

Открывая Завтра

Выездная ИТ-конференция

г. Пекин | 22-25 апреля 2026

NERPA

Yandex Cloud

DJON

ICS

Третий день

Программа на сегодня

09:00-09:40 Подходы и развитие ИИ в компании

Виталий Попов, директор департамента реализации инфраструктурных проектов Софтлайн Решения

09:40-10:00 ITSM для людей - первый шаг навстречу порядку

Иван Сташко, менеджер направления ESM\ITSM Софтлайн Решения

10:00-10:30 Кофе-брейк

10:30-11:30 Круглый стол ITSM и ITAM без галстуков

Модератор: Иван Сташко, менеджер направления ESM\ITSM

11:30-12:30 Круглый стол Комплексный подход к модернизации ИТ-Инфраструктуры. Опыт наших клиентов.

Модератор: Виталий Попов, директор департамента реализации инфраструктурных проектов Софтлайн Решения

12:30 Обед



Подходы и развитие ИИ в компании

как начать и как продолжить

Практическая логика внедрения: от первого процесса и пилота до масштабирования на новые функции

Open-source / on-prem / облако

Пилот → эксплуатация → масштаб

AI как часть нового процесса



Зачем компании вообще системно идти в ИИ

Что изменилось

- Open-source LLM/VLM стали достаточно зрелыми для прикладных сценариев
- Появился понятный формат запуска: discovery → пилот → интеграция → эксплуатация
- On-prem и изолированный контур снимают часть барьеров по безопасности

Где эффект заметен быстрее всего

- Потоки документов и писем
- Техподдержка, маршрутизация, copilot
- Мониторинг, инциденты, сводки и рекомендации

Что получает бизнес

- Скорость первого ответа
- Меньше ручной рутины
- Лучшее качество передачи контекста между людьми и системами

Ключевой вывод

Самый сильный результат дают не “чат-боты сами по себе”, а решения, где ИИ включён в поток данных, правила, контроль качества и следующий шаг процесса.

Именно поэтому разговор про ИИ быстро становится разговором про операционную модель компании.

Тезис №1. ИИ можно встроить в процесс — но лучше переделать процесс под ИИ

Встраивание в старый процесс

ИИ сидит отдельным шагом рядом с человеком
Много ручных проверок и повторного ввода данных
Ответ ИИ не становится действием — его ещё нужно “дотащить” до системы
ROI спорный: красиво, но медленно масштабируется



Процесс, спроектированный под ИИ

Единая точка входа: письмо, документ, тикет, лог, фото
Сразу идёт извлечение, классификация, обогащение контекстом и проверка правил
Результат — это уже карточка кейса, JSON, маршрут, уведомление или следующий шаг
Человек подключается только там, где нужен контроль или исключение

Практическое правило: не спрашивать “куда приткнуть ИИ в старую схему?”, а пересобирать путь кейса так, чтобы ИИ стал естественным первым слоем анализа и подготовки действия.

Как начать: первый use case должен быть маленьким, большим и измеримым

1

Выбрать поток

Там, где много однотипного входа: письма, документы, обращения, инциденты

2

Ограничить задачу

Например: классификация, извлечение полей, summary, маршрутизация

3

Взять реальные данные

Не игрушечный датасет, а живые кейсы компании с шумом и ошибками

4

Сразу задать метрики

Точность, скорость ответа, доля ручной доработки, экономия времени

5

Оставить human-in-the-loop

Человек подтверждает спорные места и учит систему на исключениях

6

Довести до действия

Выход — не текст ради текста, а JSON, тикет, API, уведомление, маршрут

Типовой горизонт входа Discovery 2–3 недели → пилот 4–8 недель → решение о масштабировании по цифрам, а не по ощущениям.

Тезис №2. Студенты могут решать задачи — это недорогой и рабочий ресурс

Где студенты реально полезны

- Сбор и разметка данных, подготовка датасетов
- Тестирование промптов, сценариев, quality loops
- Сбор пайплайнов вокруг open-source моделей
- Прототипирование интерфейсов и API
- Сравнение моделей и фиксация метрик на реальных кейсах

Но не в одиночку

- Нужен сильный архитектурный контур
- Нужны готовые правила качества и приёмки
- Нужно понимание бизнес-процесса, а не только модели

Рабочая модель команды

1 опытный архитектор / тимлид + 2–5 студентов или junior-инженеров на пилот и быстрые итерации.

Почему это экономически разумно

- Входной барьер в open-source стек ниже, чем в классическую enterprise-разработку
- Много задач сводится к системной проверке гипотез, а не к “магии модели”
- Пока бизнес ищет подход, дешевле покупать скорость эксперимента, чем зрелость “на вырост”

Риск

без наставничества такая команда быстро упрётся в качество, инфраструктуру и интеграцию.

Тезис №3. Оборудование — основная проблема, но с пониманием эксплуатации становится проще

Что болит на старте

- Непонятно, какой GPU и сколько VRAM действительно нужны
- Модель “завелась” ≠ решение стабильно работает в процессе
- Совместимость SDK, драйверов, контейнеров и моделей съедает сроки
- Реальные метрики — latency, throughput, утилизация и стоимость владения

Что помогает упростить выбор

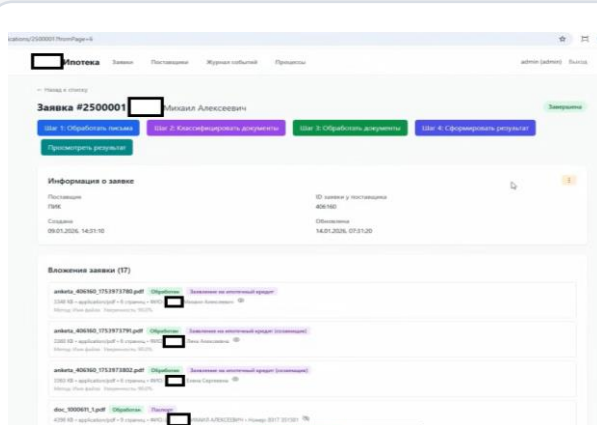
Идти от нагрузки конкретного use case, тестировать стенд заранее и принимать решение не по “бренду GPU”, а по зрелости стека под вашу задачу.



Практический подход

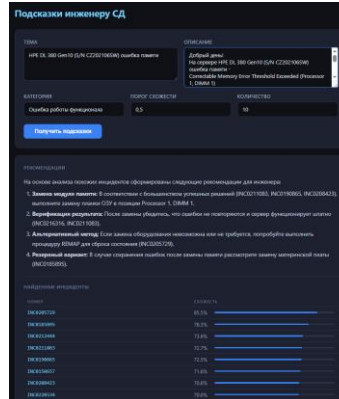
подготовка стенда → портирование моделей и SDK → прогон тест-плана → сравнение результатов → рекомендация по пилоту.

С чего обычно начинается реальная польза: 3 класса кейсов



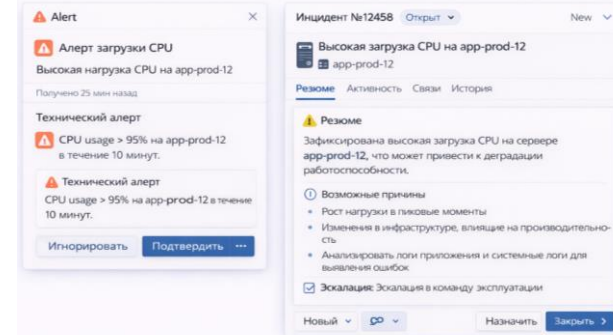
Документы и письма

- Извлечение полей и классификация
- Проверка полноты пакета
- Структурированный результат для целевой системы



Техподдержка и маршрутизация

- Автоклассификация и назначение линии
- Summary для оператора
- База знаний и copilot как следующий шаг



Мониторинг и инциденты

- Резюме алерта и контекст
- Первичная диагностика
- Рекомендации и переход к проактивной аналитике

Главное: у всех трёх кейсов одинаковая архитектурная логика — входящий поток → AI-анализ → контроль → действие в процессе.

Тезис №4. Аппетит приходит во время еды: ИИ начинает проникать везде



Следствие для менеджмента

Нужно думать не только про отдельные внедрения, а про библиотеку повторно используемых AI-компонентов и про очередь процессов на переосмысление.

Что важно запомнить

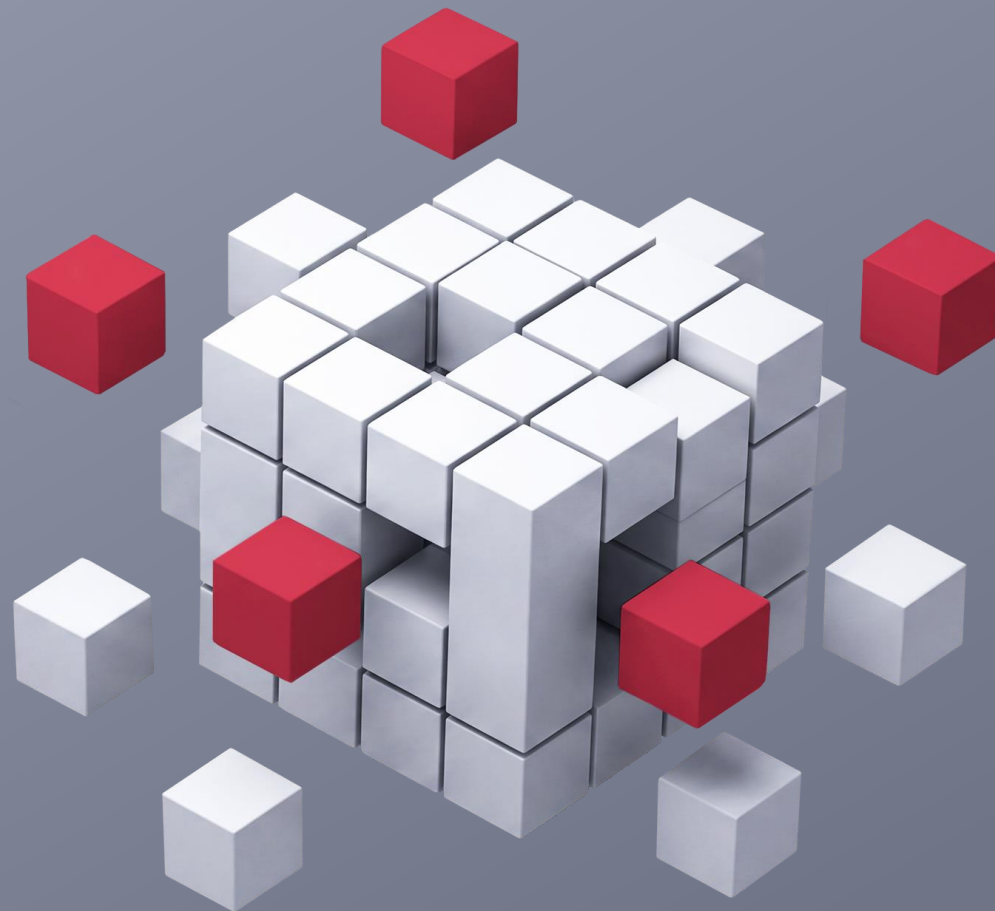
Лучше не встраивать ИИ в старый процесс, а пересобрать процесс под работу ИИ

Начинать стоит с одного измеримого use case на реальных данных

Для старта можно дешево усиливаться студентами и junior-командой — при наличии сильного наставника

Главный практический барьер — оборудование и inference-стек, но он управляем при нормальной инженерной методике

После первого эффекта ИИ начинает естественно расползаться по соседним функциям — и это уже путь к AI-first компании



Формула старта

больной процесс + реальные данные + метрики + human-in-the-loop + готовность менять сам процесс