



# Искусственный интеллект

Что такое хорошо,  
а что такое плохо?

Дегтярев Евгений, СТО  
Компания УМАРТА



Отсканируй, чтобы  
перейти на сайт



# Решаемые нами с ИИ задачи



## Рутинные операции

Эти задачи повторяются регулярно, требуют быстрого выполнения и часто связаны с поддержкой процессов, коммуникацией или обработкой данных.

Для их решения чаще используются Языковые модели (LLM) и NLP- библиотеки, модели для генерации изображений, звука и видео

- ✓ Генерация текстов
- ✓ Генерация кода
- ✓ Проверка кода
- ✓ Суммаризация
- ✓ Дизайн и маркетинг (генерация изображений, видео, звука)



## Стратегические (инновационные) задачи

Эти задачи требуют глубокого анализа, прогнозирования или создания новых решений. Они часто уникальны и направлены на улучшение продуктов, процессов или бизнес-стратегии.

Для их решения чаще используется машинное обучение (ML) и глубокое обучение (DL)

- ✓ Прогнозирование нагрузки и аномалий (нагрузка на оборудование)
- ✓ Рекомендательные системы (персонализация контента)
- ✓ Кластеризация (группировка данных, обнаружение аномалий(анти-фрод))
- ✓ Оптимизация (маршрутов, запасов)

# Вредные советы

## 1 Совет

Выдайте молодому поколению ИИ помощников: Пусть и не начинают сами думать. Зато ваша ценность всегда будет велика.

Для не развитого критического мышления и отсутствия опыта по прорабатываемой теме любой ответ ИИ может казаться правильным, но любая модель способна галлюцинировать.

## 2 Совет

Обязательно используйте ИИ для всего, даже если вы не понимаете что там сделает ИИ. Внедряйте ИИ в каждый процесс, даже если традиционным способом задача решается дешевле!

ИИ это такой же инструмент как ложка или молоток. Если начать есть суп вилкой, то это получится не сильно эффективно, не смотря на то, что ею можно быстрее нанести травм собеседнику :)

## 3 Совет

Полностью исключайте человека из цепочки принятия решений! Что он может вообще, кожаный мешок?

ИИ может ошибаться, особенно в нестандартных ситуациях. Отсутствие человеческого контроля может привести к катастрофическим последствиям (например, в медицине, финансах или управлении транспортом).

## 4 Совет

Используйте ИИ чтобы заменить людей, а не помочь им!

ИИ – это инструмент для помощи людям, а не их замены. Полная замена сотрудников может привести к потере ценного опыта, снижению качества работы и социальным проблемам.



# Пример LLM – проверка кода

Модель подключена к GIT репозиторию кода проекта. Старший разработчик проекта после получения кода от члена команды на включения его в основную ветку репозитория выполняет запрос типа:

Ты – опытный сотрудник информационной безопасности, специализирующийся на анализе кода и выявлении уязвимостей. Твоя задача – провести анализ Java-кода из подключенного Git-репозитория на предмет потенциальных уязвимостей, таких как SQL-инъекции, XSS, утечки данных, неправильная обработка исключений, слабая криптография и другие риски. После анализа необходимо составить модель угроз, описать найденные уязвимости и дать рекомендации разработчикам по их устранению.

Пожалуйста, выполни следующие шаги:

- ✓ Проанализируй Java-код из репозитория и выяви потенциальные уязвимости.
- ✓ Составь модель угроз, классифицируя риски по уровню критичности (высокий, средний, низкий).
- ✓ Предложи конкретные рекомендации по устранению каждой уязвимости.
- ✓ Если возможно, предложи улучшения для повышения общей безопасности кода.

Пожалуйста, предоставь подробный отчет в следующем формате:

## 1. Найденные уязвимости:

- Описание уязвимости.
- Уровень критичности.
- Рекомендации по устранению.

## 2. Модель угроз:

- Таблица с классификацией угроз и их уровнем критичности.

## 3. Общие рекомендации:

- Советы по улучшению безопасности кода.

Проанализируй код из репозитория: [git-link]

1. SQL-инъекции: Наличие чувствительных параметров в URL-адресе JDBC:

```
main.datasource.url =  
jdbc:postgresql://${CRD_POSTGRES_HOST:localhost}/${CRD_POSTGRES_DB:people_crd}
```

Риск: Возможность SQL-инъекций через подстановку данных в базу.

2. Утечка паролей:

```
main.datasource.password = ${CRD_POSTGRES_PASSWORD:root}
```

Уровень риска: Высокий

3. Слабая аутентификация:

```
ui.login.defaultUsername = admin  
ui.login.defaultPassword = admin
```

Модель угроз:

Уязвимость	Описание	Уровень риска
SQL инъекции	Возможность выполнения опасных SQL-запросов через параметры конфигурации	Высокий
Информационная утечка паролей	Подбор пароля через прямое доступа к конфигу	Высокий

Рекомендации:

1. Для SQL-инъекций:

```
main.datasource.url = jdbc:postgresql://${CRD_POSTGRES_HOST}/secure_database?  
password=encrypted_password
```

Использовать параметры конечной точки аутентификации, хранить пароли в безопасном месте.

2. Для утечки паролей:

- Мигрировать на более современную систему аутентификации
- Регулярно менять пароли по истечении срока действия

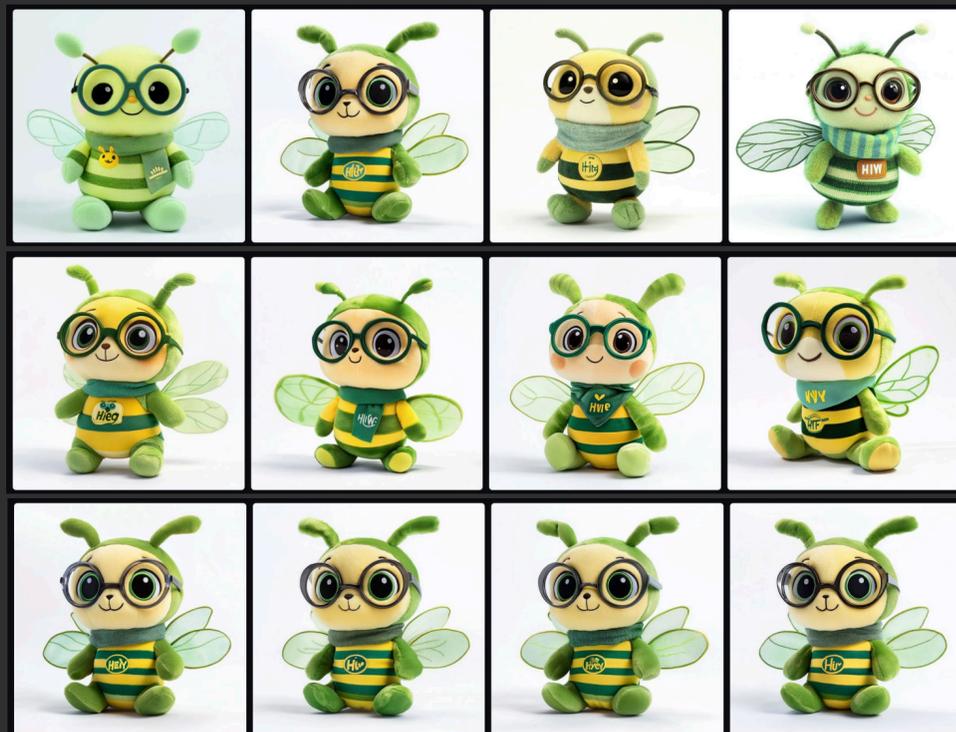
3. Общие улучшения:

```
# Использование централизованной системы конфигурации  
security.encryption.key = ${ENCRYPTION_KEY}
```

Использовать безопасный способ хранения конфиденциальных данных.

# Пример генерация изображений – ускорение работы дизайнера

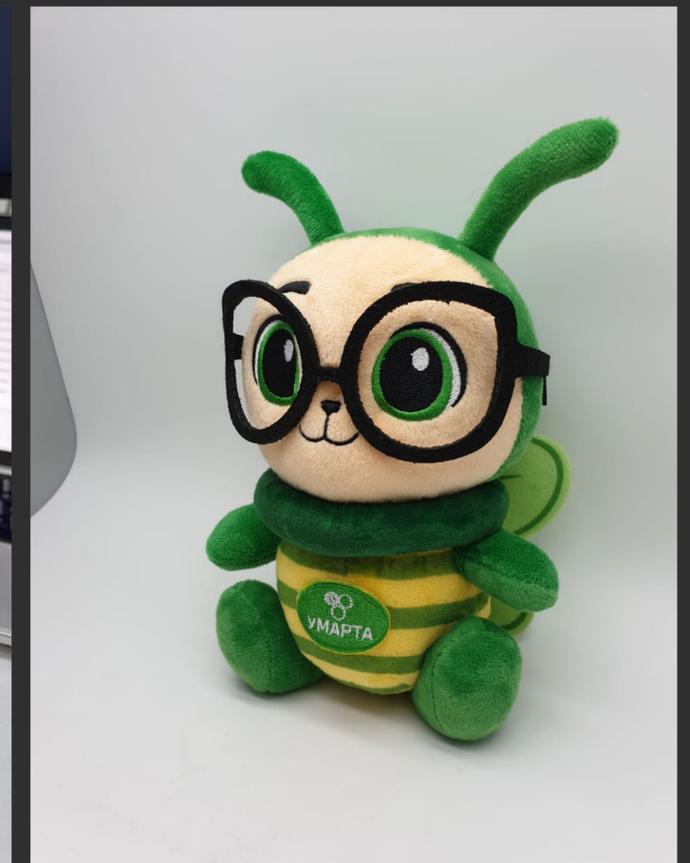
Midjourney



Дизайнер, с учетом ограничений производства



Финальное изделие



# Пример LLM coder – подготовка прототипа

Создай одностраничный прототип простой CRM-системы на HTML и JavaScript. Прототип должен включать следующие элементы: Заголовок страницы: "CRM-система".

## Вкладка 1:

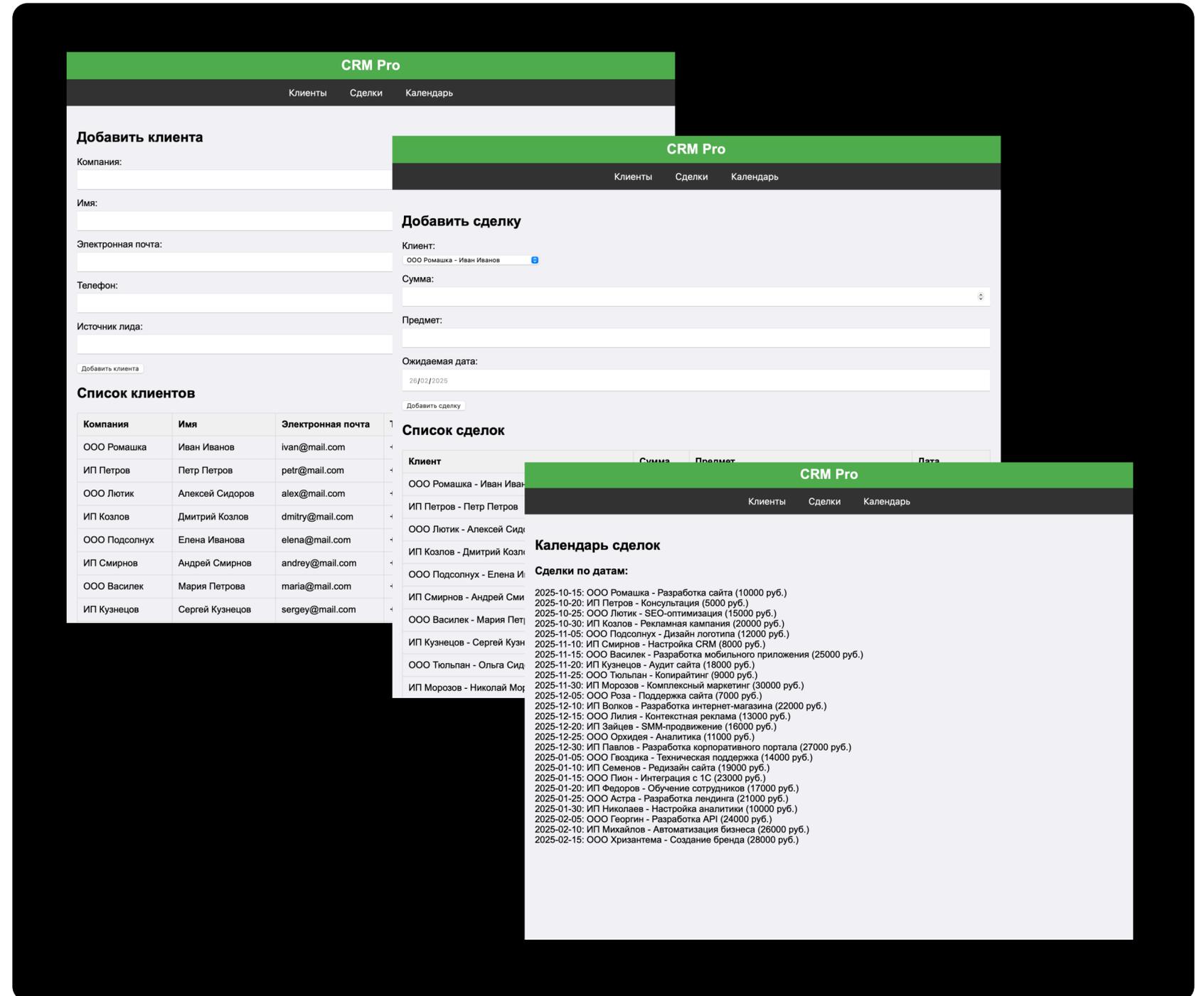
Форма добавления клиента: Поля для ввода: Компания, Имя, Электронная почта, Телефон, Источник лида. Кнопка "Добавить клиента". Таблица для отображения добавленных клиентов: Колонки: Компания, Имя, Электронная почта, Телефон, Источник лида, Действия. В колонке "Действия" должны быть кнопки "Удалить" и "изменить" для каждого клиента. Логика на JavaScript: При нажатии на кнопку "Добавить клиента" данные из формы должны добавляться в таблицу. При нажатии на кнопку "Удалить" соответствующий клиент должен удаляться из таблицы. При нажатии на кнопку "изменить" должна быть возможность отредактировать данные.

## Вкладка 2:

Сделка. Выбор клиента из вкладки 1, возможность указать сумму, предмет и ожидаемую дату.

## Вкладка 3:

Календарное представление сделок. Данные должны сохраняться только в памяти (localStorage не требуется). В системе должны быть сразу тестовые данные. Код должен быть компактным и уместиться на одной странице. Используй стилизацию, возьми какой-то любой логотип и придумай название продукта.



# LLM нужно использовать с осторожностью

Вариация на тему старой менеджерской шутки про 9 женщин, которые должны родить ребёнка за месяц. Следовательно 108 женщин должны родить 12 детей.

При этом опытный руководитель задаёт вопрос, а сколько из них были беременными на момент начала проекта

Молодым сотрудникам была предложена простая математическая задача – посчитать вероятность того, что ровно 1 из 108 случайно выбранных 108 женщин окажется беременной\*.

Естественно считать никто не стал. Был спрошен Deep Seek. А затем и Chat GPT. Оба привели правильное решение, но ответы были разными. Как вы думаете, кто дал правильный ответ?

- Дано:
- Общее количество женщин: 75 млн
  - Количество беременных женщин в год: 3 млн
  - Вероятность беременности для одной женщины:  $p = \frac{3\,000\,000}{75\,000\,000} = 0.04$
  - Количество женщин в выборке:  $n = 108$
  - Количество беременных в выборке:  $k = 1$

Формула биномиального распределения:

$$P(k) = C(n, k) \cdot p^k \cdot (1 - p)^{n-k}$$

где  $C(n, k)$  — число сочетаний из  $n$  по  $k$ .

Вычисляем:

$$C(108, 1) = 108$$

$$P(1) = 108 \cdot 0.04^1 \cdot 0.96^{107}$$

Итоговая вероятность:

$$P(1) \approx 108 \cdot 0.04 \cdot 0.96^{107} \approx 0.073$$

Ответ:

0.073



Данные:

- Всего женщин: 75 млн
- Беременных в год: 3 млн
- Вероятность беременности:

$$p = \frac{3\,000\,000}{75\,000\,000} = 0.04$$

- Выбираем  $n = 108$  женщин
- Хотим найти вероятность ровно 1 беременной:

Формула биномиального распределения:

$$P(k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Подставляем:

$$P(1) = \binom{108}{1} (0.04)^1 (0.96)^{107}$$

Я посчитаю это.

Вероятность того, что среди 108 случайно выбранных женщин будет ровно одна беременная, составляет ≈5.48%. [-]

\* Условия задачи специально упрощены, дабы не травмировать детские мозги теорией вероятностей слишком сильно

# LLM нужно использовать с осторожностью

## Проблемы с результатом

1. В Java-коде не было импортов, пришлось добавить в prompt.
2. Была выбрана устаревшая версия Spring Boot. Пришлось добавить в prompt.
3. В настройках безопасности не было кода обработки CORS и CSRF. Пришлось добавить в prompt.
4. В сущностях не были сгенерированы геттеры и сеттеры. Пришлось добавить в prompt.
5. Ошибки в коде: вызовы несуществующих методов, вызовы API других версий, и т.д.
6. В веб-приложении неверная структура папок.
7. Несовместимые версии библиотек. Пришлось добавить в prompt.
8. Совершенно без css. Пришлось добавить в prompt.
9. Добавленный css ужасен. Пришлось добавить Material в prompt.
10. **НО В ИТОГЕ ОНО РАБОТАЕТ!**

Employees					
<a href="#">CREATE</a>					
First Name	Last Name	Middle Name	Date of Birth	Start Date	Actions
Евгений	Дегтярев	Нориевич	Invalid Date	01.07.2023	<a href="#">EDIT</a> <a href="#">REMOVE</a>

You

Напиши пожалуйста web-приложение для хранения данных о сотрудниках в компании.

Технические требования:

1. Язык программирования для backend – Java. Фреймворк – Spring Boot
2. Язык программирования для frontend – JavaScript. Фреймворк – react.
3. СУБД – PostgreSQL.
4. Должна быть интеграция с корпоративным Idap сервером (реализация freeipa) для использования логинов и паролей пользователей из Idap.
5. Допустим один внутренний пользователь admin, которого должно быть можно заблокировать после подключения к Idap и он может быть доступен только при недоступности домен контроллера.

Функциональные требования:

В базе данных должны храниться данные о сотрудниках: фамилия, имя, отчетство, дата рождения, дата выхода на работу, дата изменения оклада, оклад, дата встречи с руководителем, заметки по встрече от руководителя. Изменений окладов в истории может быть много, также может быть много встреч и заметок по ним

Приложение должно иметь две роли:

администратор, который видит все данные и может добавлять пользователей

пользователь, который видит только назначенных ему сотрудников, заметки по встречам с ними и может сам добавить заметку к этим сотрудникам

## Add Employee

First Name *	Last Name *
Евгений	Дегтярев
Middle Name *	Date of Birth *
Нориевич	ДД.ММ.ГГГГ
Start Date *	
01.07.2023	
<a href="#">SUBMIT</a> <a href="#">CANCEL</a>	

# Контакты



Технический директор  
Евгений Дегтярев

Телефон  
+7 927 762-31-78

E-mail  
[ed@umarta.dev](mailto:ed@umarta.dev)

Telegram  
[@degtyarjov](https://t.me/@degtyarjov)

