

SAPLAND · 14 АПРЕЛЯ · 11:00

ABAP + AI =

новая скорость SAP-разработки

Автор: Никита Калущкий

Дата

14 апреля

Время

11:00 МСК

Длительность

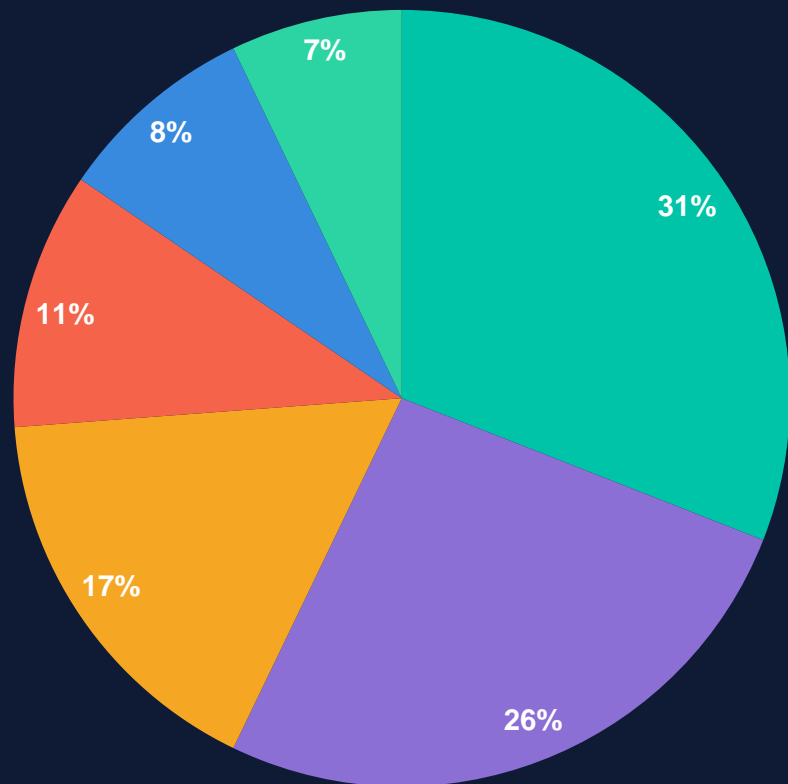
1 час

Формат

Онлайн

Что вас интересует: 84 вопроса участников

Анализ анкет перед вебинаром



- 31%** **Инструменты и выбор AI**
Какой AI выбрать? Cursor, Claude, Joule?
- 26%** **Генерация кода и ускорение**
Насколько реально ускорение? Живые примеры
- 17%** **Реальные кейсы и практика**
Опыт коллег, успехи и провалы
- 11%** **Безопасность / закрытый контур**
Как без Joule и без утечки данных?
- 8%** **Рефакторинг и документация**
Legacy-код, ABAP Doc, спецификации
- 7%** **AI-агенты и интеграция**
Агенты, MCP, NW-интеграция

АВАР-разработка входит в новую фазу зрелости

Внутреннее напряжение в командах

AI ускоряет

- Генерацию АВАР-кода
- Автоматизацию документации
- Анализ сложной бизнес-логики
- Рефакторинг legacy-кода
- Поиск ошибок и дампов (ST22)

Открытые вопросы

- *Каким инструментам можно доверять?*
- *Можно ли работать без SAP Joule?*
- *Какие ограничения критичны в закрытом контуре?*
- *Как не создать риски для безопасности?*
- *Как выстроить работу системно?*

Что разберём на вебинаре

01

Генерация АВАР-кода с помощью AI

Что работает, что — нет; живые примеры кода · запрошено 26% участников

02

Автоматическая документация и рефакторинг legacy

АВАР Doc, FORM→CLASS, функциональные спецификации · запрошено 34% участников

03

AI без SAP Joule: работа в закрытом контуре

Безопасность данных, локальные модели · топ-тема у 11% участников

04

Реальные кейсы + AI-агенты и интеграция

Опыт коллег, агенты, MCP, on-premise · запрошено 24% участников

Что такое искусственный интеллект?

Простое определение — без маркетинга

AI — это программа, обученная на огромном количестве текста. Она умеет предсказывать «что написать дальше» — очень хорошо. Это не разум. Это очень продвинутое автодополнение.

Что умеет хорошо

- Генерировать и объяснять текст и код
- Находить паттерны в большом объёме данных
- Суммировать, переводить, документировать
- Имитировать стиль и структуру — убедительно
- Работать 24/7 без усталости и без жалоб

Чего не умеет

- Думать как человек — нет понимания, только паттерн
- Точно считать — математику "угадывает", не вычисляет
- Знать текущие события — знания ограничены датой обучения
- Отличить правду от убедительной лжи (галлюцинации)
- Помнить прошлые разговоры — каждый диалог с чистого листа

Вывод: AI — мощный инструмент, а не волшебник. Чем лучше вы его понимаете — тем эффективнее используете.

Обучение...



LLM — большая языковая модель

Как это работает и какие они бывают

Как создают LLM

- 1 Сбор данных**
Триллионы слов: Wikipedia, книги, код на GitHub, статьи. Модель читает весь интернет.
- 2 Предобучение**
Нейросеть учится предсказывать следующее слово. Миллионы итераций на тысячах GPU.
- 3 Fine-tuning + RLHF**
Люди оценивают ответы. Модель учится быть полезной, точной и безопасной.
- 4 Готово к работе**
Модель развёртывается в облаке. Вы пишете промпт — она отвечает.

Какие бывают LLM

GPT-4o / GPT-5

OpenAI

Самый массовый. ChatGPT — интерфейс поверх него

Claude Sonnet/Opus

Anthropic

Сильный в коде и длинных текстах. Этот вебинар готовил я с его помощью

Gemini

Google

Интегрирован в Google Workspace и поиск

Llama 3 / Mistral

Open source

Бесплатные модели. Можно запустить локально — без интернета

SAP Joule

SAP

Закрывающаяся модель, встроена в ВРР. Знает контекст SAP-системы

AI-агенты: от чат-бота к автопилоту

Когда AI не просто отвечает — а действует

Обычный LLM (чат-бот)

- Вы пишете вопрос
- Модель отвечает текстом
- Вы читаете, сами делаете дальше
- Один шаг — один ответ

AI-агент

- Вы задаёте цель
- Агент сам планирует шаги
- Вызывает инструменты, читает файлы, пишет код
- Несколько шагов — без вашего участия

Пример: «Найди все программы в пакете Z_FINANCE, добавь ABAP Doc к методам без документации и создай транспорт» — агент делает это сам, шаг за шагом.

Планирование

Разбивает задачу на шаги сам, без ваших инструкций

Инструменты

Вызывает API, читает файлы, выполняет код

Память

Помнит результаты предыдущих шагов в рамках сессии

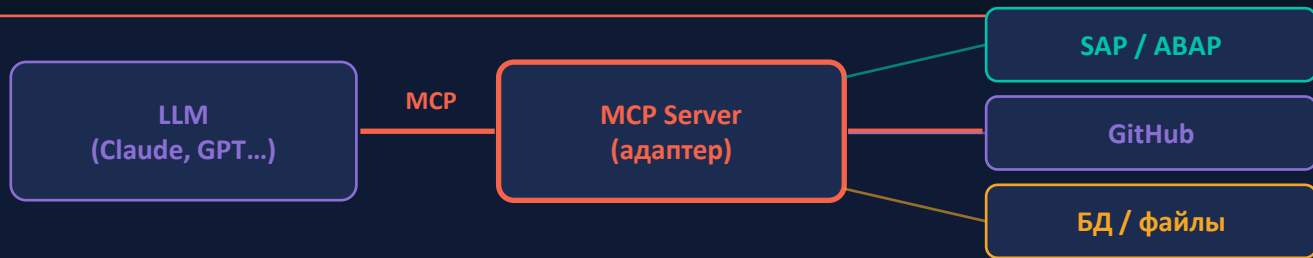
Самокоррекция

Если шаг не удался — меняет подход и пробует снова

MCP — Model Context Protocol

Стандарт, который позволяет AI работать с вашими инструментами

Аналогия: MCP — это как USB для AI. Один стандартный разъём, к которому подключаются любые инструменты: базы данных, IDE, файлы, API, браузер.



Стандарт, не самоделка

Один протокол — все AI-инструменты. Написал MCP-сервер раз — работает с любым LLM

AI видит ваш контекст

Модель читает файлы, структуру репозитория, данные БД — и отвечает точнее

Для ABAP-разработчика

MCP-сервер для ADT → AI видит объекты репозитория, запускает код, читает дампы

Уже работает сегодня

Claude, Cursor, VS Code Copilot поддерживают MCP. Open source, бесплатно

RAG — Retrieval-Augmented Generation

Как научить AI работать с вашими данными

RAG — это способ дать AI доступ к вашим документам и данным. Вместо того чтобы переобучать модель, мы просто подкладываем нужную информацию прямо в запрос. AI отвечает — опираясь на ваши данные, а не на общие знания.

1

Вопрос

Пользователь задаёт вопрос:
«Какие рейсы LH есть
из Франкфурта?»

2

Поиск

Система ищет релевантные
фрагменты в базе данных
SPFLI / документах / PDF

3

Контекст

Найденные данные
добавляются к запросу в LLM
вместе с вопросом

4

Ответ

AI отвечает, опираясь на
ваши реальные данные —
точно и без галлюцинаций

В SAP-контексте: RAG позволяет AI отвечать на вопросы по вашей документации, техническим спецификациям, таблицам SPFLI/SFLIGHT или Z-таблицам — без обучения новой модели и без выгрузки данных наружу.

RAG vs переобучение модели: переобучение — дорого, долго, нужны тысячи примеров. RAG — дешево, работает сразу, данные можно обновлять в любой момент. Именно поэтому RAG стал стандартом для корпоративного AI.

LSP — Language Server Protocol

Как редактор кода «понимает» ваш язык программирования

LSP — стандарт, который позволяет редактору кода (VS Code, Eclipse, IntelliJ) понимать язык программирования: подсвечивать ошибки, предлагать автодополнение, навигировать по коду. Без LSP редактор видит просто текст.

Без LSP

- Редактор видит текстовый файл — и ничего больше
- Нет подсветки ошибок, нет автодополнения
- Перейти к определению метода — невозможно

С LSP

- + Редактор знает типы, методы, классы вашего кода
- + Ошибки подсвечиваются прямо в процессе набора
- + F3 — перейти к определению. Shift+F6 — переименовать везде

ABAP в SAP ADT: Eclipse + ADT использует LSP-подобный протокол для связи с сервером SAP. Именно поэтому у вас работает автодополнение, навигация и проверка синтаксиса прямо в IDE.

Причём тут AI? Cursor и GitHub Copilot подключаются к LSP-серверу вашего языка — и получают тот же контекст, что и редактор: типы, структуру, ошибки. Это позволяет AI генерировать код, который действительно компилируется, а не просто выглядит правдоподобно.

Cursor и GitHub Copilot: под капотом

Что за модели, на чём обучались и как применять в АВАР

Cursor

VS Code + AI · AnySphere, 2022

Модели: GPT-5.3 Codex (основная, от OpenAI), Claude Sonnet/Opus, Gemini, собственный Composer (MoE). Можно выбрать любую под задачу.

Плюсы для АВАР-разработчика

- + Глубокий контекст репозитория — понимает структуру проекта
- + Agent-режим: многошаговые задачи без участия разработчика
- + Поддержка MCP — можно подключить SAP через кастомный сервер
- + Privacy Mode: код не сохраняется и не идёт в обучение
- + Лучший вариант для рефакторинга крупных АВАР-файлов локально

Ограничения

- Нет нативной интеграции с ADT/SE80 — код нужно копировать
- Не знает Z-таблицы и бизнес-логику вашей системы
- По умолчанию отправляет код в облако — не для закрытого контура

Вывод: Cursor — мощнее для рефакторинга и агентных задач; Copilot — удобнее прямо в IDE. Оба дополняют, а не заменяют друг друга.

GitHub Copilot

GitHub + OpenAI + Microsoft · с 2021

Обучение: начинал на Codex (GPT-3 + миллиарды строк публичного кода с GitHub). Сейчас: GPT-4.1, GPT-5, Claude Opus/Sonnet — на выбор.

Встроен в ADT (Eclipse) — Есть расширение для Eclipse — привычная среда АВАР-разработчика

Inline autocomplete — Предлагает код прямо в редакторе, по мере набора — без переключения контекста

Слабее знает АВАР — Обучался на публичном коде; АВАР на GitHub мало → галлюцинации чаще, чем в Python/JS

Multi-model с 2024 — Можно переключиться на Claude Opus или GPT-5 для сложных задач прямо в IDE

Данные обучения — С апреля 2025 GitHub может использовать ваши взаимодействия для дообучения (можно отключить)

Генерация АВАР-кода с AI

Пример: агентство путешествий · таблицы SPFLI, SFLIGHT, SBOOK

ПРОМПТ

"Напиши метод АВАР OO, который по коду рейса (CARRID + CONNID) возвращает список свободных мест на все даты вылета из таблиц SPFLI и SFLIGHT. Учти: мест не менее 10."

АВАР OO — сгенерировано AI

```
METHOD get_available_flights.  
  SELECT f~fldate, f~seatsmax - f~seatsocc  
        AS free_seats  
  INTO TABLE rt_flights  
  FROM spfli AS l  
  INNER JOIN sflight AS f  
    ON l~carrid = f~carrid  
   AND l~connid = f~connid  
  WHERE l~carrid = iv_carrid  
   AND l~connid = iv_connid  
   AND f~seatsmax - f~seatsocc >= 10.
```

Таблицы демо-системы SAP: **SPFLI** — маршруты (CARRID, CONNID, CITYFROM, CITYTO) **SFLIGHT** — рейсы (дата, мест всего / занято) **SBOOK** — бронирования пассажиров **SCUSTOM** — клиенты агентства

Хорошо работает

- JOIN по стандартным таблицам (SPFLI, SFLIGHT, SBOOK)
- Вычисляемые поля (seatsmax - seatsocc)
- Фильтрация и АВАР Дос-комментарии
- Report-шаблоны, ALV-отчёты по рейсам

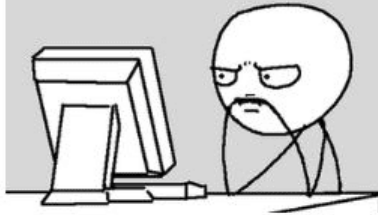
Требует контроля

- Кастомные расширения Z-таблиц поверх демо-модели
- Бизнес-правила конкретного клиента (тарифы, классы)
- Производительность на реальных объёмах
- Всегда проверять перед транспортом

Генерация АВАР-кода с AI

Before Chat GPT

* Developer coding - 2 hours

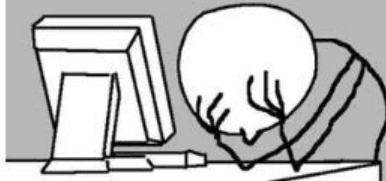


After Chat GPT

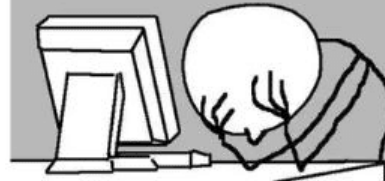
* ChatGPT generating code - 5 min



* Developer debugging - 6 hours



* Developer debugging - 24 hours



Рефакторинг и автодокументация

Три применения AI, которые экономят часы

FORM → CLASS: рефакторинг

AI строит скелет классов и методов из PERFORM-рутин. Разработчик контролирует бизнес-логику и тестирует результат.

Автоматическая документация

AI генерирует "!"-комментарии (ABAP Doc): описание метода, параметры IMPORTING/EXPORTING/RETURNING, исключения.

Анализ legacy-кода

Понять чужую программу на тысячу строк за 30 секунд. AI объясняет логику, указывает точки входа и зоны риска.

Функциональная спецификация

AI читает старый ABAP-код и генерирует описание бизнес-логики в виде текста — для передачи консультанту или заказчику.

Практика: вставляем 50-строчный FORM-рутин → получаем класс + ABAP Doc + функциональную спецификацию

AI без SAP Joule: работа в закрытом контуре

Как внедрить AI без риска для безопасности данных

Универсальные LLM

- Claude, ChatGPT, Gemini
- Рефакторинг и документация вне IDE
- Анонимизируйте код перед отправкой
- Без согласования CISO — не выгружать

Локальные модели

- Ollama + CodeLlama / DeepSeek
- Работают внутри периметра
- Данные не покидают инфраструктуру
- Требуют GPU-сервера и настройки

SAP Joule / BTP AI

- Нативная интеграция в ADT и BAS
- Знает контекст объектов репозитория
- Доступен с BTP-подпиской
- Оптимален для on-system задач

Реальный эффект: где выигрыш во времени

Оценки на основе практики SAP-команд

×4–6

Генерация кода

быстрее vs написание с нуля

~70%

Документация

времени экономится

×3

Анализ кода

быстрее понимание legacy

-40%

Отладка / ST22

времени на root cause



Почему Joule — не всегда лучший выбор

Честный разбор ограничений для ABAP-разработчика

Жёсткая привязка к SAP Cloud

Joule работает только внутри ВТР и S/4HANA Cloud. On-premise ECC / S/4HANA on-prem — нет доступа. Нужна ВТР-подписка.

Не знает ваш кастомный код

Joule не видит Z-таблицы, кастомные FM и бизнес-логику вашей системы без дополнительной настройки Document Grounding.

Отстаёт по интеллекту от ChatGPT / Claude

Базовый Joule использует более слабые модели. ChatGPT-4o и Claude Sonnet стабильно превосходят его в сложных задачах рефакторинга и объяснения кода.

Сложная настройка —

Document Grounding, ВТР provisioning, role mapping — несколько дней работы архитектора

Нет IDE для On-prem —

Интеграция с ADT/Eclipse — только для ВТР-окружений. On-premise разработчики остаются в стороне

Закрытая модель

—
Нельзя выбрать модель или настроить промпт-стратегию. Black box — полное доверие SAP

Медленный релизный цикл —

Обновления выходят с SAP-релизами. Конкуренты (Claude, GPT) обновляются ежемесячно

Сравнение AI-инструментов для ABAP-разработчика

Стоимость и возможности · команда из 5 человек

	ChatGPT Plus OpenAI	Claude Pro Anthropic	GitHub Copilot Pro Microsoft / GitHub	Cursor Pro Anysphere
	\$20/мес	\$20/мес	\$10/мес	\$20/мес
Качество модели	Отличное	Отличное	Отличное	Отличное
Знание ABAP	Среднее	Среднее	Среднее	Среднее
On-premise SAP	Да ✓	Да ✓	Да ✓	Да ✓
Интеграция с ADT/IDE	Нет ✗	Нет ✗	Да ✓	Частично
Agent-режим	Да ✓	Да ✓	Да ✓	Да ✓
Закрытый контур	Нет ✗	Нет ✗	Нет ✗	Privacy Mode
5 чел / год	\$1 200	\$1 200	\$600	\$1 200

Источники: [1] GitHub Copilot Plans — github.com/features/copilot/plans [2] Cursor pricing — cursor.com/pricing [3] Claude / ChatGPT pricing — anthropic.com/pricing, openai.com/pricing

Спасибо за внимание!



Вопросы?