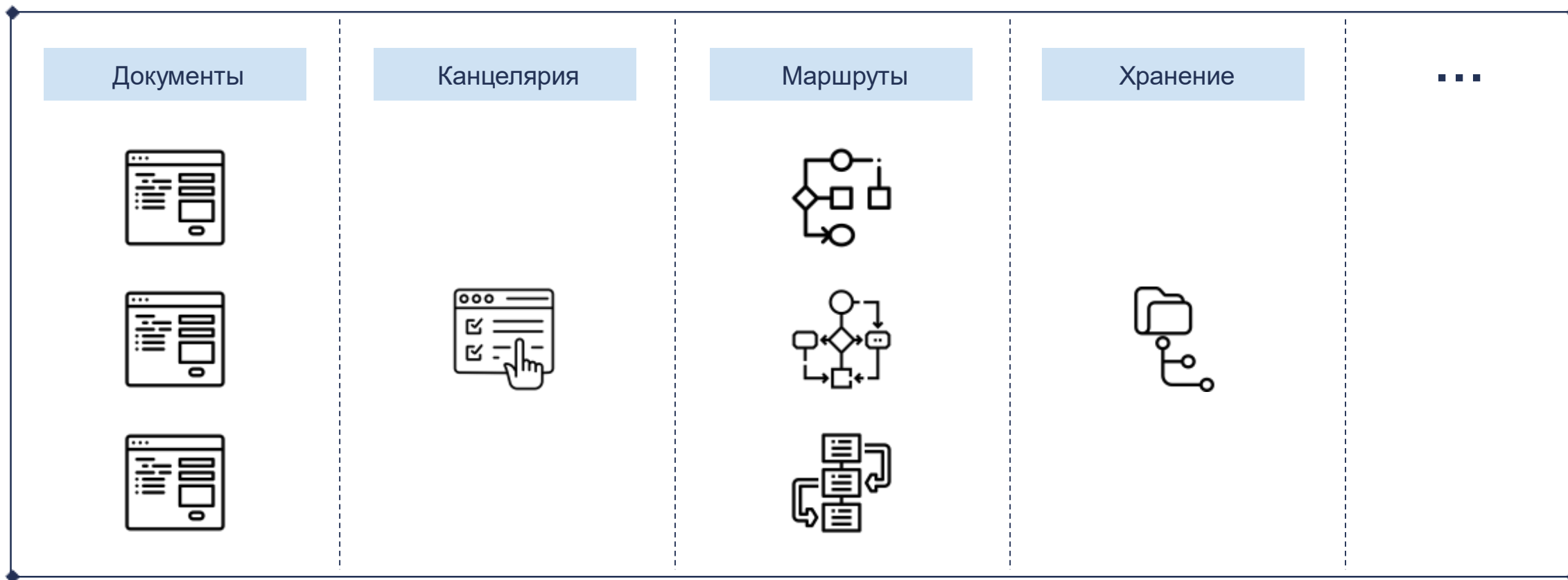
Low-code меняет процесс разработки,
упрощая каждый этап

Применение Low-code инструментария при автоматизации документооборота





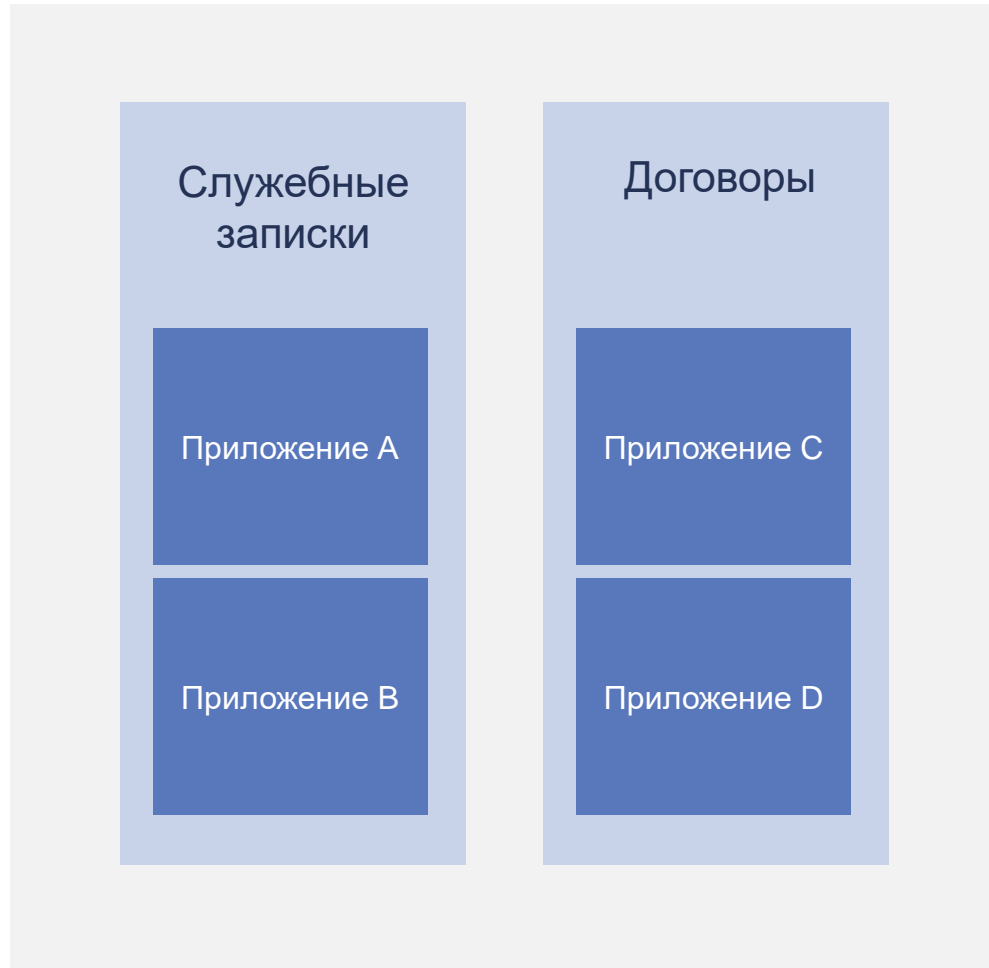
Традиционный СЭД



Монолитная система для работы со всеми типами документов



Изоляция приложений



Работа системы 24 на 7, не требуется перезагрузка при внесении каких-либо изменений



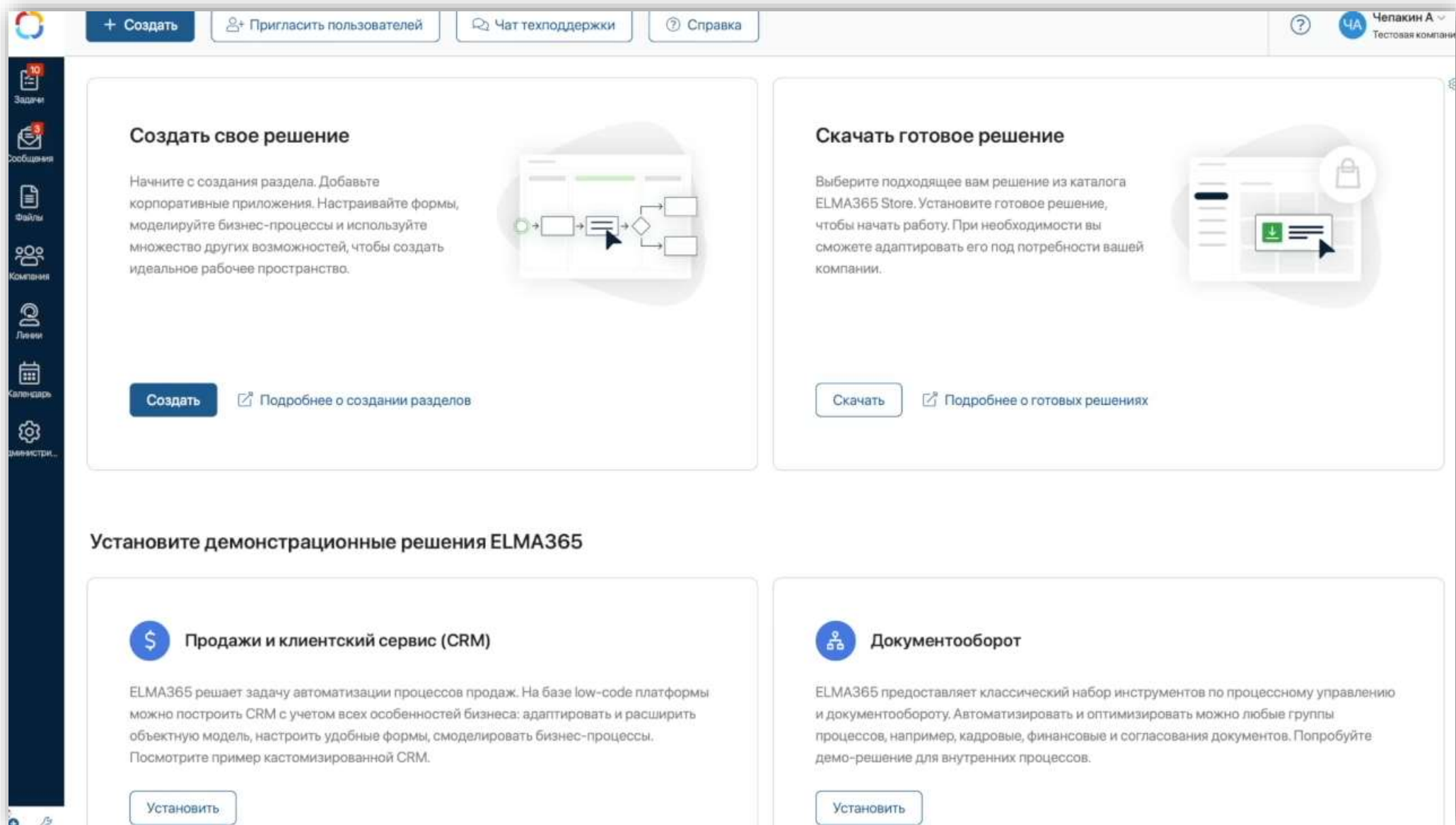
Готовые приложения могут быть легко экспортированы и импортированы



Приложения и разделы изолированы друг от друга. Изменения внутри приложения не касаются окружения



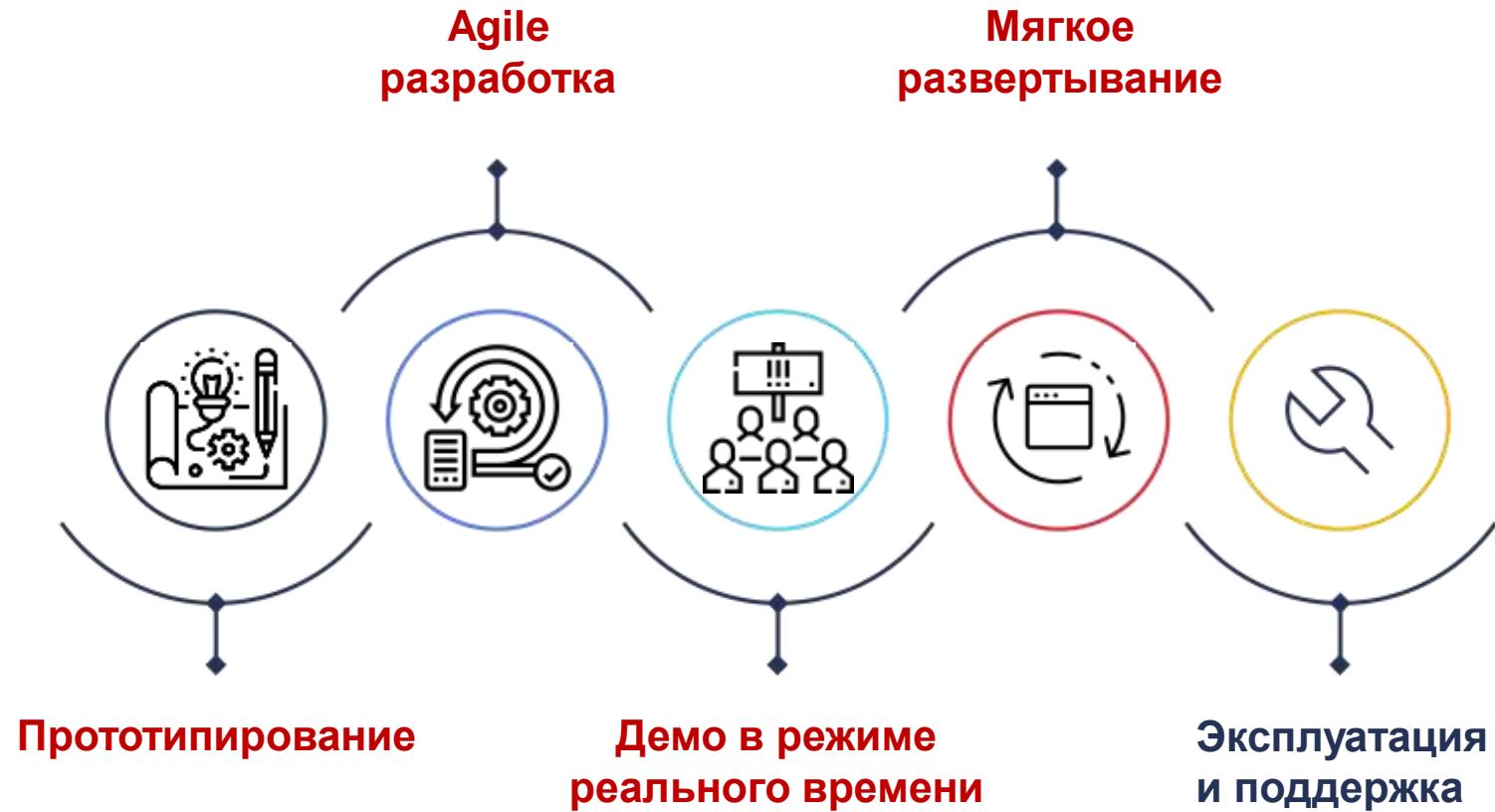
Изоляция приложений



The screenshot displays the ELMA365 user interface. At the top, there is a navigation bar with buttons for '+ Создать', 'Пригласить пользователей', 'Чат техподдержки', and 'Справка'. The user profile 'Чепакин А.' is visible in the top right corner. A vertical sidebar on the left contains icons for 'Задачи', 'Сообщения', 'Файлы', 'Компания', 'Линии', 'Календарь', and 'Администрирование'. The main content area is divided into four sections:

- Создать свое решение**: Encourages users to start by creating sections and adding corporate applications. It includes a flowchart icon and a 'Создать' button with a link to 'Подробнее о создании разделов'.
- Скачать готовое решение**: Promotes downloading solutions from the ELMA365 Store. It features a download icon and a 'Скачать' button with a link to 'Подробнее о готовых решениях'.
- Установите демонстрационные решения ELMA365**: This section contains two sub-cards:
 - Продажи и клиентский сервис (CRM)**: Describes how ELMA365 automates sales processes on a low-code platform. It includes an 'Установить' button.
 - Документооборот**: Highlights the platform's tools for process management and document exchange. It also includes an 'Установить' button.

Разработка Low-code приложения



Low-code меняет процесс разработки,
упрощая каждый этап



Классический подход

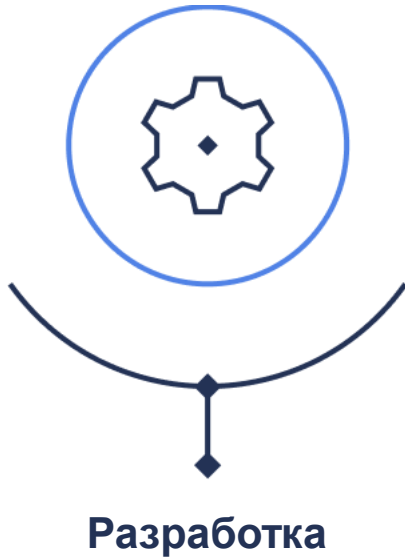
Продолжительная и трудоемкая формализация требований

Необходимость оставлять пробелы в требованиях с определением «по ходу проекта»

Серьезные требования к аналитикам, работающими над тех. заданием

Low-code

- ✓ Требования формируются быстро в лайт-формате
- ✓ Подготовка прототипа за несколько дней
- ✓ Обсуждение требования на прототипе облегчает понимание Бизнеса и ИТ



Классический подход

Высокие требования к команде разработки

Уточнения и пересогласования требований во время разработки увеличивают срок выхода приложения

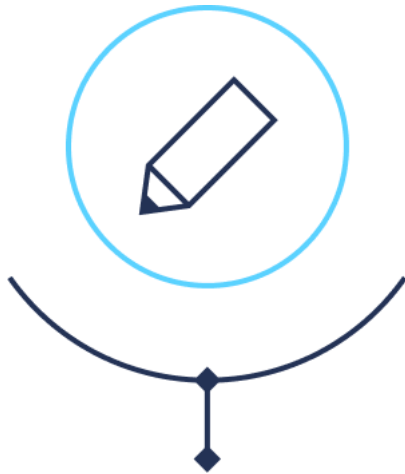
Ротация команды разработки увеличивает сроки и стоимость

Low-code

- ✓ Создание приложения без кодирования
- ✓ Снижение требований к составу команды
- ✓ Полное соответствие agile-подходу к разработке – движение спринтами с детализацией требований «по месту»

Прототипирование и разработка





Стабилизация

Классический подход

Устранение замечаний вероятно приводит к фундаментальной переработке приложения

Изменения требований и тех. задания по инициативе бизнеса в момент демонстрации финального результата

Сложность сдачи интерфейсной части и удобства использования, т.к. эти аспекты часто откладываются «на потом»

Low-code

- ✓ Быстрые демо-встречи с бизнес-заказчиками
- ✓ Внесение изменений и улучшений «на лету»
- ✓ Проработка и адаптация интерфейсов по запросу бизнеса

Стабилизация

Добавить Командировка ✕


Направление*

Цель командировки*

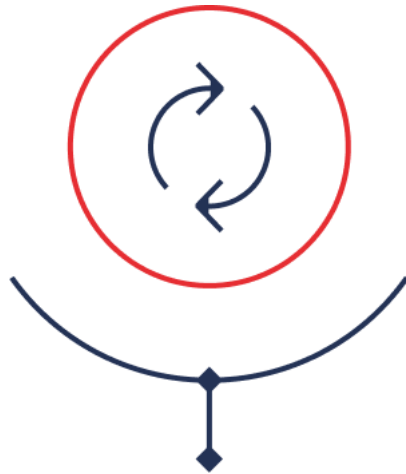
Дата начала командировки*

Дата окончания командировки*

Сумма под отчет Р

Инструкция 





Развертывание

Классический подход

Сложная процедура управления обновлениями и изменениями

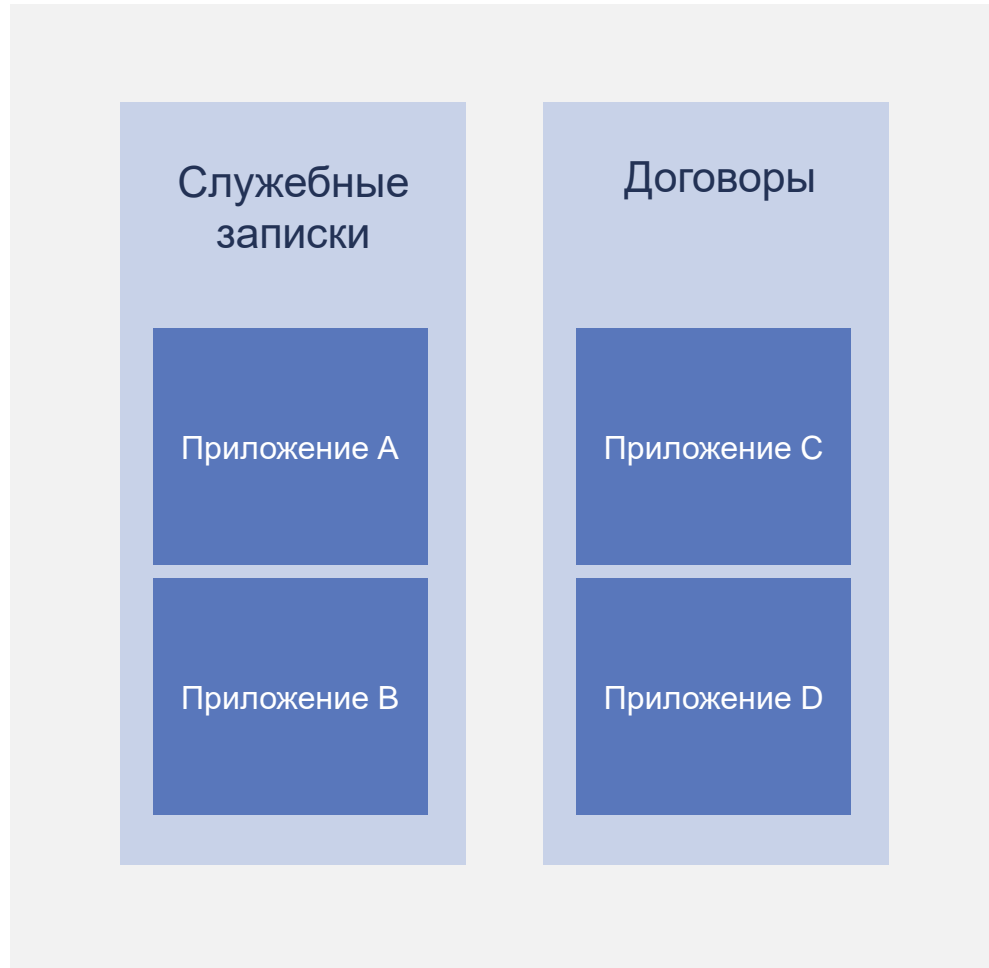
Запуск нового приложения порождает риски сопряжения с текущим ИТ-ландшафтом

Дополнительные усилия и меры для обеспечения непрерывности работы приложений

Low-code

- ✓ Архитектурная изоляция приложений
- ✓ Развертывание приложений без остановки системы
- ✓ Разделение сред разработки, тестирования и эксплуатации и быстрый перенос приложений между ними

Развертывание



Работа системы 24 на 7, не требуется перезагрузка при внесении каких-либо изменений



Готовые приложения могут быть легко экспортированы и импортированы



Приложения и разделы изолированы друг от друга. Изменения внутри приложения не касаются окружения





**Эксплуатация
и поддержка**

Классический подход

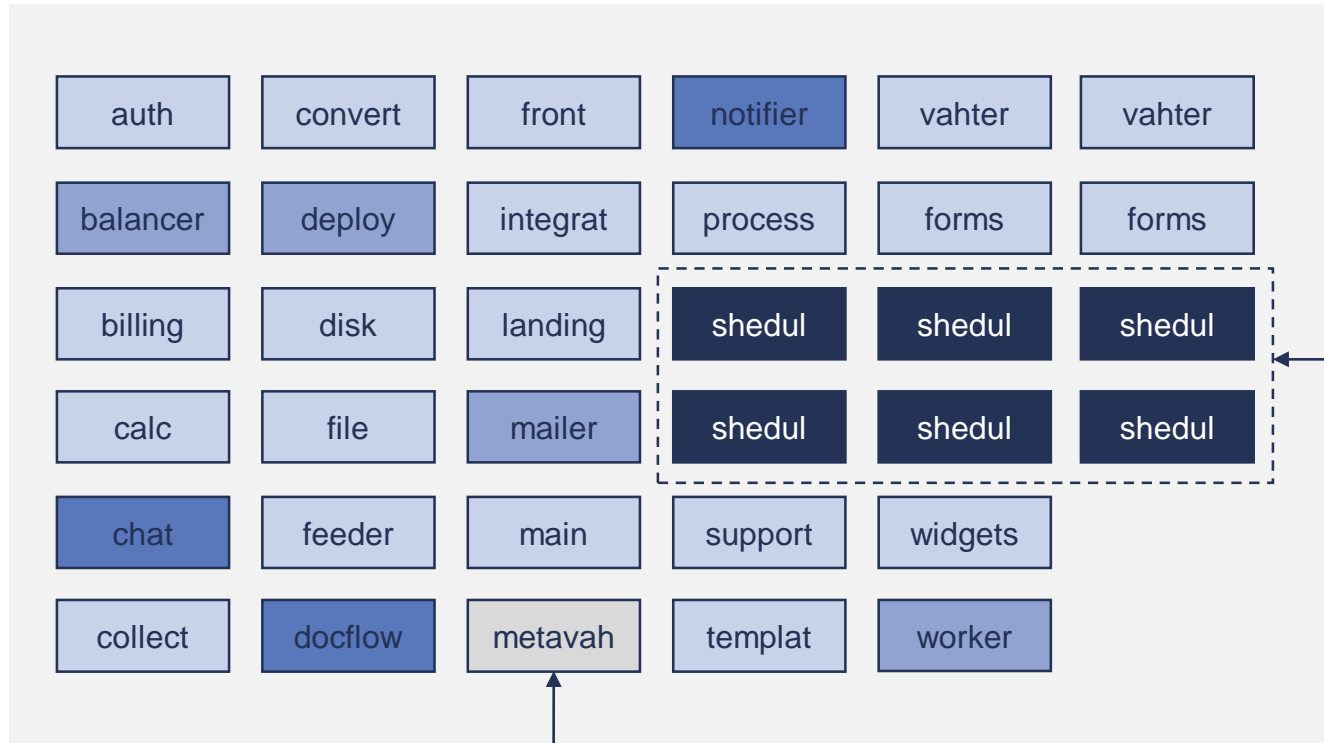
Дополнительные меры и усилия по обеспечению доступности приложений
Значительные расходы на масштабирование при увеличении нагрузки на приложение

Идеи бизнеса по развитию приложения формируют бэклог в ИТ

Low-code

- ✓ Оптимальное использование вычислительных ресурсов
- ✓ Быстрое масштабирование мощности при высоких нагрузках
- ✓ Неограниченные возможности разработки

Микросервисы, контейнеры, Kubernetes



Гибкое масштабирование
производительности
по высоко-нагруженным
микросервисам

Kubernetes. Платформа
управления нагрузкой
и масштабированием
вычислительных мощностей
на уровне контейнеров

Принципы: Легкость входа / Многослойность



ELMA Low-code BPM обеспечивает легкость входа – это значит, что приложение может быть создано быстро и просто. При этом, это приложение может усложняться до необходимой степени с помощью DevOps-инструментария. ELMA Low-code BPM позволяет создавать сколько угодно сложные приложения.



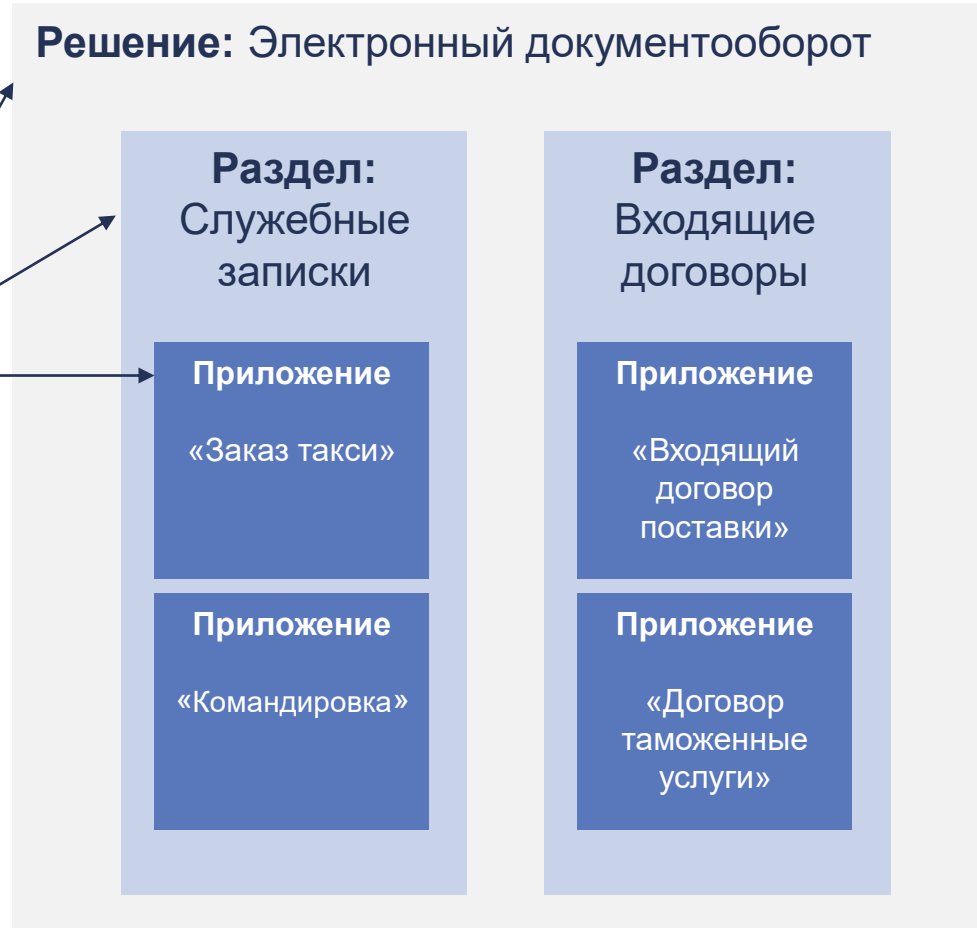
Low-code features



Уровни абстракции и архитектурная изоляция

Уровни абстракции:

- Решение
- Раздел
- Приложение



- ✓ Обеспечен асинхронный режим разработки и обновления на любом уровне абстракции
- ✓ Принцип независимости от окружения
- ✓ Разработка отдельных приложений, разделов или решений с разными уровнями глубины

Расширяемая объектная модель

Рекламные кампании / Настройки формы

Расширенный режим Сохранить X

Наименование РК

Дата/время

Код* data_vremya

Подсказка

Устанавливать текущие дату и время

Время опционально

Поиск и сортировка по полю

Регион

Макет

Товары

Приложение

Уровни промо

Строка

Число

Выбор «да/нет»

Дата/время

Категория

Деньги

Номер телефона

Электронная почта

Изображение

Файлы

Ф.И.О.

Ссылка

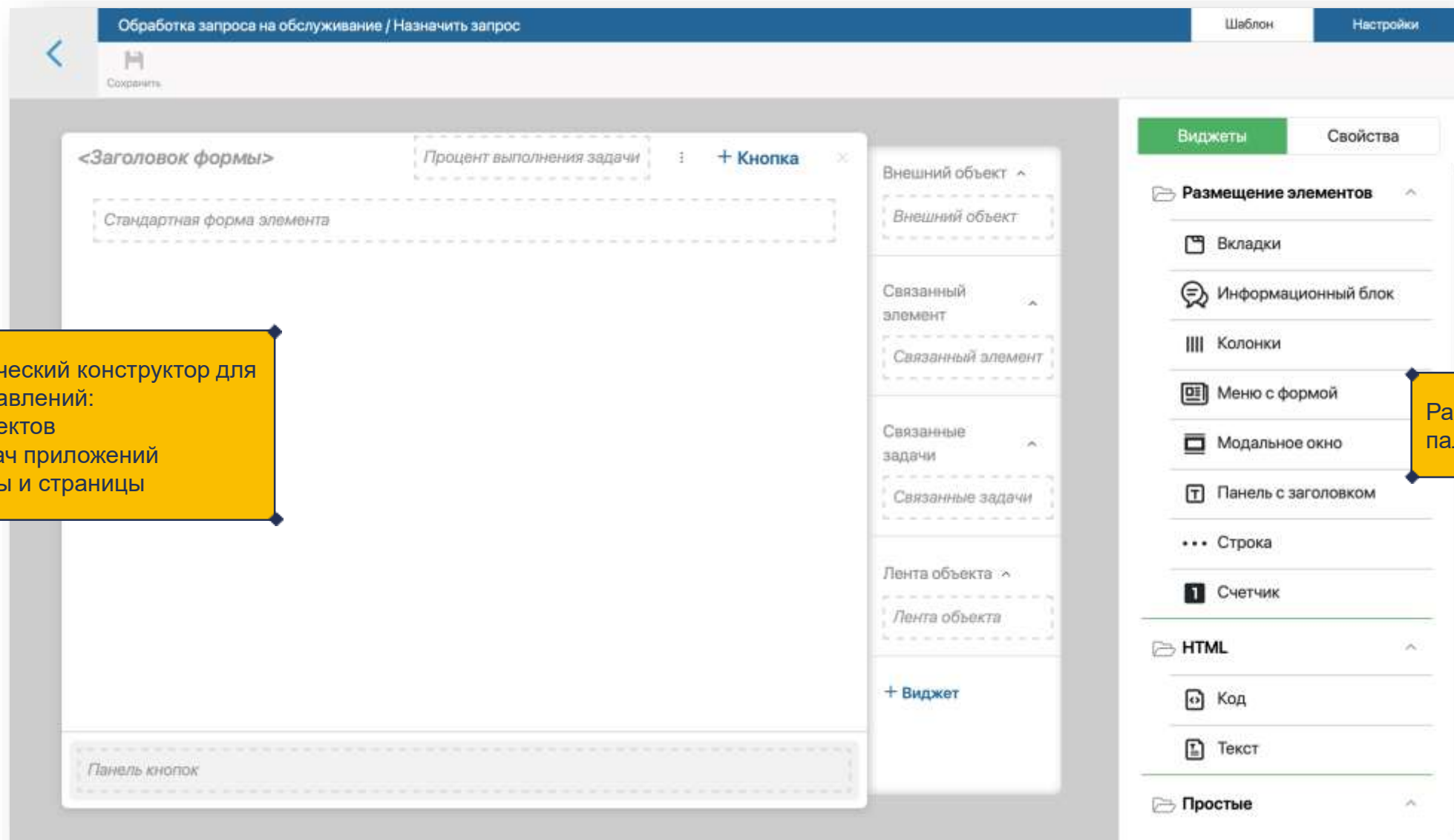
Таблица

Пользователи

Приложение

Создание объекта автоматизации внутри Приложения. Определяем произвольный набор данных для объекта.

Графический конструктор

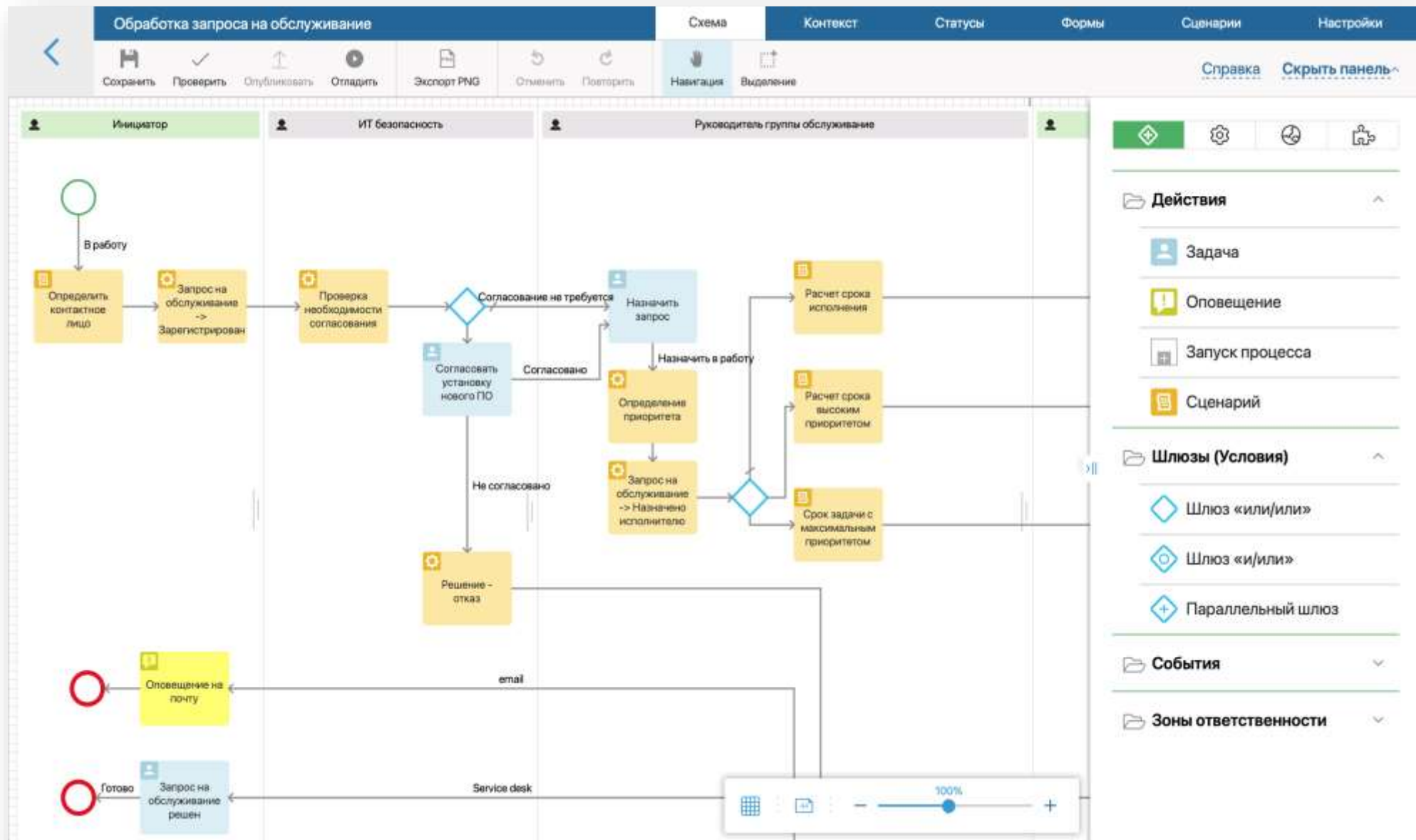


Единый графический конструктор для разных представлений:

1. Формы объектов
2. Формы задач приложений
3. Интерфейсы и страницы

Расширяемая палитра виджетов

BRM для управления бизнес-логикой



Генерация API для приложений «на лету»

The screenshot displays the 'Service desk' interface. On the left is a navigation menu with items like 'Главная страница', 'Запрос на обслуживан...', 'Инцидент', 'CMDBtst', 'Сервис', 'CMDB', 'SLA', 'Отчет', and 'Отчет_демо'. The main area is titled 'Запрос на обслуживание / API'. It contains a description: 'Работа с элементами Приложения осуществляется через Web API. Доступные функции перечислены ниже. Полезные ссылки: Документация по Web API'. Below this are tabs: 'Список элементов', 'Получить элемент', 'Создать элемент', 'Изменить элемент', and 'Изменить статус'. The 'Создать элемент' tab is active, showing details for 'Создание элемента Приложения'. The method is 'post' and the address is 'https://o2si34iy66uv6.elma365.ru/pub/v1/app/service_desk_demo/zapros_na_obs_luzhivanie/create'. There is a link to 'Открыть документацию по методу create'. Below the details are buttons for 'JSON пакет' and 'Проверить'. A JSON object is displayed in a code block:

```
{
  "context": {
    "kratkoe_opisanie": "example",
    "opisanie": "example",
    "servis": [
      "00000000-0000-0000-0000-000000000000"
    ],
    "reshenie_predostavit_mne": true,
    "contact_person": [
      "00000000-0000-0000-0000-000000000000"
    ],
    "sposob_obratnoi_svyazi": [
      {
        "code": "code",
        "name": "name"
      }
    ],
    "prioritet": [
      {
        "code": "code",
        "name": "name"
      }
    ]
  }
}
```

Созданное Приложение автоматически становится доступным для запуска извне веб-сервисом

Мобильное приложение

Приложение автоматически доступно на мобильных устройствах

Тип транспорта

Автомобиль

Вид транспорта

Новый

Тип кредита

Залоговый кредит

Автомобиль

Марка

BMW

AUDI
BMW
Subaru
Skoda

2890000

Калькулятор

Транспорт

Торговая точка

Тверская

Тип транспорта

Автомобиль

Вид транспорта

Новый

Тип кредита

Залоговый кредит

Автомобиль

Марка

Subaru

Модель

Legacy

Стоимость ТС, руб *

2300000

ПВ, руб *

690000

Срок кредита, мес *

14

Да Нет **Страховые и сервисные услуги**

Тип продукта *

КАСКО

Компания *

Ренессанс

Тариф *

Оптимум

Срок, лет *

2

Стоимость, руб

460000

Включено в кредит

Да Нет

Предварительные расчёты

Платёж в месяц: 185,063.00 руб.

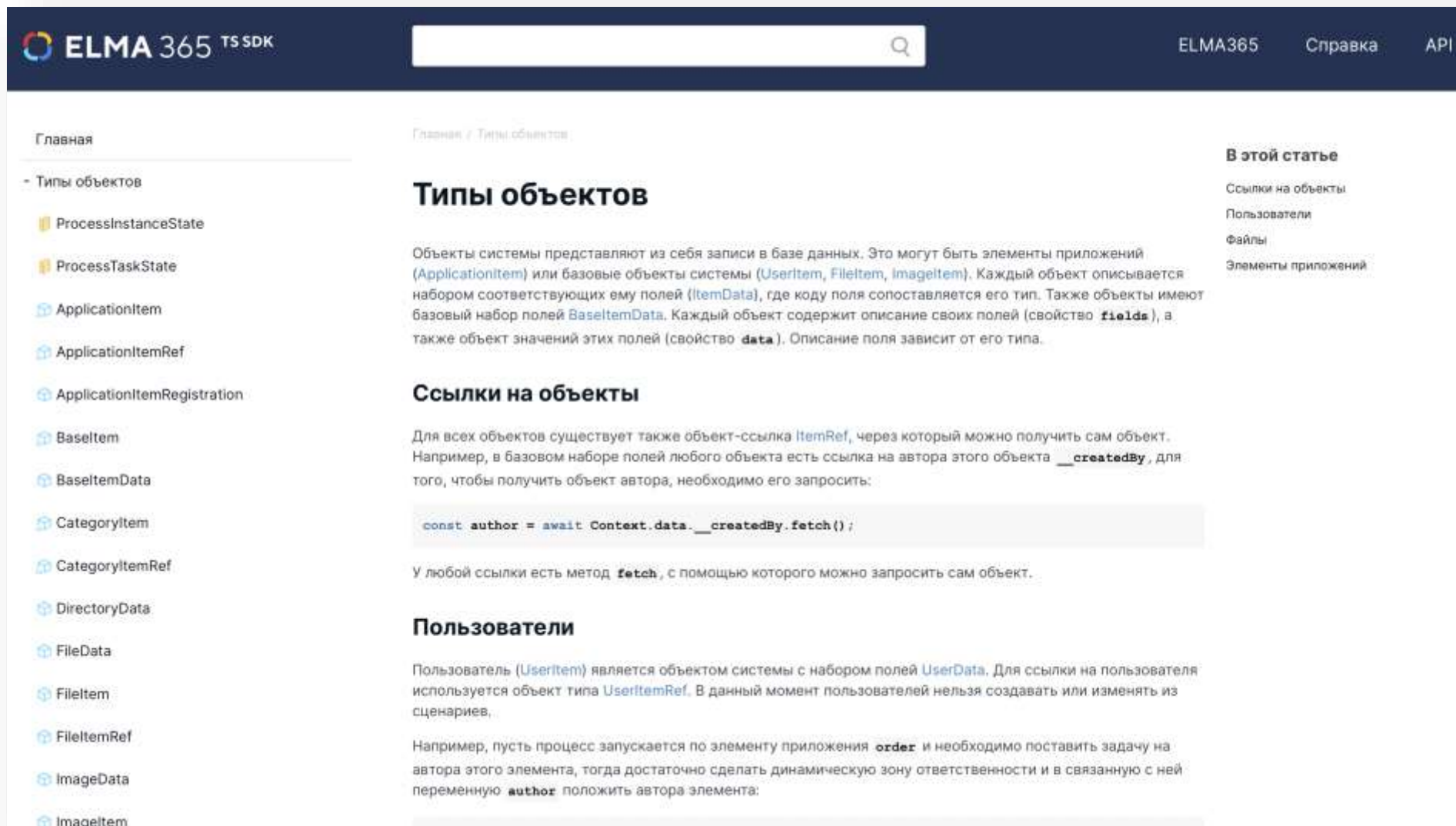
Общая стоимость: 2,590,896.00 руб.

Расчитать график платежа

График платежа

Да Нет Вычислять график на лету

Дата платежа	Сумма платежа	Остаток
01/26/2020	185063	2405823
02/26/2020	185063	2220760
03/26/2020	185063	2035697
04/26/2020	185063	1850634
05/26/2020	185063	1753475
06/26/2020	185063	1654823
07/26/2020	185063	1551132
08/26/2020	185063	1451986
09/26/2020	185063	1356389
10/26/2020	185063	1251053
11/26/2020	185063	1150634



The screenshot shows the ELMA 365 TS SDK documentation page for "Types of Objects". The page has a dark blue header with the ELMA 365 logo, a search bar, and navigation links for "ELMA365", "Справка", and "API". A left sidebar lists various object types, with "Types of Objects" selected. The main content area features a breadcrumb "Главная / Типы объектов", a title "Типы объектов", and a descriptive paragraph. Below this is a section "Ссылки на объекты" with a code example and a paragraph. The final section is "Пользователи" with a paragraph. A right sidebar titled "В этой статье" lists links to "Ссылки на объекты", "Пользователи", "Файлы", and "Элементы приложений".

Главная / Типы объектов

Типы объектов

Объекты системы представляют из себя записи в базе данных. Это могут быть элементы приложений (*ApplicationItem*) или базовые объекты системы (*UserItem*, *FileItem*, *ImageItem*). Каждый объект описывается набором соответствующих ему полей (*ItemData*), где коду поля сопоставляется его тип. Также объекты имеют базовый набор полей *BaseltemData*. Каждый объект содержит описание своих полей (свойство **fields**), а также объект значений этих полей (свойство **data**). Описание поля зависит от его типа.

Ссылки на объекты

Для всех объектов существует также объект-ссылка *ItemRef*, через который можно получить сам объект. Например, в базовом наборе полей любого объекта есть ссылка на автора этого объекта **__createdBy**, для того, чтобы получить объект автора, необходимо его запросить:

```
const author = await Context.data.__createdBy.fetch();
```

У любой ссылки есть метод **fetch**, с помощью которого можно запросить сам объект.

Пользователи

Пользователь (*UserItem*) является объектом системы с набором полей *UserData*. Для ссылки на пользователя используется объект типа *UserItemRef*. В данный момент пользователей нельзя создавать или изменять из сценариев.

Например, пусть процесс запускается по элементу приложения **order** и необходимо поставить задачу на автора этого элемента, тогда достаточно сделать динамическую зону ответственности и в связанную с ней переменную **author** положить автора элемента:

В этой статье

- Ссылки на объекты
- Пользователи
- Файлы
- Элементы приложений

Вопросы

