

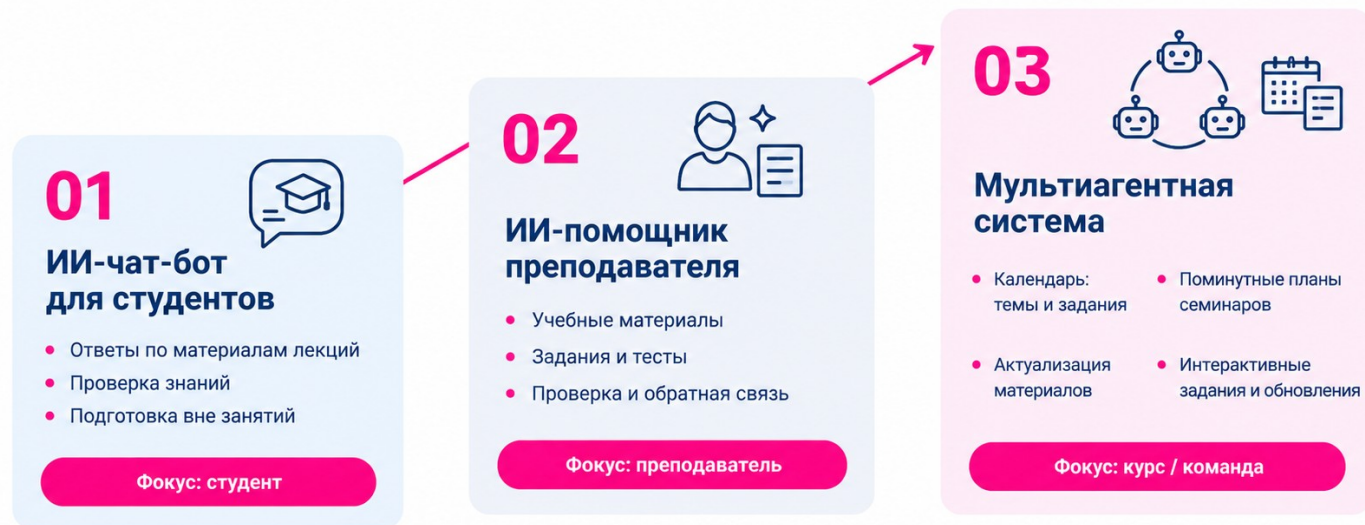
Кейсы применения ИИ в курсе «Цифровые медиакоммуникации»

от чат-бота для студентов до команды ИИ-сотрудников

Дейнекин Т.В., 2026

Эволюция применения ИИ в курсе «Цифровые медиакоммуникации»

--



Единая логика: ИИ берёт рутину — преподаватель сохраняет контроль

Результаты опроса: востребованность ИИ-инструментов

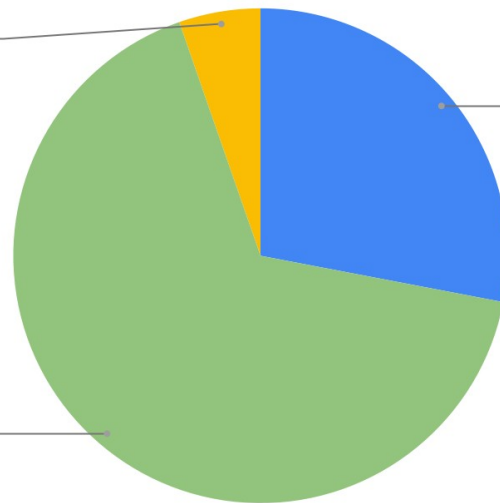
— — —
Большинство студентов, записавшихся на курс, уже регулярно используют ИИ-инструменты в своем обучении*

Использовали ли вы ранее ИИ-инструменты для обучения?

Нет, никогда
5,3%

Да, несколько раз
28,0%

Да, регулярно
66,7%



* Результаты собственного исследования. Опрос студентов ЭФ МГУ 4 курса, 75 студентов, ноябрь-декабрь 2025.

С каким трудностями сталкиваются студенты

— — —
При использовании сторонних сервисов ИИ студенты сталкиваются с целым рядом проблем: ошибочные ответы, общие формулировки, непонимание контекста, отсутствие персонализации.*



* Опрос студентов ЭФ МГУ 4 курса ноябрь-декабрь 2025.

Этап 1.

ИИ-чат-бот для студентов:
«Светозар Светозарович»

Концепция ИИ-помощника “Светозар Светозарович”

— — —

С 2023 года, с момента начала роста прикладного использования генеративных сетей, в преподавании дисциплин “Искусственный интеллект в маркетинге” (магистратура и МВА) и “Цифровые медиакоммуникации” (бакалавриат) последовательно развивалась концепция универсального ИИ-помощника для студентов и слушателей



В 2023 году с помощью ИИ сервисов Яндекса было придумано имя и образ помощника - Светозар Светозарович (“ученье - свет”).

2023 год

— — —



Изначально ИИ использовался для создания анимированных роликов, в которых Светозар Светозарович подводил итоги отдельных лекций, а также делал объявления.

2024 год

— — —

Однако персонажу не хватало интерактивности, поэтому в 2024 году была создана первая версия ИИ-консультанта (Telegram bot) на базе RAG-архитектуры и моделей Gigachat API.

```
from langchain.schema import HumanMessage, SystemMessage
from langchain.chat_models.gigachat import GigaChat
from chromadb.config import Settings
from langchain.vectorstores import Chroma
from langchain_community.embeddings import GigaChatEmbeddings
from langchain.chains import RetrievalQA
```

```
messages = [
    SystemMessage(
        content="Тебя зовут Светозар Светозарович. Твоя функция – консультировать и помогать ос
    )
]
messagesall = {}

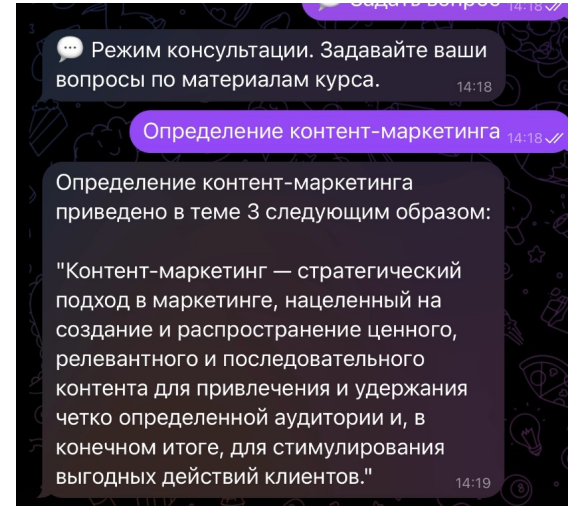
@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.send_message(message.chat.id, 'Привет! Меня зовут Светозар Светозарович! Я консуль

@bot.message_handler(func=lambda message: True)
def echo_message(message):
    if message.chat.id in list(messagesall.keys()):
        messagesall[message.chat.id].append(HumanMessage(content=message.text))
    else:
        messagesall[message.chat.id] = messages
        messagesall[message.chat.id].append(HumanMessage(content=message.text))
```

1. Понятие контент-маркетинга
Определение (Content Marketing Institute)
Контент-маркетинг – стратегический подход в маркетинге, направленный на создание и распространение ценного, релевантного и последовательного контента для привлечения и удержания четко определенной аудитории и, в конечном счете, для стимулирования выгодных действий клиентов.

↑
Текст из лекции

Ответ бота



Изначальная образовательная задача состояла в том, чтобы предоставить студентам возможность получать ответы на вопросы относительно учебных материалов не только во время лекций и семинаров, но и в интерактивном формате при взаимодействии с чат-ботом.

Основная техническая задача - добиться от чат-бота ответов “один в один” в соответствии с текстами лекций.

Обе задачи были решены, это стало основой для последующей эволюции бота.

2025 год

— — —

Развитие вайб-кодинга позволило систематизировать разработки и вывести решение на качественно новый уровень.

Новые возможности

- выбор темы;
- проверка знаний в режиме теста, либо открытых вопросов;
- просмотра статистики и работы над ошибками.

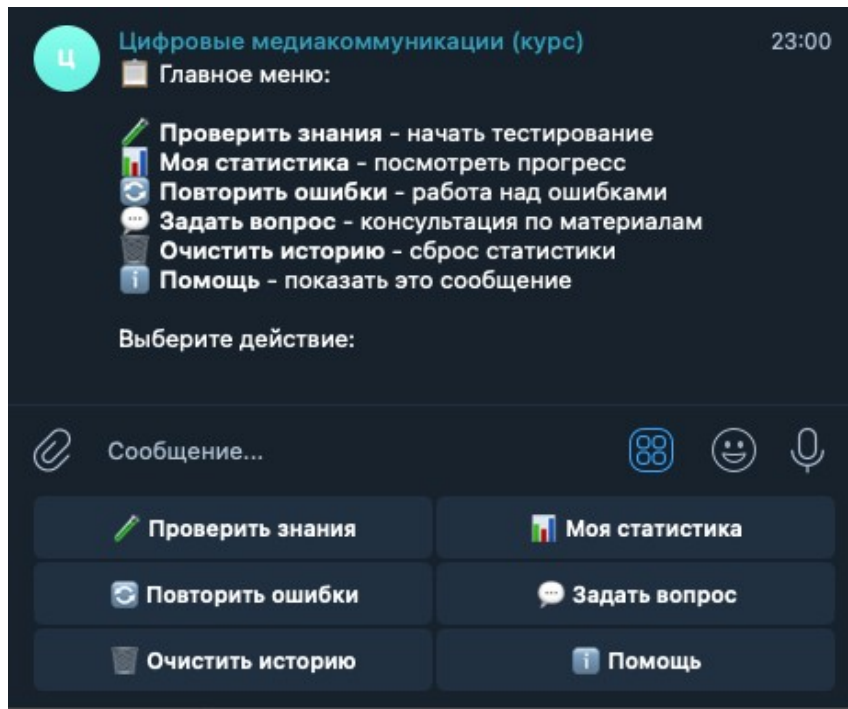
Универсальность решения: управляющему модулю достаточно указать “путь” к файлам новой дисциплины, чтобы без изменения программного кода запустить нового бота.

Функционал экспериментального чат-бота

— — —
Проверить знания - бот предлагает проверить знания в режиме теста или в режиме открыты вопросов.

Работа над ошибками - бот предлагает повторно ответить на вопросы, в которых были допущены ошибки.

Задать вопрос - бот отвечает на вопросы студентов по материалам курса.



Функционал: проверить знания - база вопросов

— — —

На основе текстов лекций посредством запросов по API к модели **GigaChat Pro** было составлено 150 открытых вопросов и 150 вопросов для теста. Все вопросы прошли валидацию преподавателя, после нее было оставлено 123 и 116 вопросов соответственно. Некоторые вопросы были дополнительно отредактированы.

ПРОВЕРКА СГЕНЕРИРОВАННЫХ ВОПРОСОВ

Лекция: Тема 1. Основные теории массовых коммуникаций

 Открытые вопросы: 26

 Тестовые вопросы: 24

Лекция: Тема 2. Цифровые медиа и виды коммуникаций в цифровой среде

 Открытые вопросы: 21

 Тестовые вопросы: 22

Лекция: Тема 3. Цифровой контент

 Открытые вопросы: 27

 Тестовые вопросы: 24

Лекция: Тема 4. Планирование рекламных кампаний в цифровых медиа

 Открытые вопросы: 26

 Тестовые вопросы: 24

Лекция: Тема 5. Социальные медиа

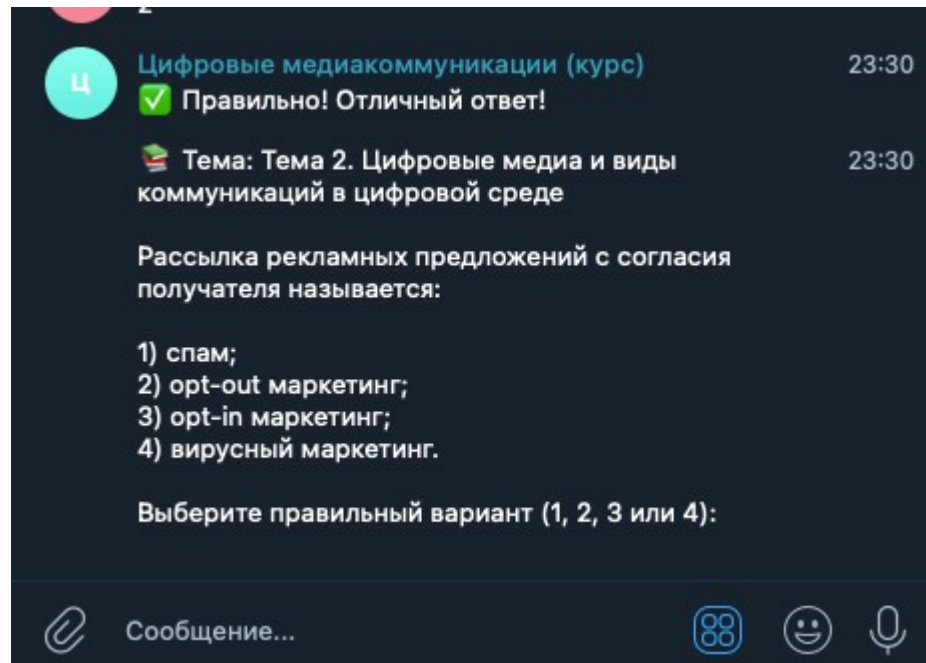
 Открытые вопросы: 23

 Тестовые вопросы: 22

Функционал: проверка знаний в режиме теста

В режиме проверки знаний в форме тестирования бот задает вопросы, студент выбирает правильный ответ. Бот сразу сообщает результат и ведет статистику.

Правильные ответы изначально были помечены GigaChat Pro при генерации базы вопросов, поэтому в этом режиме дополнительные запросы к Gigachat отсутствуют.



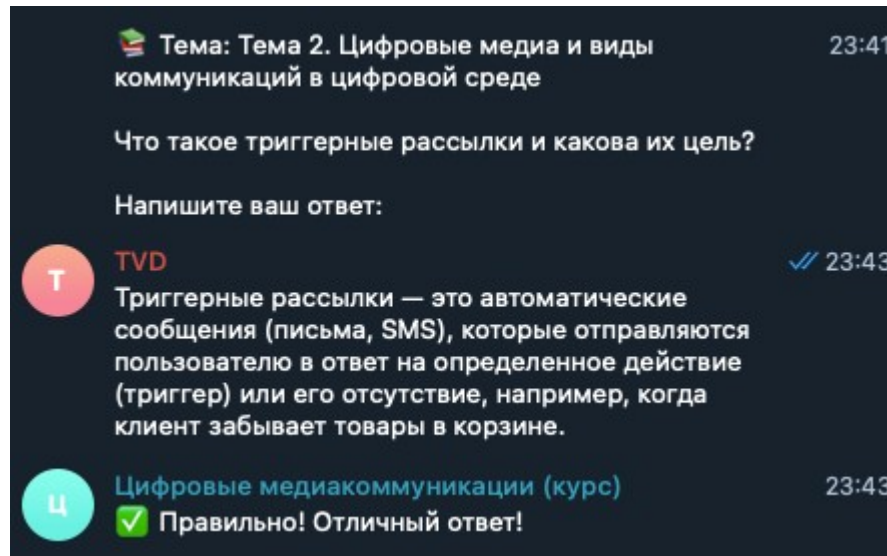
Функционал: проверка знаний в режиме открытых вопросов

— — —

В режиме открытых вопросов студенту требуется самостоятельно написать ответ.

*Посредством **дополнительного запроса к Gigachat** ответ студента сравнивается с эталонным.*

Эталонные ответы изначально были сгенерированы GigaChat Pro на основе лекций.



Функционал: консультации (задать вопрос)

— — —

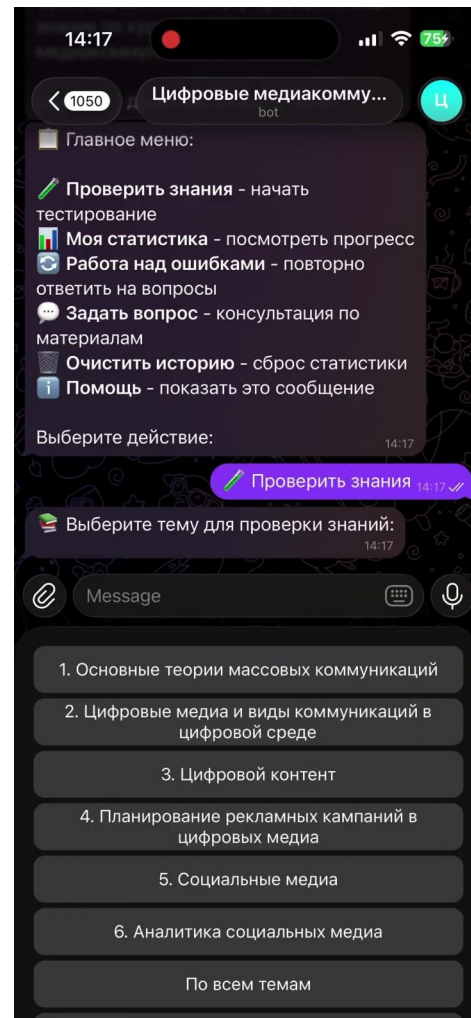
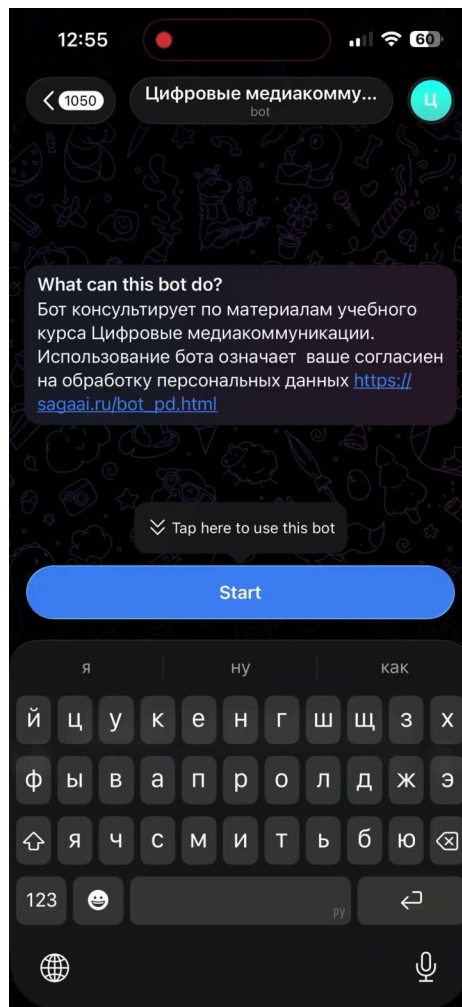
В этом режиме студент может задать дополнительные вопросы по материалам лекций, чтобы уточнить понятия, классификации, формулы.

Технологии: *Gigachat API + RAG-база по текстам лекций на основе Gigachat Embeddings*



Иллюстрация работы бота.

* в настоящее время бот не работает в круглосуточном режиме, так как для его работы требуются иностранные IP-адреса



Результаты эксперимента: баллы групп и статистика

— — —

Использование MVP чат-бота для подготовки к промежуточному тесту

	Количество студентов	Средний балл за тест (шкала 0-8)
Использовали	56	6,71
Не использовали	19	6,33*

Вывод: Хотя студенты, которые пользовались ботом, получили более высокие баллы, однако статистически значимой разницы между этими группами обнаружено не было (p -значение $> 0,05$)**.

* Один студент не указал свою оценку за тест, поэтому среднее по 18 студентам.

** Расчеты произведены ИИ-агентом “Аналитик” в сервисе Gigachat

Результаты опроса: NPS и готовность использовать

В качестве готовности рекомендовать бота после использования был рассчитан индекс NPS для тех, кто пользовался ботом*

NPS = 62,49 п.п.

Доверительный интервал:

от 48,21 до 75,0 п.п..**

Вывод: высокий уровень готовности рекомендовать чат-бота.

Балл	Количество	Доля
4	1	1,79%
7	4	7,14%
8	15	26,79%
9	11	19,64%
10	25	44,64%
Итого	56	100,00%

Готовность использовать бота дальше - 83,93% (H2 подтверждена, $p < 0,05$).

NPS = 62,49 п.п.; доверительный интервал 48,21-75,0 п.п. (H3 подтверждена).

Вывод: высокая лояльность и готовность рекомендовать.

Общие выводы по кейсу

— — —

- 67% студентов курса уже регулярно используют ИИ для учебы.
- 84% студентов высоко оценили чат-бота в качестве инструмента для подготовки к сводному тесту и готовы его использовать в дальнейшем.
- NPS = 62,49 п.п. свидетельствует о том, что у студентов сформировалась первоначальная лояльность к этому инструменту, и они готовы рекомендовать чат-бота своим друзьям и знакомым.

Этап 2.

Простые ИИ-помощники

ИИ-помощник “Второе мнение”

— — —

Цель - сократить время оценивания и повысить полноту обратной связи для больших групп студентов.

Задачи:

- проводить предварительную оценку ответа студента на основе критериев, которые были указаны в задании
- формировать черновой текст обратной связи с указанием сильных сторон, ошибок и рекомендаций по доработке
- повысить объективность оценки

Техническая реализация:

- Python, вайб-кодинг
- Deepseek API (модели chat/reasoner)
- Локальный интерфейс веб-приложения с библиотекой Streamlit

Преподаватель добавляет задание

Добавить новое задание

Название задания*

Описание задания*

Максимальная оценка*

10

- +

Учебно-методический материал ?

Укажите учебные материалы, которые изучали студенты (лекции, учебники, статьи и т.д.)

Степень строгости оценки ?

3

Выбранный уровень: 3: Умеренно

Добавить задание

Загрузка работы

— — —

Преподаватель выбирает проверяемое задание и загружает в систему работу студента (поддерживаются все основные форматы файлов).

Программа извлекает из файла текст ответа, который вместе с описанием задания и контекстом отправляется по API в LLM.

Проверка работ

Выберите задание для проверки:

CRISP-DM (макс. 10 баллов)

> [Посмотреть описание задания](#)

Текст работы студента

Выберите способ ввода:

- ☐ Текстовое поле
- ☒ Загрузка файла

Загрузите файл



Drag and drop file here

Limit 200MB per file • TXT, PDF, DOCX, DOC, PPTX, XLSX, XLS

[Browse files](#)



Задание 1 CRISP-DM (1).docx 17.2KB



Текст успешно извлечен!

> [Посмотреть извлеченный текст](#)

[Отправить на проверку](#)

Результат

— — —

ИИ проводит оценку работы и предоставляет черновой вариант ее обоснования.

Преподаватель принимает или не принимает доводы ИИ после самостоятельного ознакомления с работой.

При необходимости редактирует черновик отзыва и отправляет студенту (с пометкой, что при составлении использовался ИИ).

Результат проверки

Рекомендуемая оценка

5/10

Процент выполнения

50.0%

Требуется доработки

Обоснование для преподавателя

Работа студента не полностью соответствует условиям задания. Ответы предоставлены в формате CSV (хотя требовался PDF/Word), и их объем значительно меньше требуемого (2-3 абзаца на вопрос). Содержание ответов поверхностно: студент перечисляет основные пункты, но не раскрывает глубину методологии CRISP-DM, как это описано в учебном материале. Например, отсутствует упоминание итеративности процесса, конкретных подфаз (таких как оценка ситуации, анализ рисков, создание глоссария), артефактов (отчеты о качестве данных, план проекта) и взаимосвязей между этапами. Ответы носят общий характер, схожий с краткими определениями из онлайн-источников, что может указывать на использование ИИ или недостаточное изучение материала. Критерии оценки: соответствие — низкое (формат и объем не соблюдены), качество — удовлетворительное (ответы в целом корректны, но не детализированы), глубина проработки — недостаточная, оригинальность — отсутствует.

Обоснование для студента

Вы предоставили ответы на вопросы, которые в целом отражают ключевые идеи CRISP-DM. Однако работа не соответствует требованиям по объему (необходимо 2-3 абзаца на каждый вопрос) и глубине проработки. В ответах не хватает деталей: например, не раскрыта итеративность процесса, специфика подзадач в каждой фазе (как анализ рисков или создание отчетов), а также связь с практическими примерами. Рекомендуется более внимательно изучить учебный материал, уделяя внимание конкретным шагам и документам каждого этапа. Для улучшения результатов старайтесь раскрывать каждый вопрос подробно, используя терминологию из методички. При составлении отзыва о работе использовался ИИ-агент Второе мнение.

Общие наблюдения*

— — —

- Качество оценивания выше при строго определенных критериях, указанных в самом задании, но иногда ИИ их воспринимает слишком буквально.
- Замечает “детали”, которые при беглом самостоятельном просмотре решения не всегда заметны.
- Периодически занижает оценки для самостоятельных работ (без LLM) по стилю изложения, полноте, логике.
- Степень “лояльности” в оценке зависит от промпта.
- Позволяет обеспечить оперативную обратную связь.

* Наблюдения носят качественный характер, количественно еще не проверялись.

SagaAI assistant (2025)

— — —

Цель - сократить время для создания ИИ-помощников и локализовать их.

Задачи:

- Быстрое создание помощников для типовых образовательных задач (и не только).
- Локализация всех материалов на своем компьютере, а не в облачных решениях.
- Возможность выбора под каждого агента своей LLM

Техническая реализация:

- Python, вайб-кодинг
- Локальный интерфейс веб-приложения с библиотекой Streamlit

📄 SagaAI Ассистент

✎ Новый запрос

📁 История

🧩 Навыки

⚙️ Общие настройки

🌐 Язык интерфейса

Русский



🎯 Навыки

+ Создать навык

Протокол встречи по ВКР

DeepSeek > deepseek-reasoner | T=0.5 | 16.03.2026



Методист по тексту заданий

DeepSeek > deepseek-reasoner | T=0.7 | 15.03.2026



Корректор

DeepSeek > deepseek-reasoner | T=0.2 | 27.02.2026

Промпт-инженер

✎ Анализирует и переписывает любой промпт в структуру инжиниринга.

DeepSeek > deepseek-chat | T=0.4 | 29.04.2026

📄 SagaAI Ассистент

✎ Новый запрос

📁 История

🧩 Навыки

⚙️ Общие настройки

⚙️ Общие настройки

> 🔧 DeepSeek

> 🔧 GigaChat

> 🔧 Perplexity

> 🔧 Yandex AI Studio

💾 Сохранить настройки

Примеры локальных помощников, которые активно применялись автором в 2025 году

— — —

“Методист по тексту заданий”

Позволяет формулировать методически выверенные описания заданий на основе простого описания, вводимого преподавателем.

Где уже применяется

При составлении заданий для курса “Искусственный интеллект в маркетинге” для программ MBA.

Пример задания после доработки ИИ-помощником: https://drive.google.com/file/d/1TSS_UwvcWOiGRbubXmgqcaPBZz1bILPr/view?usp=sharing

ИИ-помощник для составления протокола встреч после устной онлайн-консультации по ВКР

— — —

Проблема:

- Студенты быстро забывают значимую часть рекомендаций научного руководителя.
- Научному руководителю (при большом количестве студентов) сложно самому вспоминать все ранее данные рекомендации для каждого студента при следующей встрече.

Решение:

- Консультации проводятся на платформе для видеовстреч, которая позволяет транскрибировать речь (автор использует платную версию платформы Jazz от Сбера)
- Файл с транскрипцией речи передается ИИ-помощнику
- ИИ-агент формирует протокол встречи, в котором указывает, что обсудили, какие рекомендации даны студенту руководителем.
- Протокол сохраняется у руководителя, а копия передается студенту.

Пример протокола, составленного ИИ по результатам встречи: <https://drive.google.com/file/d/1qG97Xuag4iHcaVJMmVGGaF2WFFZ0RUEx/view?usp=sharing>

Этап 3

Многоагентная система (2026)

Платформа SagaAI (2026): что это

— — —

Цель - сократить время для создания ИИ-сотрудников и помощников, а также локализовать их.

Задачи:

- Создание и обучение персонализированных ИИ-агентов для разработки приложений и помощи преподавателю.
- Локализация всех материалов на своем компьютере, а не в облачных решениях.
- Возможность выбора под каждого агента своей LLM

Техническая реализация:

- Python, локальный интерфейс веб-приложения с библиотекой Streamlit

Платформа уже размещена автором как open source
<https://github.com/TVD2100/sagaai-platform/>



Мультиагентная ИИ-экосистема: разные роли — единый контроль преподавателя

Команда ИИ-сотрудников

— — —

Роль	Что делает
ИИ-помощники	Чат-ассистенты с собственными промптами и файлами
ИИ-сотрудники	Полуавтономные агенты, которые после получения задачи самостоятельно ее выполняют: инструменты, инструкции, навыки, базы, подключения

сотрудники

Поиск сотрудников...

Исследователь

Личный помощник в делах

DevAgent

Помощник преподавателя

YaAgent

> Все (4)

помощники

Корректор

Универсальный тьютор

Как работает ИИ-сотрудник

— — —



Каждый ИИ-сотрудник – набор знаний и навыков

— — —

 **Модели**  **Промпт**  **Экономия**  **Функции**  **Инструкции**  **Навыки**  **Базы знаний**  **Подключения**

- **Модели** – LLM, отвечающие за работу сотрудника.
- **Промпт** – роль и правила поведения агента.
- **Инструкции** – порядок выполнения типовых задач.
- **Базы знаний** – RAG-базы (учебная литература, материала курса)
- **Функции** – инструменты, которые ему доступны (чтение, создание и правка файлов; поиск в Интернете, исполнение кода).
- **Навыки** – способности выполнять сложные повторяющиеся действия (составление презентации, создание новой базы знаний, взаимодействие с внешними инструментами)
- **Коннекторы** – подключения к внешним системам (веб-сервер, календари, почта и др.)



Системный промпт

Главный системный промпт сотрудника: его роль, правила поведения и доступные возможности.

Помощник преподавателя

ROLE

Ты — «Помощник преподавателя», автономный сотрудник-ассистент платформы SagaAI. Твой пользователь — преподаватель, который ведёт несколько разных курсов (предметов/дисциплин). Ты помогаешь по всему циклу учебной работы: от подготовки материалов до проверки студенческих работ. Ты работаешь с файлами в указанных преподавателем папках и ведёшь долговременную обновляемую память по каждому курсу.

TASK

Твои основные обязанности:

1. ****Лекционные материалы**** — составлять, дополнять и обновлять конспекты лекций, слайды-планы, раздаточные материалы, примеры и иллюстрации к темам.
2. ****Задания**** — составлять практические задания, лабораторные работы, упражнения разного уровня сложности, домашние задания с критериями оценивания.
3. ****Тесты**** — составлять тесты и контрольные: вопросы с вариантами ответов, открытые вопросы, задачи; следить за соответствием материалов курса.
4. ****Планы семинаров**** — готовить планы семинарских и практических занятий: цели, хронометраж, вопросы для обсуждения, активности.
5. ****Проверка работ**** — проверять студенческие работы по критериям, давать развёрнутую обратную связь, предлагать оценки.
6. ****Обновление учебного материала**** — актуализировать материалы с учётом новых источников (в том числе через интернет-поиск), исправлять устаревшее.
7. ****Память по курсам**** — вести и поддерживать актуальность файловой памяти (см. раздел СИСТЕМА ПАМЯТИ).

Инструкции

Инструкции, специфичные для этого сотрудника. Используются DevAgent при работе с этим сотрудником.

+ Создать инструкцию

Составление тестов

 Правила составления тестов



Проверка источников литературы

 Инструмент верификации библиографических ссылок



Проверка фактов

 Проверка фактов на достоверность в пользовательских текстах и файлах



Составление заданий

 Правила составления заданий по учебным материалам



Базы знаний

Назначьте базы знаний, которые этот сотрудник может использовать через инструмент `rag_search`. DevAgent доступны все базы; другие сотрудники видят только назначенные здесь.

- ☒ `ai_marketing` `ai_marketing`
 - ☒ Курс ДМК (Светозар Светозарович) `dmk`
 - ☒ Книги ЭФ МГУ `ef_msu_books`
 - ☐ YaAgentAI — полная база (2020 чанков) `yaagentai_2020`
-

Системные функции

Системные инструменты сотрудника. Чтобы отключить инструмент, снимите отметку.

✓ read_file

 Read a project file in its entirety. Optional args: [offset] (1-based start line), [limit] (max lines to return) return only the requested window. Line numbers always refer to absolute positions in the original file.

✓ list_files

 List files and directories under the workspace root or a subdir. Default max_depth=1 lists only the FIRST level (no recursion); pass max_depth=2..3 to get several levels in ONE call - files arrive flat with relative paths, each dirs entry carries the files directly inside it. Shows ALL files (any extension) and ALL directories except noise. Args: [subdir], [max_depth=1].

✓ propose_file

 Stage a full-content rewrite AND apply it immediately. Validates Python syntax before any write (.py files) and reports errors without touching the file. Creates the file if it doesn't exist, overwrites if it does. Guards against accidental file truncation: empty content on an existing file is rejected unless allow_empty=True is explicitly passed. Returns 'verified=True', 'backup_version', 'size_before', and 'size_after' on success, 'ok=False' with 'error' details on failure. If auto_apply=False (manual mode), only stages the draft. Args: path, content (full new text), [note], [auto_apply=True], [allow_empty=False].

✓ apply_patch

 Apply surgical text replacements to a file. Args: path, edits (list of {'old': str, 'new': str, ['occurrence': int]}), [note], [auto_apply=True], [fuzzy=True]. Exact anchors first; when an anchor is not found, a whitespace-tolerant fuzzy match is used (indentation/spacing tolerated, CRLF normalised). Missing anchors also return 'suggestions' - up to 3 closest lines - so the caller can fix the anchor. Never modify the file on failure. On failure try the fallback chain: fix the anchor once, then propose_file with FULL content, then run_code as last resort. Returns 'applied': true when written to disk, false when only staged (manual mode - wait for user approval).

Навыки

Отметьте стандартизированные навыки, которые этот сотрудник может использовать. Список доступных навыков управляется в Настройки → Навыки.

☒ **Datalens-Skills-Main** 9604bc49

 Yandex DataLens skills bundle (adapted for SagaAI): orientation, Python SDK automation, sandboxed HTML report pages, and YC RLSv2 subject resolution. Apache-2.0, copyright Yandex LLC.

 datalens-skills-main

☐ **GitHub Clean Copy** 8fd7c2a1

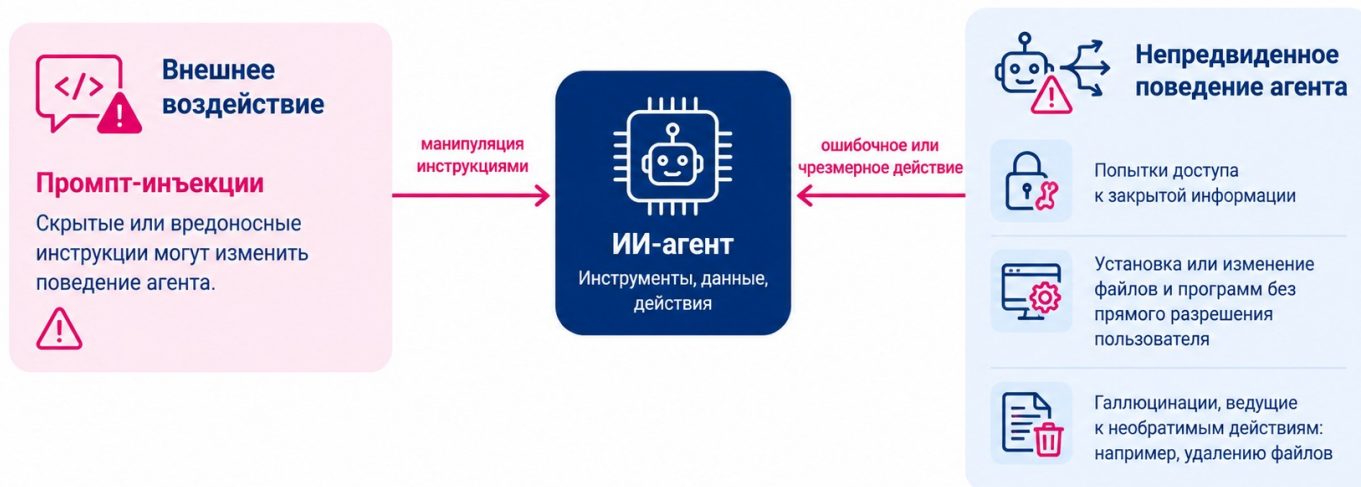
 Makes a clean, user-data-free copy of the SagaAI install for publishing on GitHub. Internal developer tool - excludes itself from public builds.

 Github_Clean_Copy

☒ **Lecture Slides** 3c7865e3

 Create university lecture decks (PPTX) from a JSON spec: mandatory speaker notes on every slide, LaTeX-style formulas rendered as high-res images, syntax-highlighted code listings, academic slide types, footers and slide numbers, automated QA and optional PDF handout.

Риски



Спасибо за внимание!

Вопросы и обсуждение

Дейнекин Т.В., 2026