

ИИ в разработке:

Как перейти от экспериментов к системному результату с AI-native SDLC

Виталий Хабаров

Техлид практик DevOps

- 15+ лет опыта работы в эксплуатации и автоматизации. Более 10 лет опыта внедрения DevOps-практик.
- Строю и оптимизирую процессы поставки и эксплуатации в компаниях любого масштаба, от стартапов до технологических гигантов.
- С 2020 провожу исследование состояния DevOps в России.
- Исследую эффективное применение ИИ в разработке.



О чём не будем говорить

01 Какая модель лучше?

02 Какие навыки использовать?

03 Как написать тот самый промпт?



О чём поговорим

- 01 Проблематика внедрения ИИ
- 02 Как системно внедрять ИИ
- 03 Как отслеживать результат
- 04 Эволюционное развитие AI-native SDLC: помощник, конвейер, фабрика



СФЛАНТ

Синергия опыта вендора, интегратора,
сервисной и консалтинговой компании



Deckhouse — продуктивное подразделение, разработчик продуктов для построения надёжной enterprise-инфраструктуры



DaaS — комплексное DevOps-сопровождение инфраструктуры в режиме 24/7 силами выделенной DevOps-команды



«Экспресс 42» — DevOps-консалтинг. От анализа узких мест в ИТ-процессах до создания роадмапа изменения ИТ для реализации цифровой трансформации

Вебинар подготовлен
при поддержке «Онтико» —
организатора ведущих ИТ-
конференций России



Проблематика



Теневое использование ИИ

До 89% сотрудников используют персональные решения для работы¹



Разработчики не хотят использовать ИИ

16 % не планируют использовать ИИ, а 46 % не доверяют результатам ИИ²



Успешный опыт не масштабируется

Внутри одной организации разница в эффекте от ИИ может быть 4 и более раз³



Качество кода ухудшилось, вырос тех. долг

Код, написанный с ИИ, содержит в 1,7 раза больше проблем, чем написанный человеком⁴



Никто уже не разбирается в системе из-за объёмов AI-кода

Испытуемые с ИИ ошибались в 1,5 раза чаще в понимании собственного кода⁵



Внедрили AI, но нет эффекта для бизнеса (ROI)

32 % организаций не могут связать внедрение ИИ с бизнес-результатами⁶

¹ [Teramind – Shadow AI Report 2026](#)

² [StackOverflow – Developer Survey 2025](#)

³ [AI Coding Tools Attract Top Performers – But Do They Create Them?](#)

⁴ [State of AI vs. Human Code Generation Report](#)

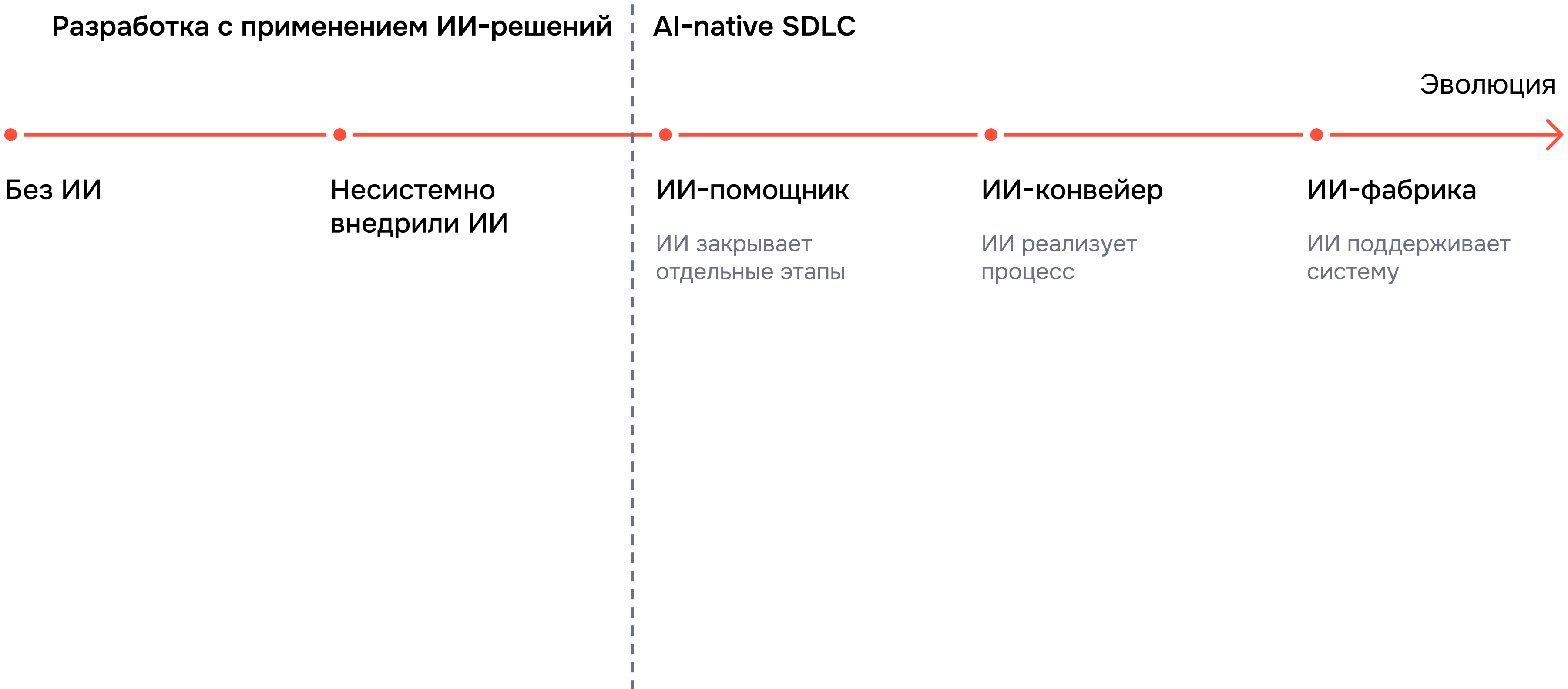
⁵ [How AI Impacts Skill Formation](#)

⁶ [The State of Code Abundance 2026](#)

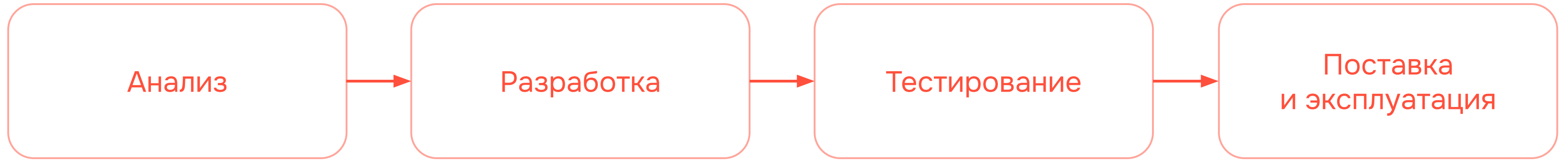
Переход к AI-native SDLC



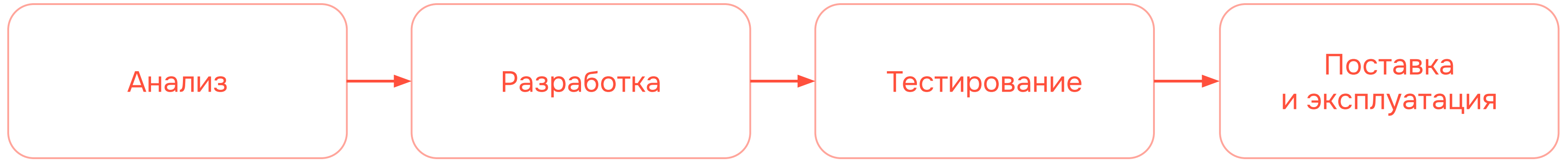
Переход к AI-native SDLC



Классический SDLC



Классический SDLC. Проблематика



- Теневое использование ИИ
- Разработчики не хотят использовать ИИ

Корпоративное решение



Нужно дать решение:

- Которым удобно пользоваться
- Которое даёт достаточно хороший результат
- У которого понятны правила использования



Для этого:

- Опрашиваем сотрудников об их опыте
- Реализуем самые частые сценарии
- Отрабатываем возражения

Метрики



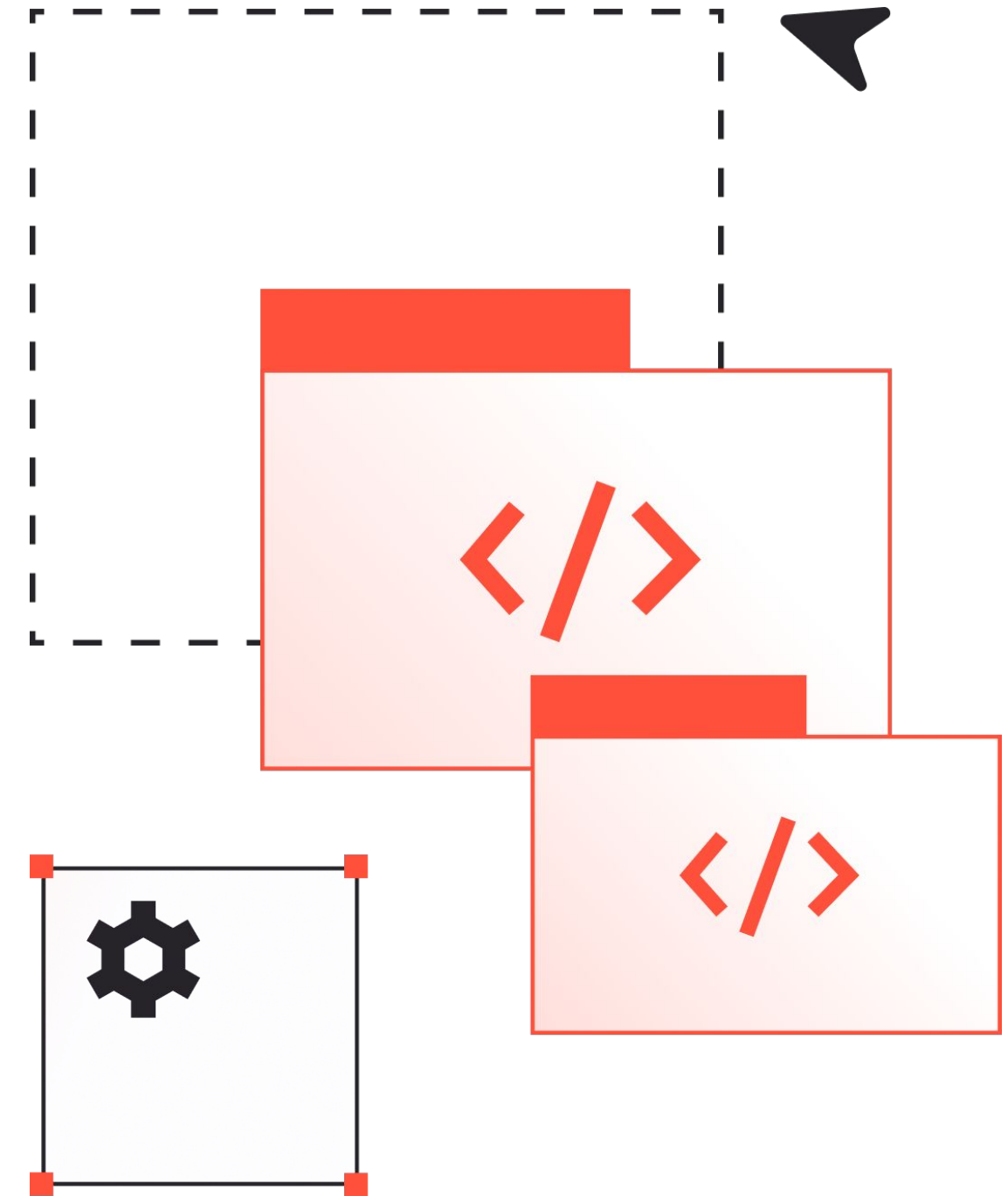
Weekly Active Users



Частота обращения к ИИ

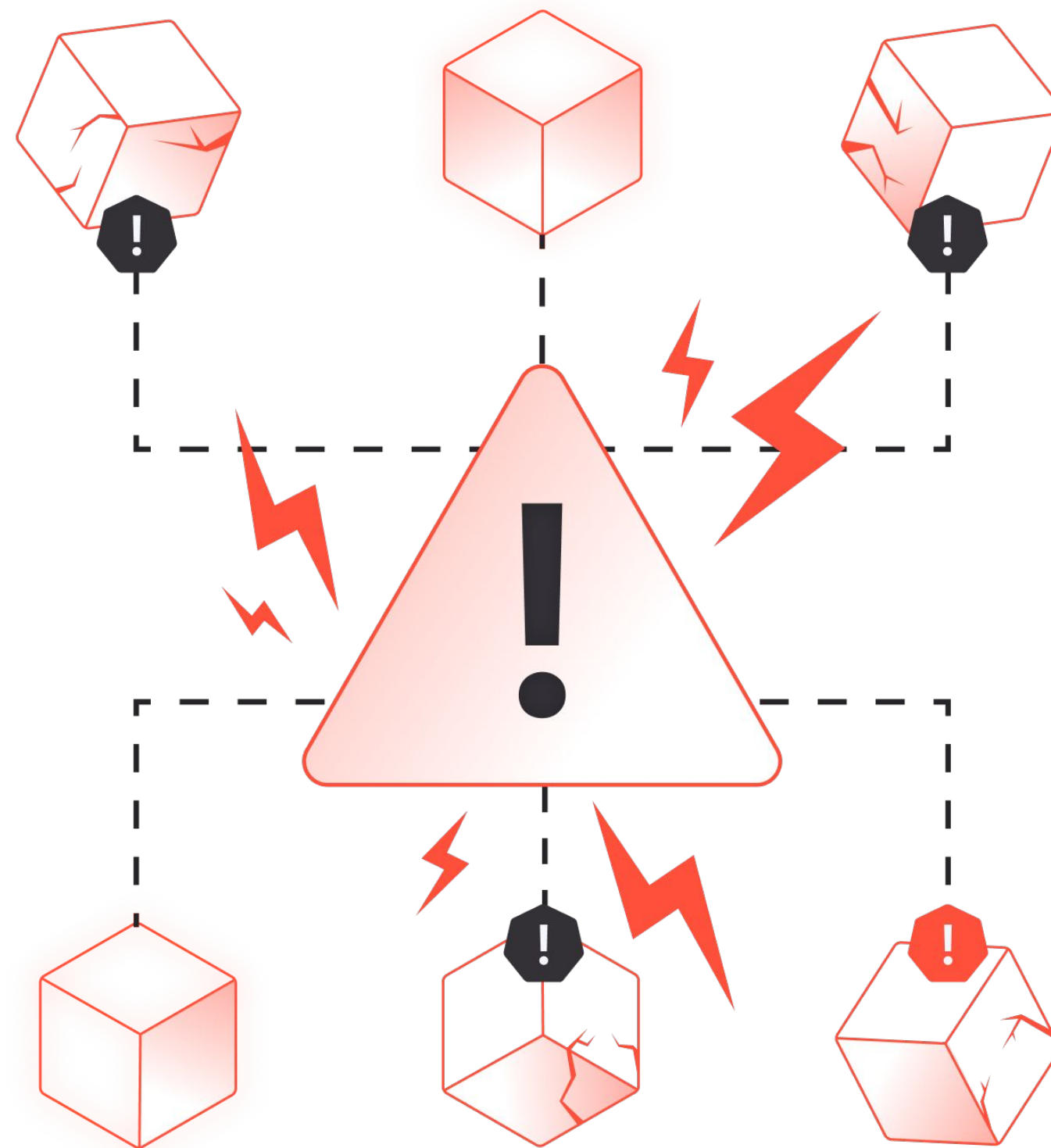


Количество задач, решённых с помощью ИИ



Риски

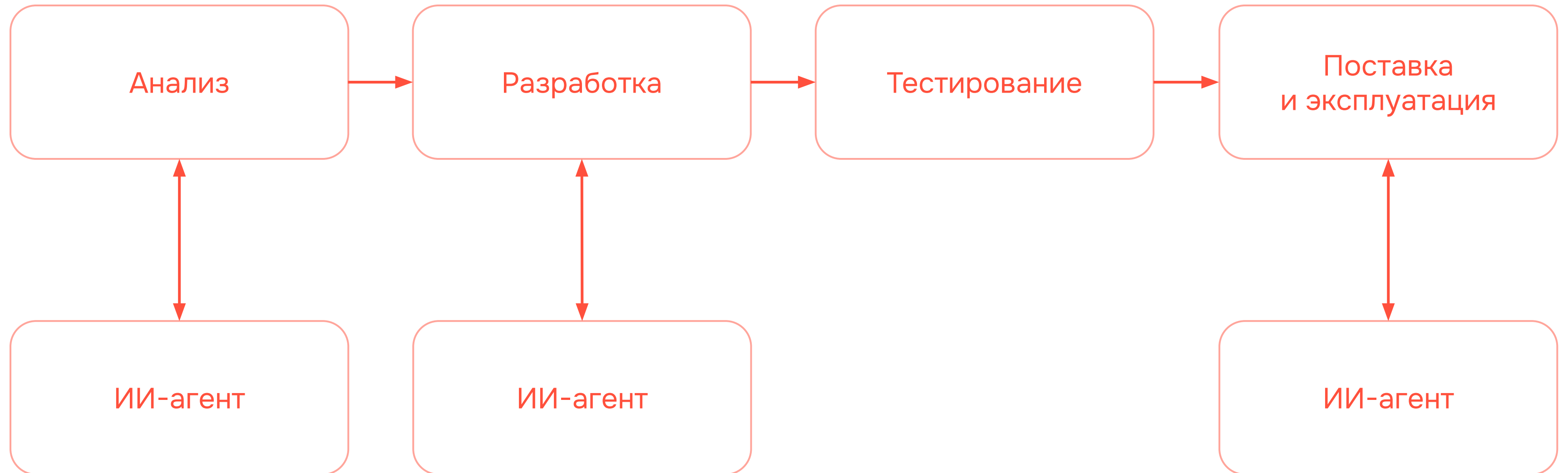
- ! Жгут токены
- ! Всё равно не пользуются
- ! Отправляют конфиденциальные данные



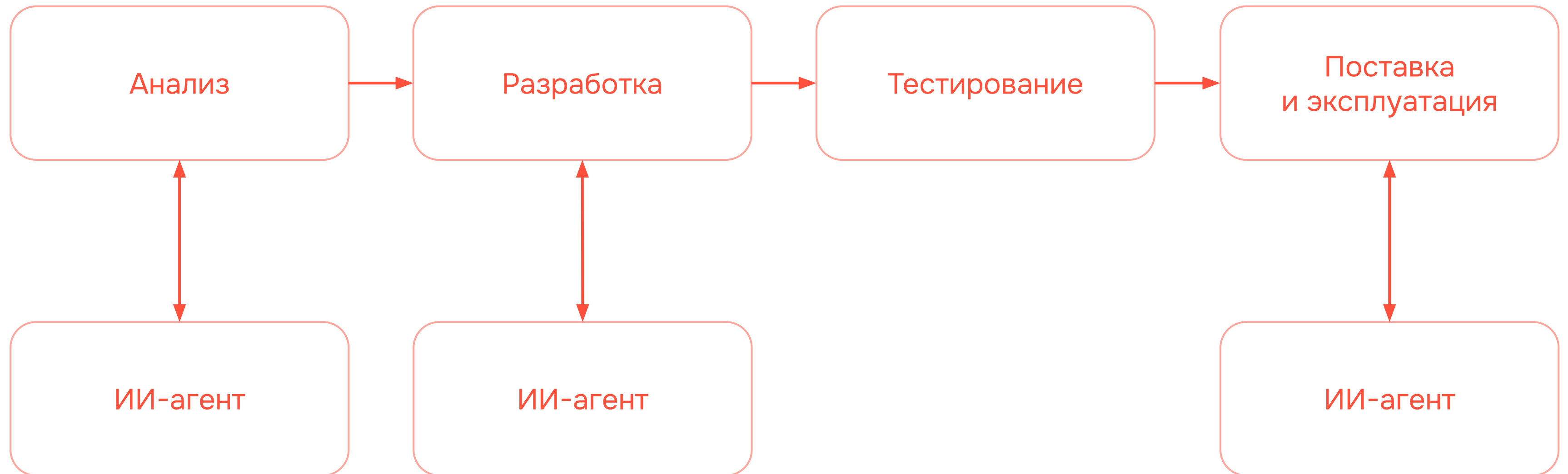
Результат

- ✓ Люди воспринимают ИИ как рабочий инструмент
- ✓ Сокращается количество ИИ-скептиков
- ✓ За ИИ-решением появляются прозрачность и управляемость

Несистемное внедрение ИИ



Несистемное внедрение ИИ. Проблематика



- Качество кода ухудшилось
- Успешный опыт не масштабируется
- Никто не разбирается в системе из-за объёмов ИИ-кода
- Внедрили ИИ, но нет эффекта для бизнеса (ROI)

Три столпа работы с агентом

Столп 1



Компетенции
пользователя

Столп 2



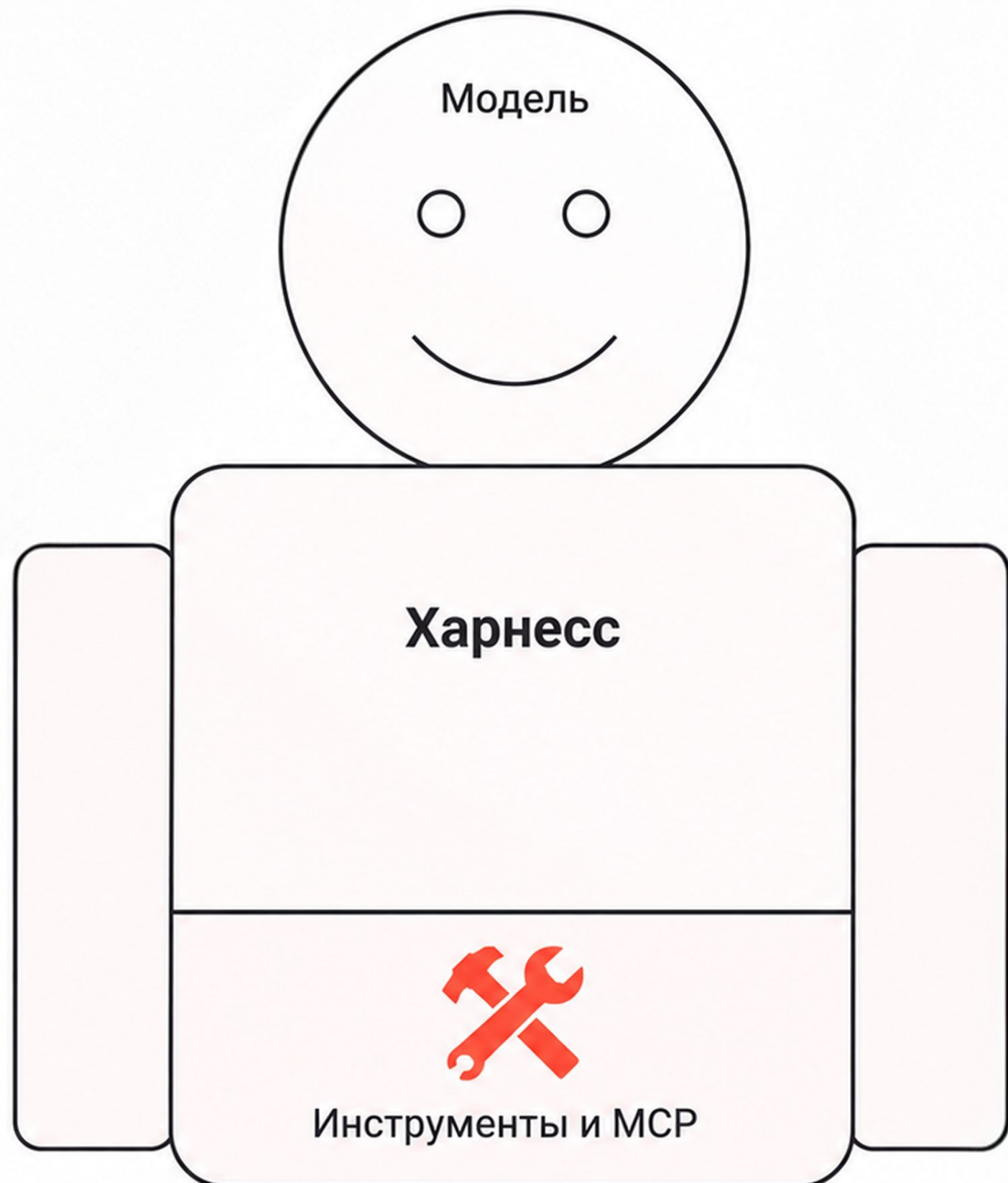
Capabilities
агента

Столп 3



Стандарты
и политики

Агент-фрилансер



Имеет:

- Знание всех возможных паттернов
- Знание всех технологий на момент обучения
- Усреднённый индустриальный опыт

Не знает:

- Ничего про ваш продукт
- Ничего про ваш контекст
- Ничего, что не вошло в обучающую выборку
- Что такое человеческое восприятие и ощущения

Три направления работы

Столп 1



Компетенции
пользователя

- Нарработка интуиции

Столп 2



Capabilities
агента

- Индекс по знаниям

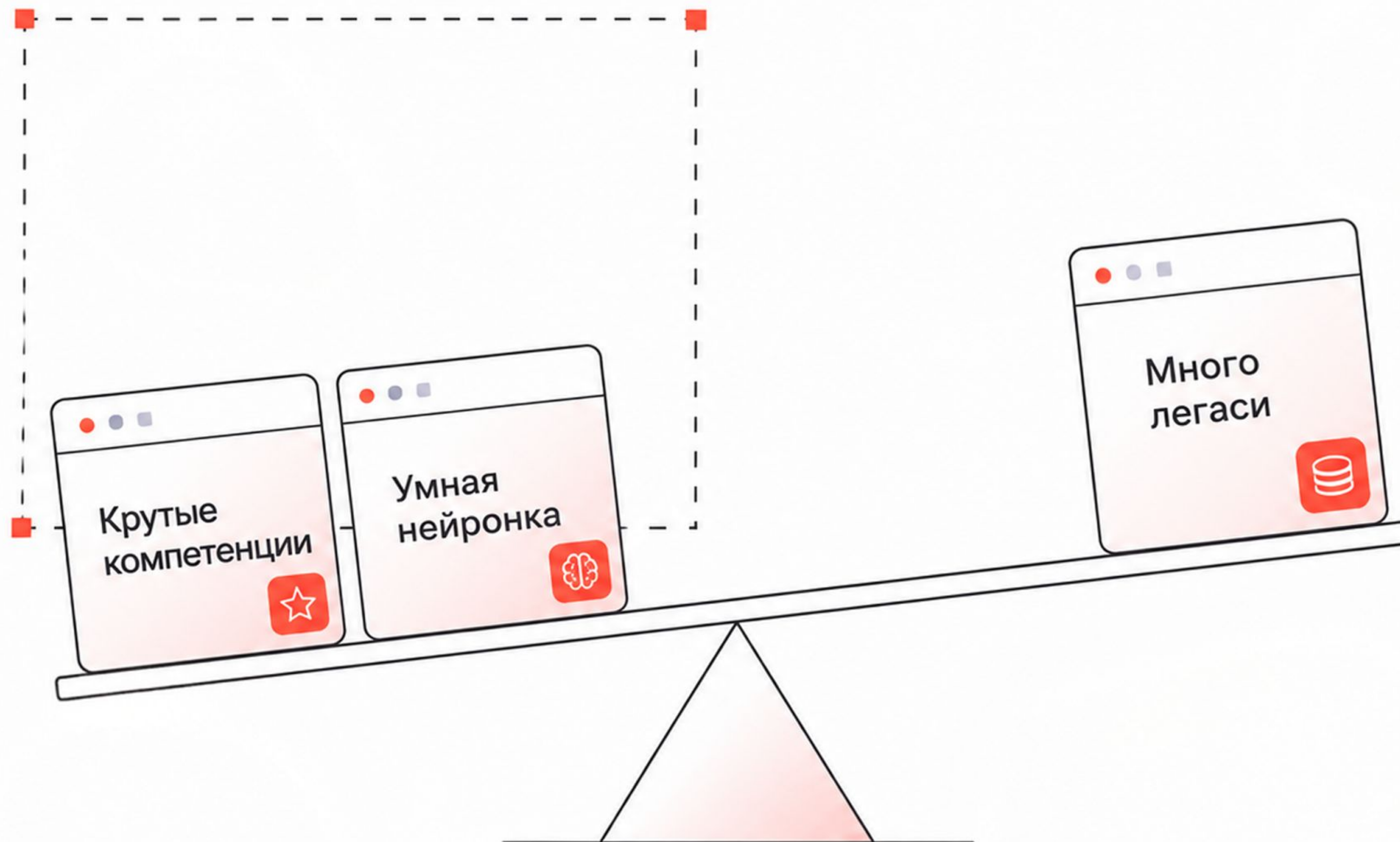
Столп 3



Стандарты
и политики

- Архитектура продукта

Результат = совокупность факторов



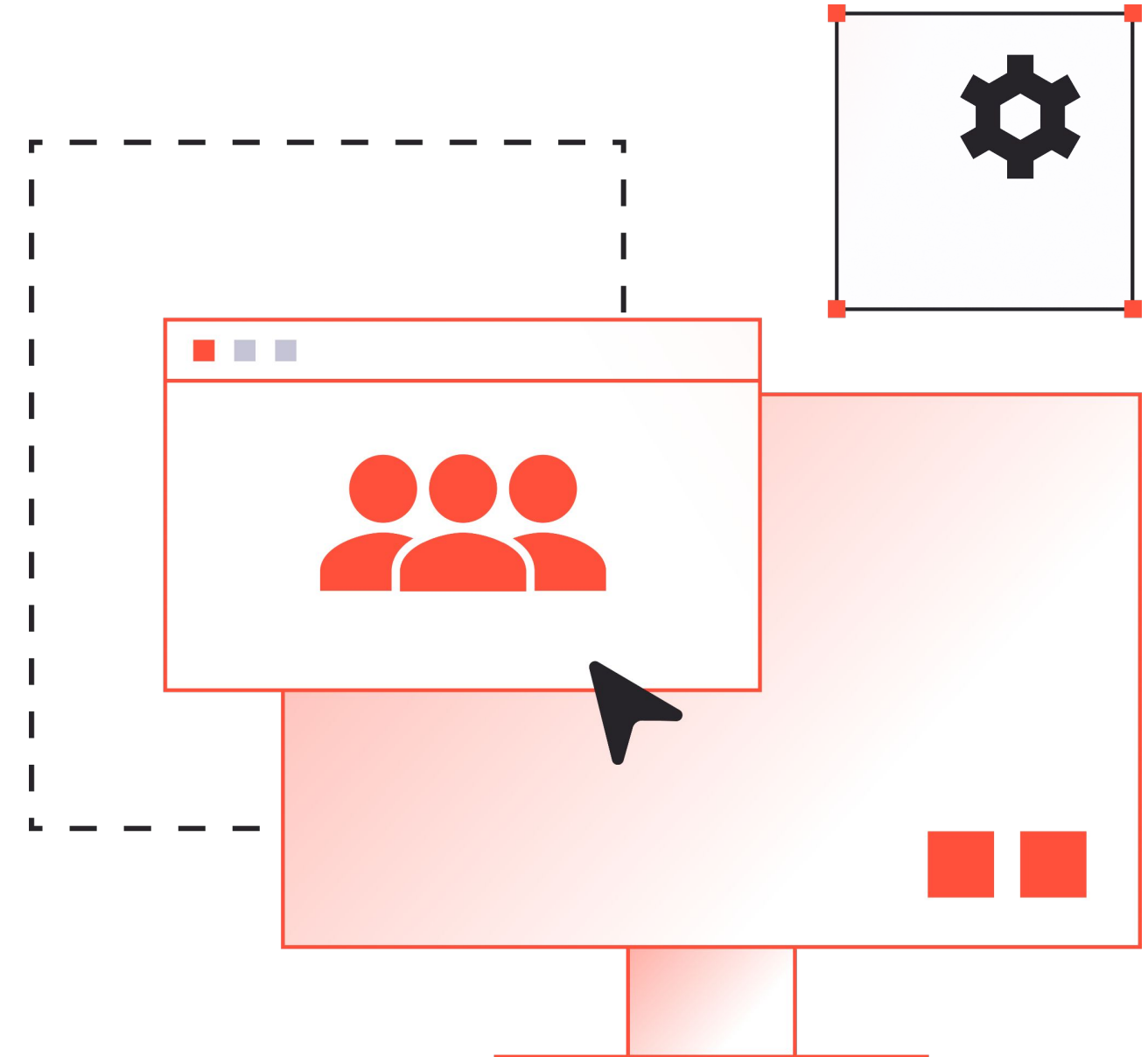
Метрики



Количество выполненных задач



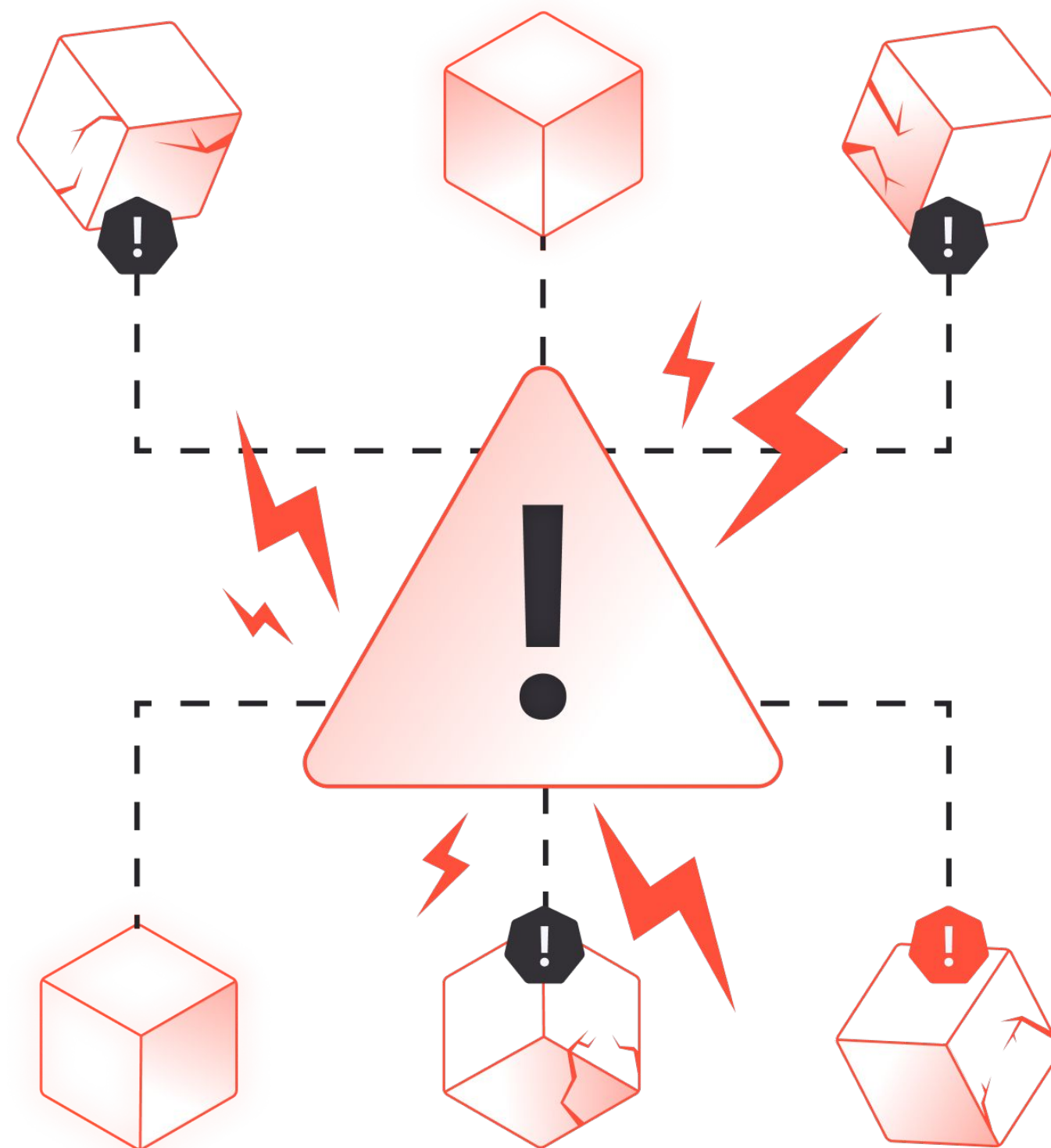
Эффективность отдельных сотрудников



Риски



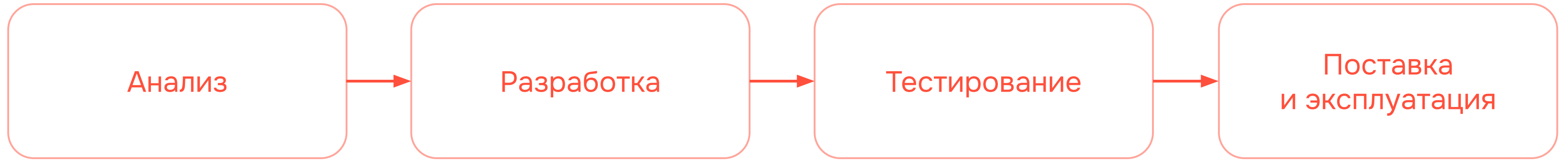
Сотрудники продолжают
работать по-старому



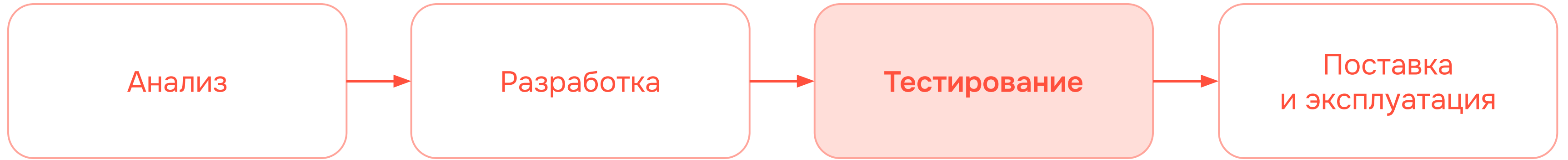
Результат

- ✓ Архитектура продукта становится управляемой
- ✓ Эффективность сотрудников растёт

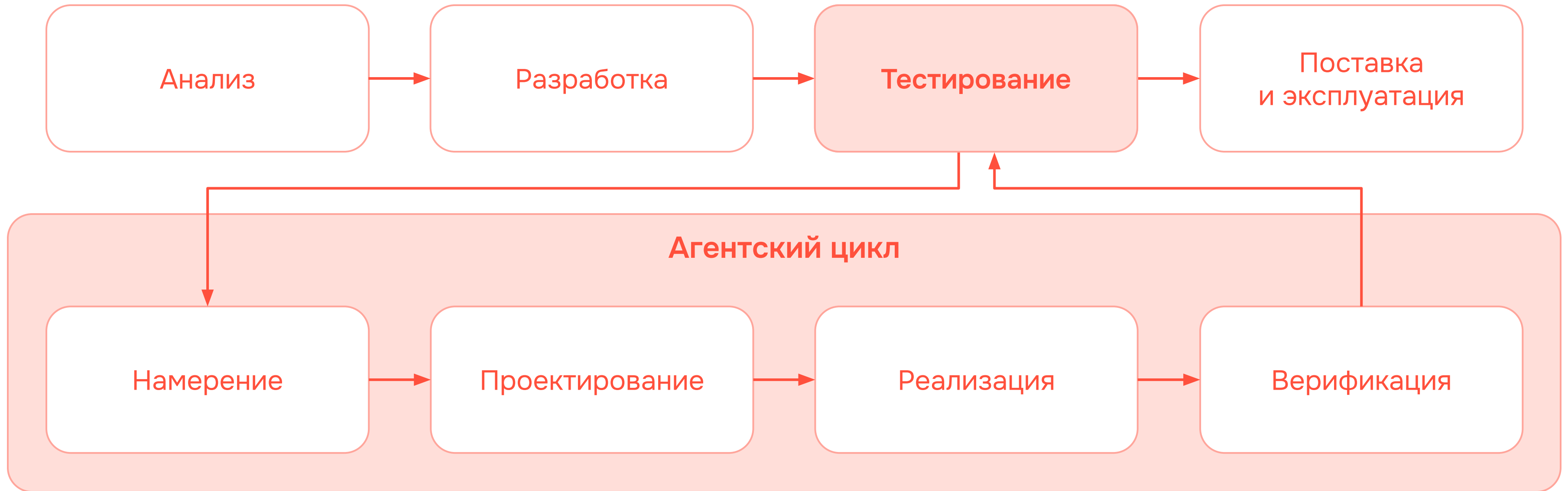
Системное внедрение ИИ



Системное внедрение ИИ

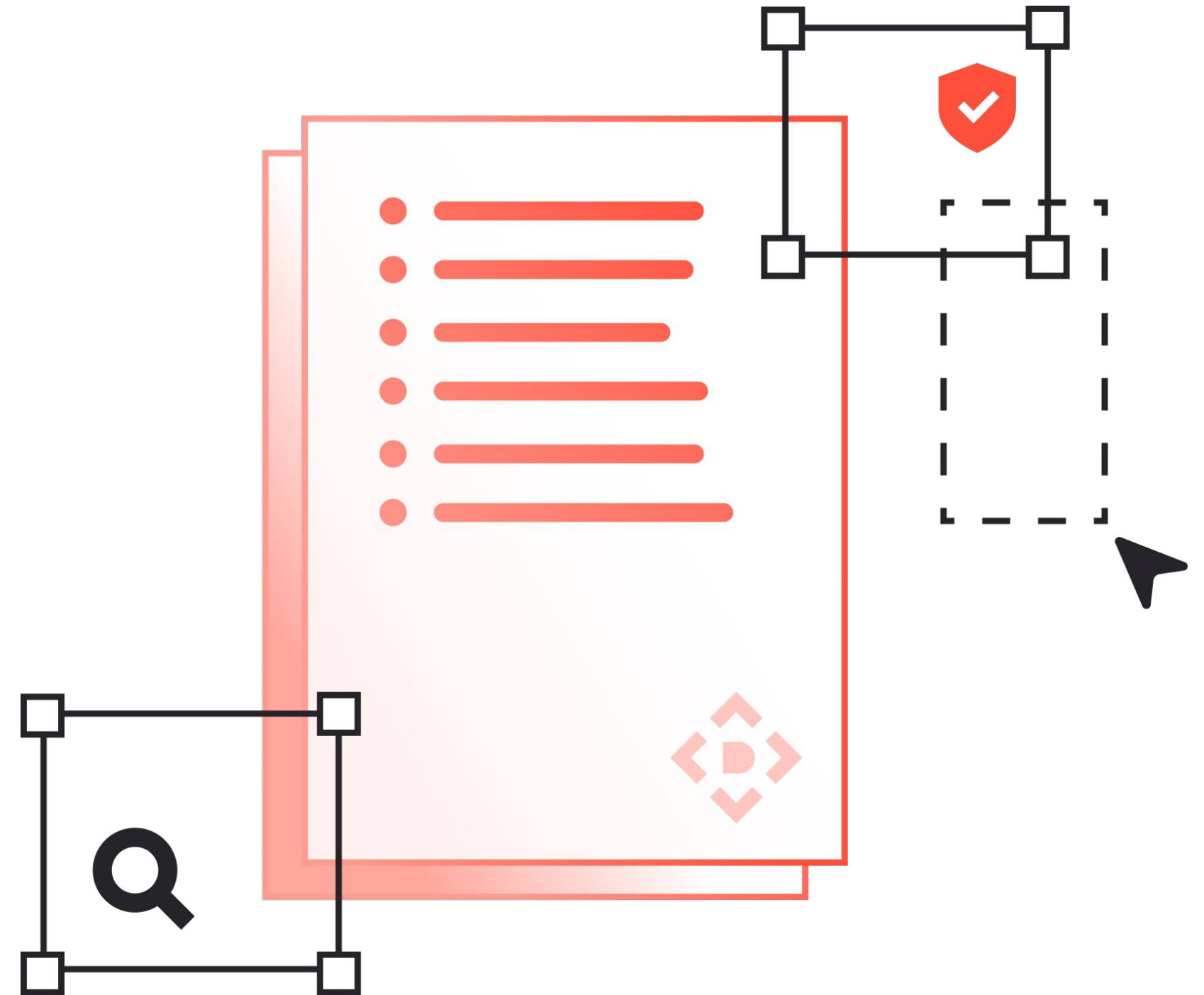


ИИ-помощник



Систематизация

- ✓ Обнаружить ограничение
- ✓ Снять ограничение с помощью автоматизации
- ✓ Переиспользовать накопившийся опыт



Три направления работы

Столп 1



Компетенции
пользователя

- Prompt engineering

Столп 2



Capabilities

- MCP и Tools

Столп 3



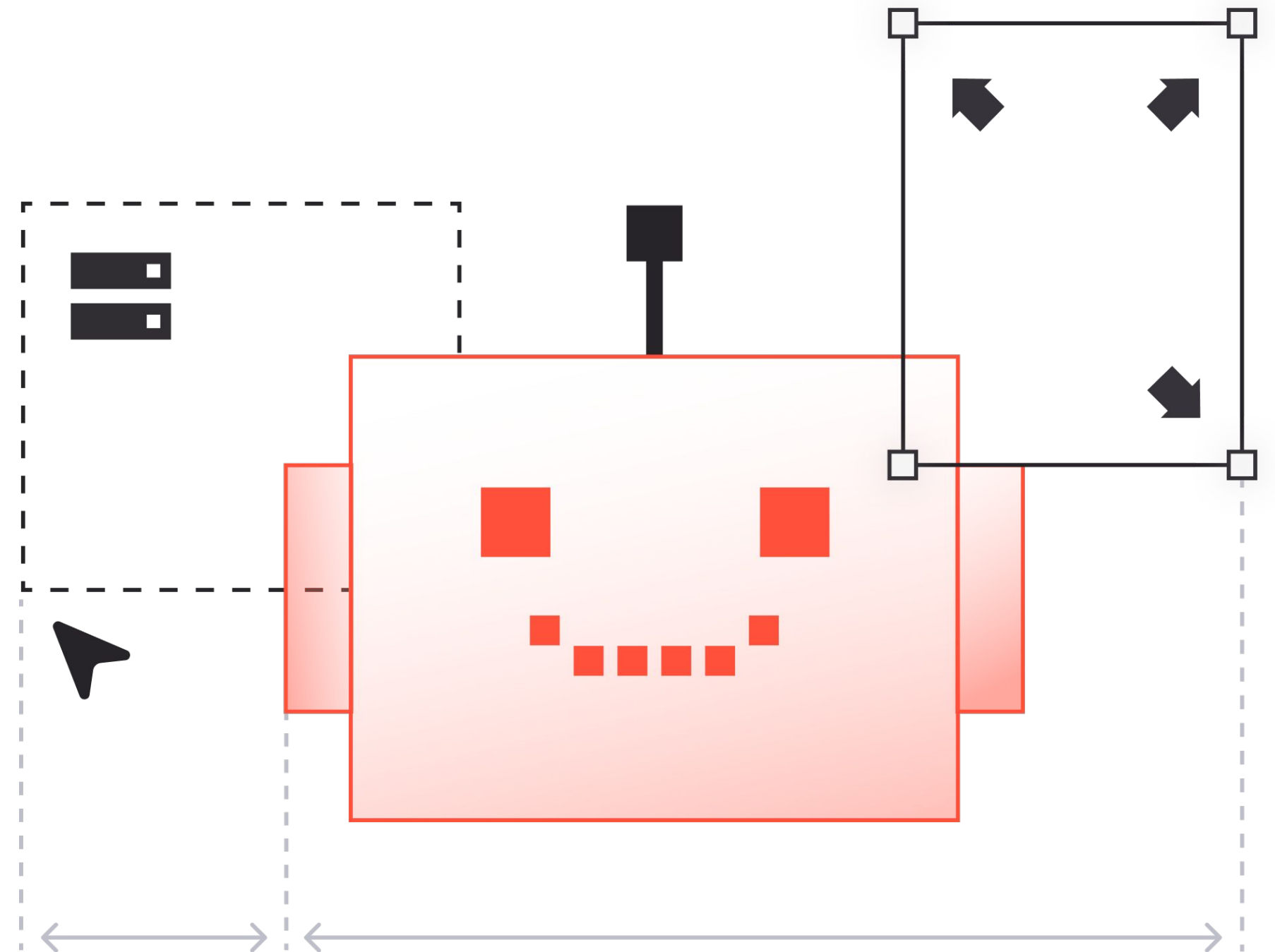
Стандарты
и политики

- Верификация

Метрики

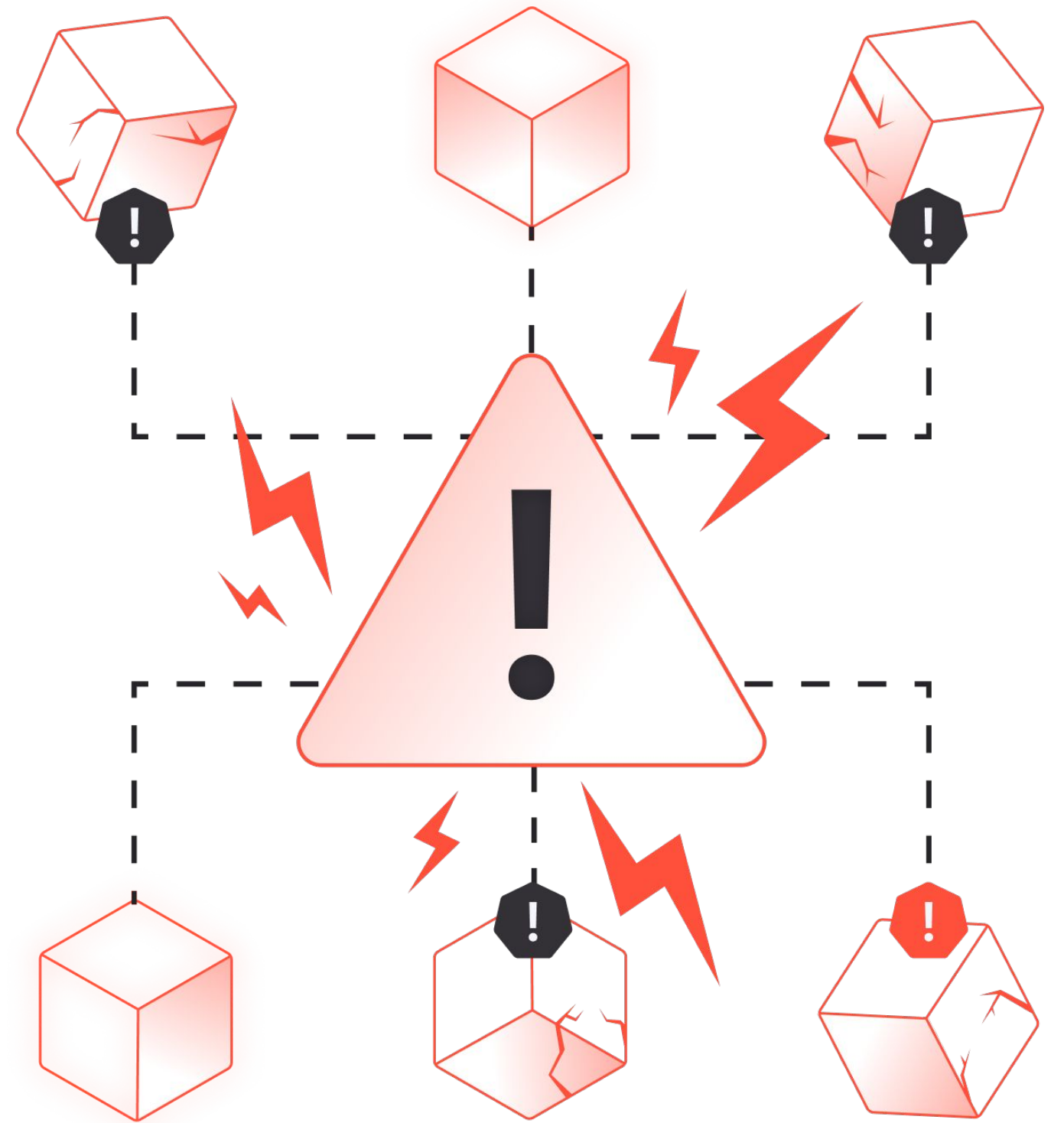


Скорость и качество выполнения задачи, автоматизируемой с помощью ИИ

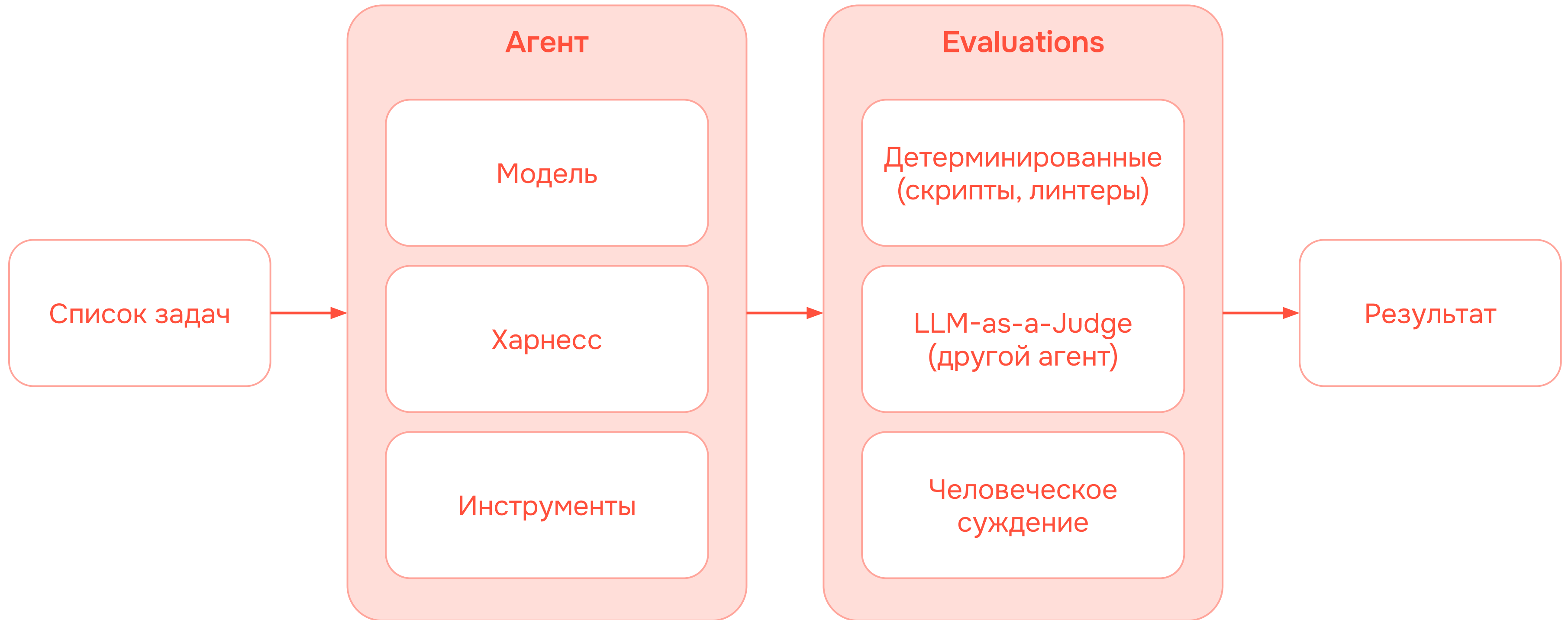


Риски

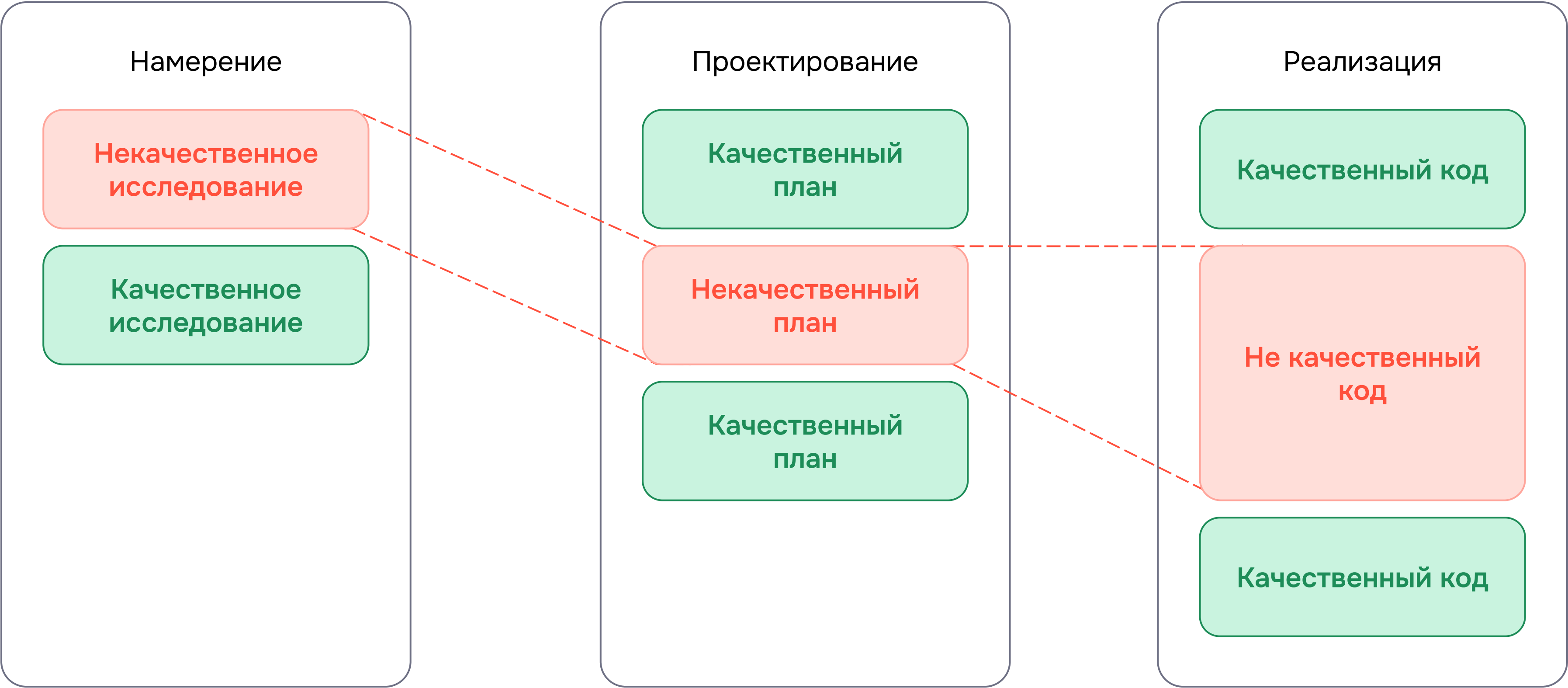
- ! Не измеряем качество агентского цикла
- ! Не контролируем промежуточные результаты



Оценка



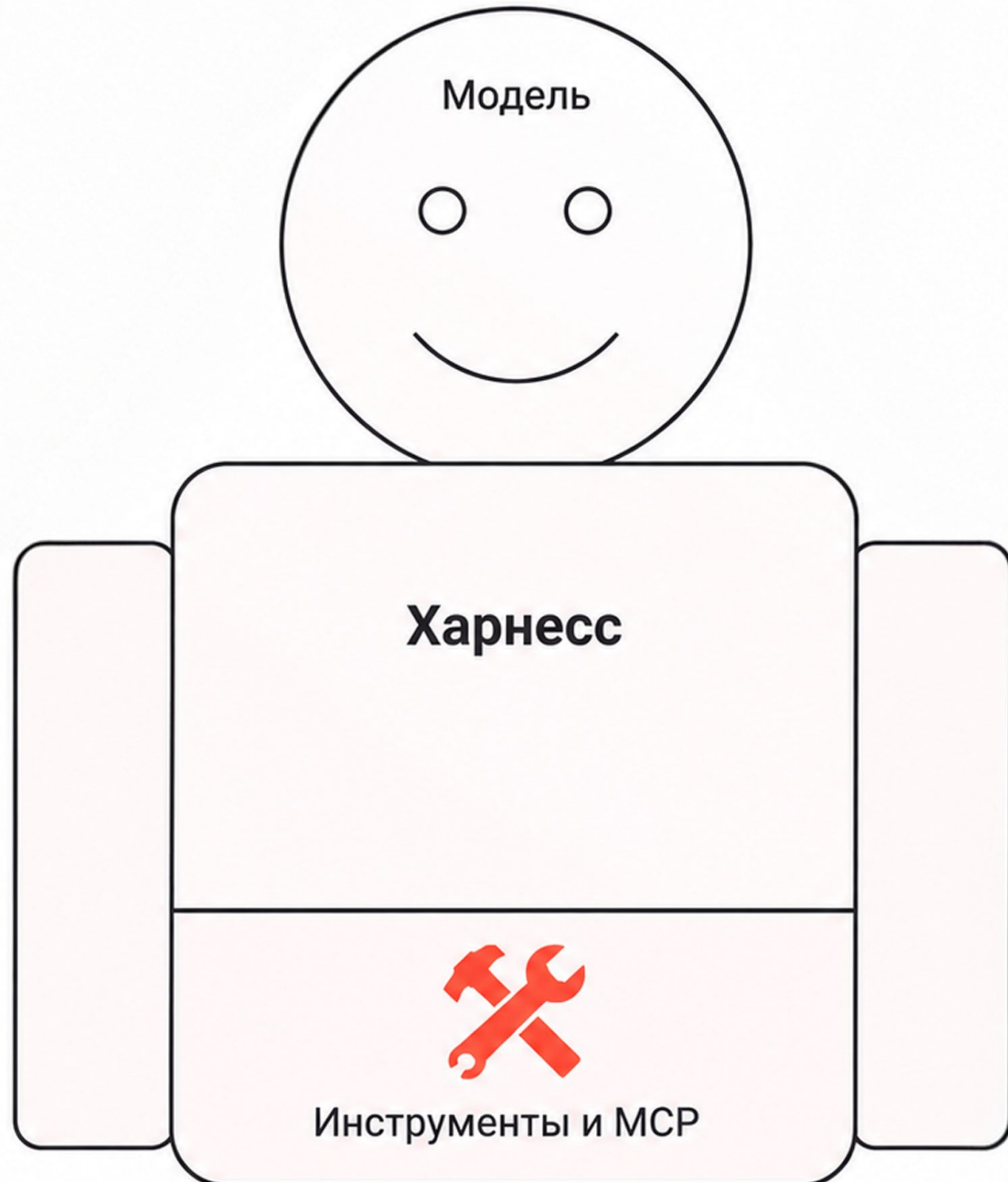
Отсутствие контроля качества



Результат

- ✓ Качество продукта становится управляемым
- ✓ Снятие ограничения, улучшение процессных метрик
- ✓ Успешные практики становятся частью системы

Аутстафф-отдел



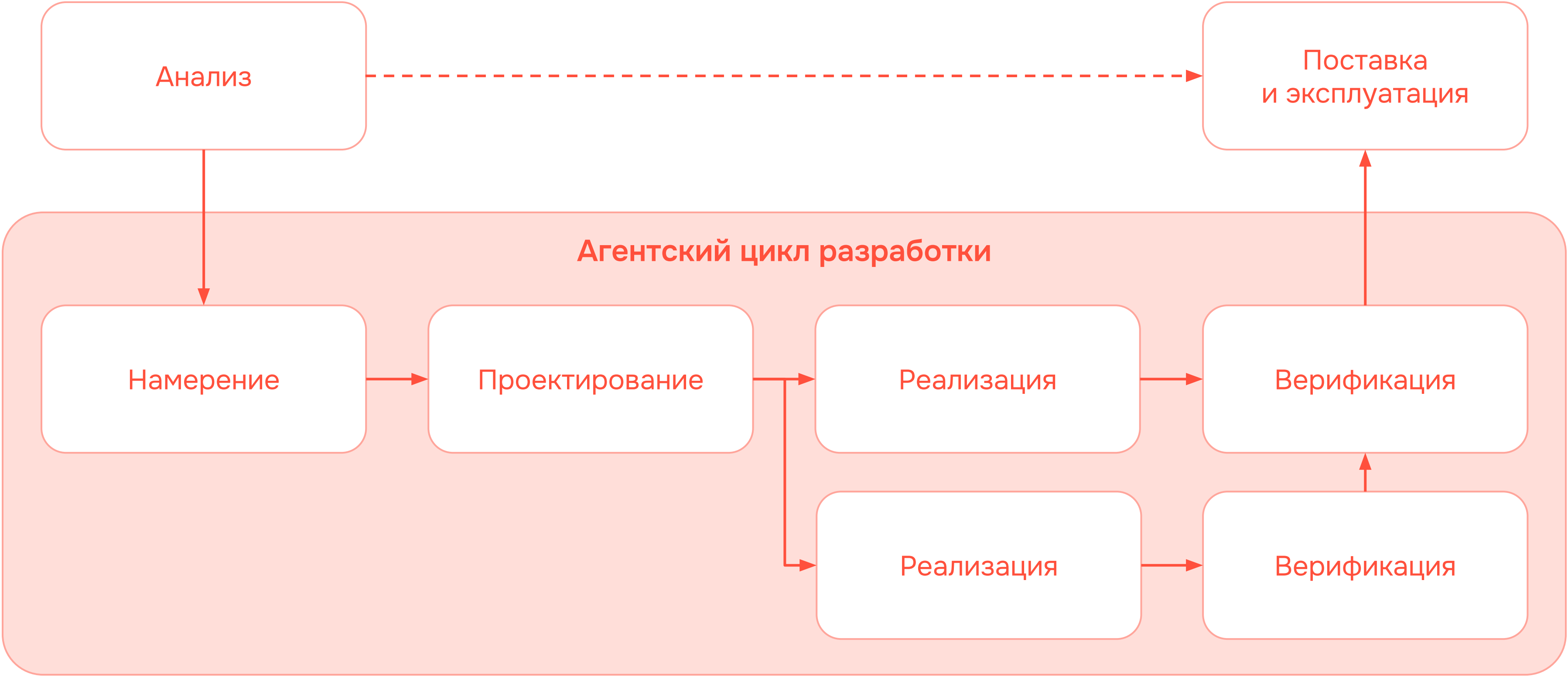
Имеет:

- Знание всех возможных паттернов
- Знание всех технологий на момент обучения
- Усреднённый индустриальный опыт

Не знает:

- Ничего про ваш продукт
- Ничего про ваш контекст
- Ничего, что не вошло в обучающую выборку
- Что такое человеческое восприятие и ощущения

ИИ-конвейер



Три направления работы

Столп 1



Компетенции
пользователя

- Работа с intent'ом

Столп 2



Capabilities

- AI Platform

Столп 3



Стандарты
и политики

- Governance

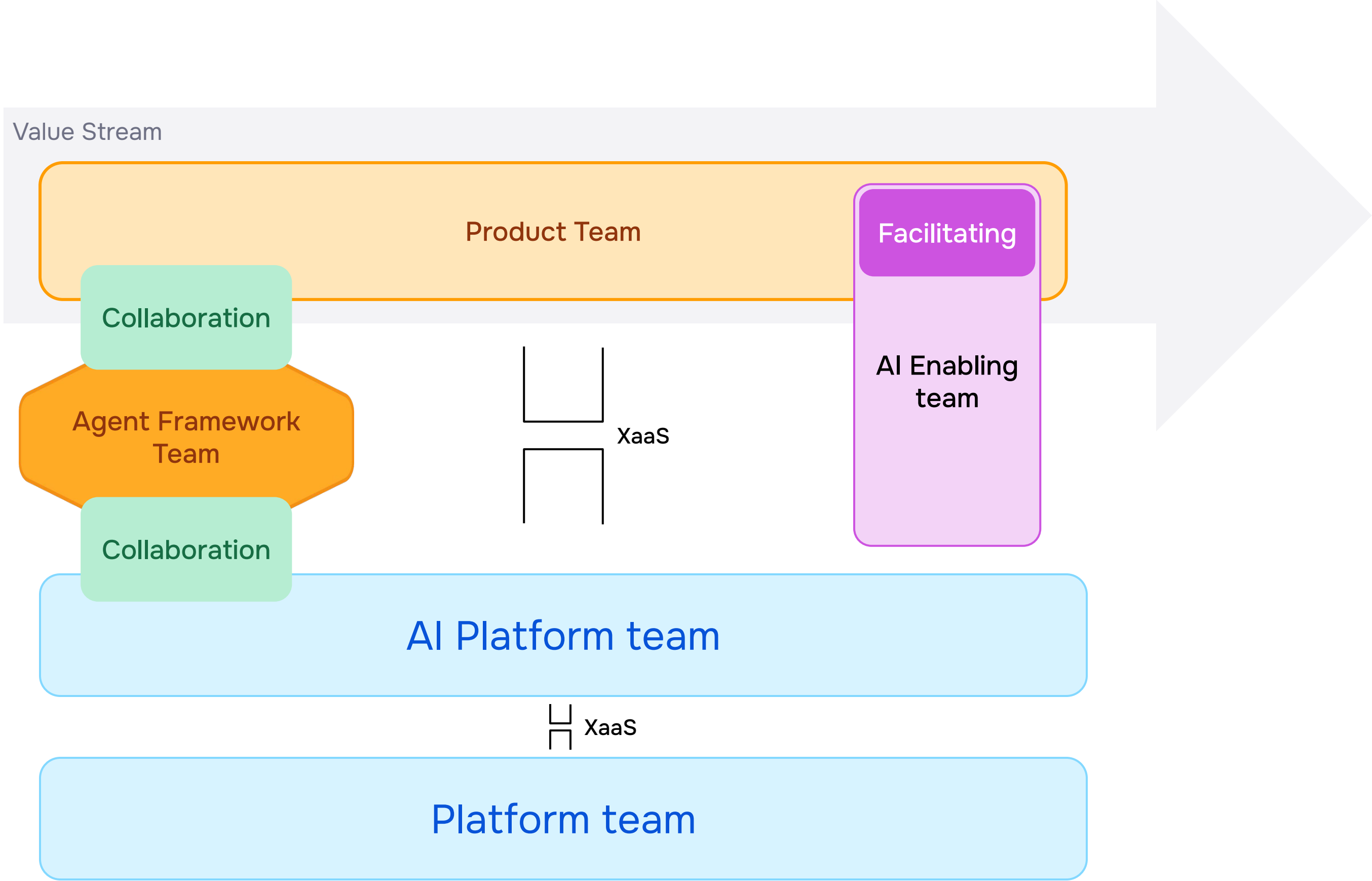
Governance

Формализация политик из документации и неявного (tacit) опыта исполнителей в машиночитаемый вид и их автоматическое применение.

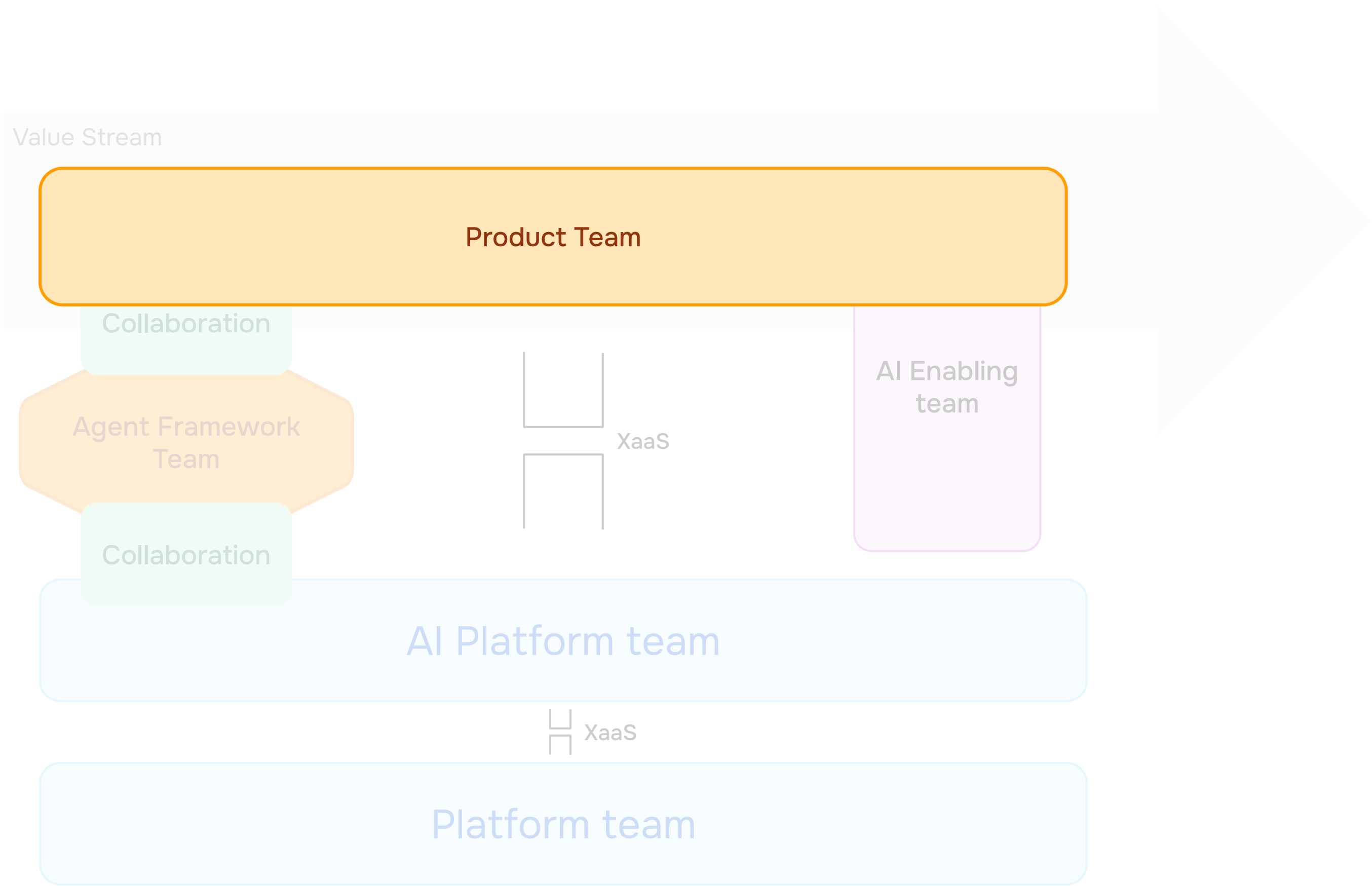
Человек не исключается из разработки, а принимает ключевые решения после автоматических проверок и сбора информации.

Вид правил	Назначение
Policy as Code	Общие политики SDLC
Architecture as Code	Архитектурные правила и ограничения
Quality Gates as Code	Проверки качества и релизной готовности
Security as Code	Политики ИБ и ограничения
Agent Rules as Code	Роль, автономность, данные, инструменты агента

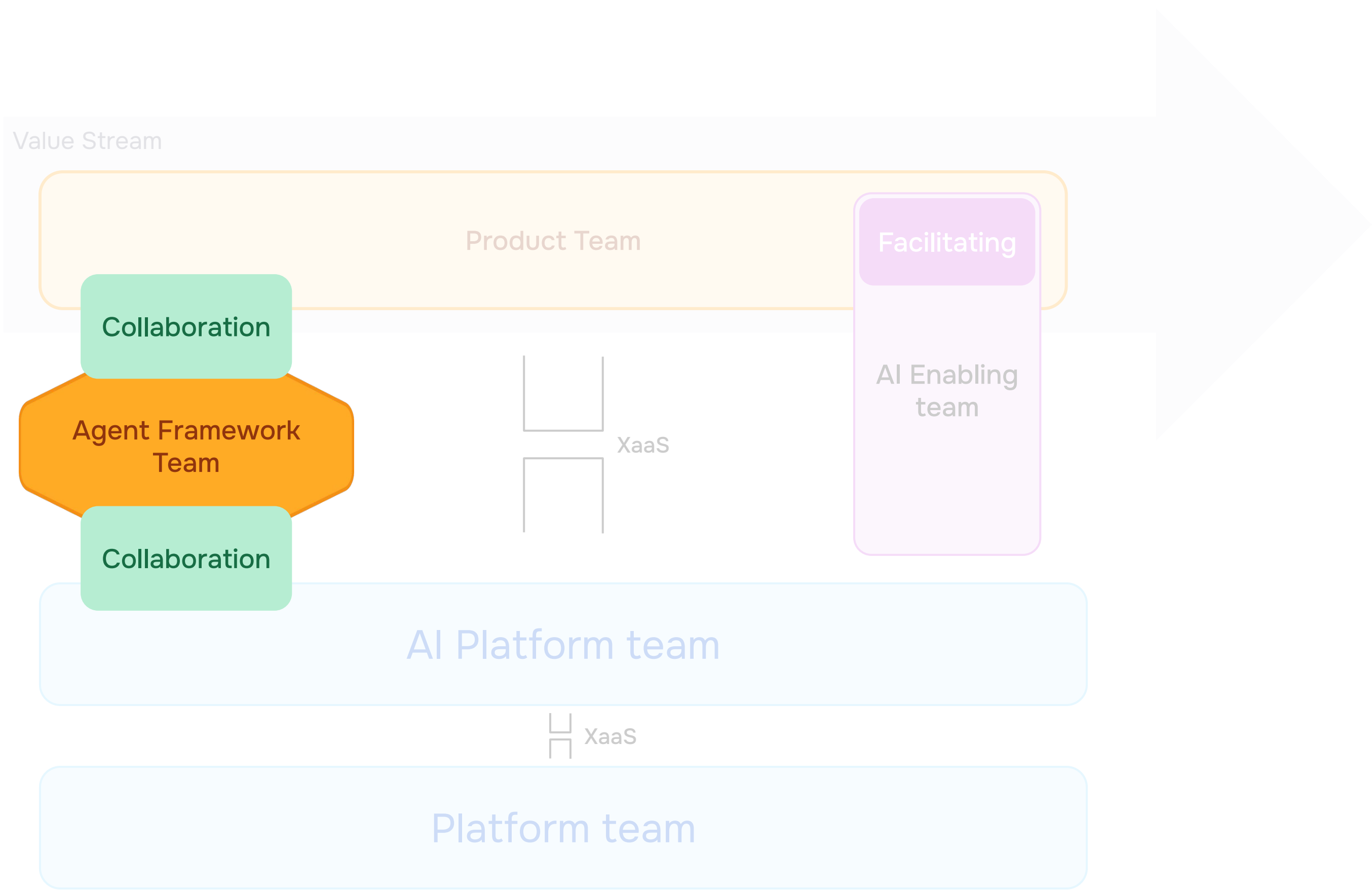
Ролевая модель



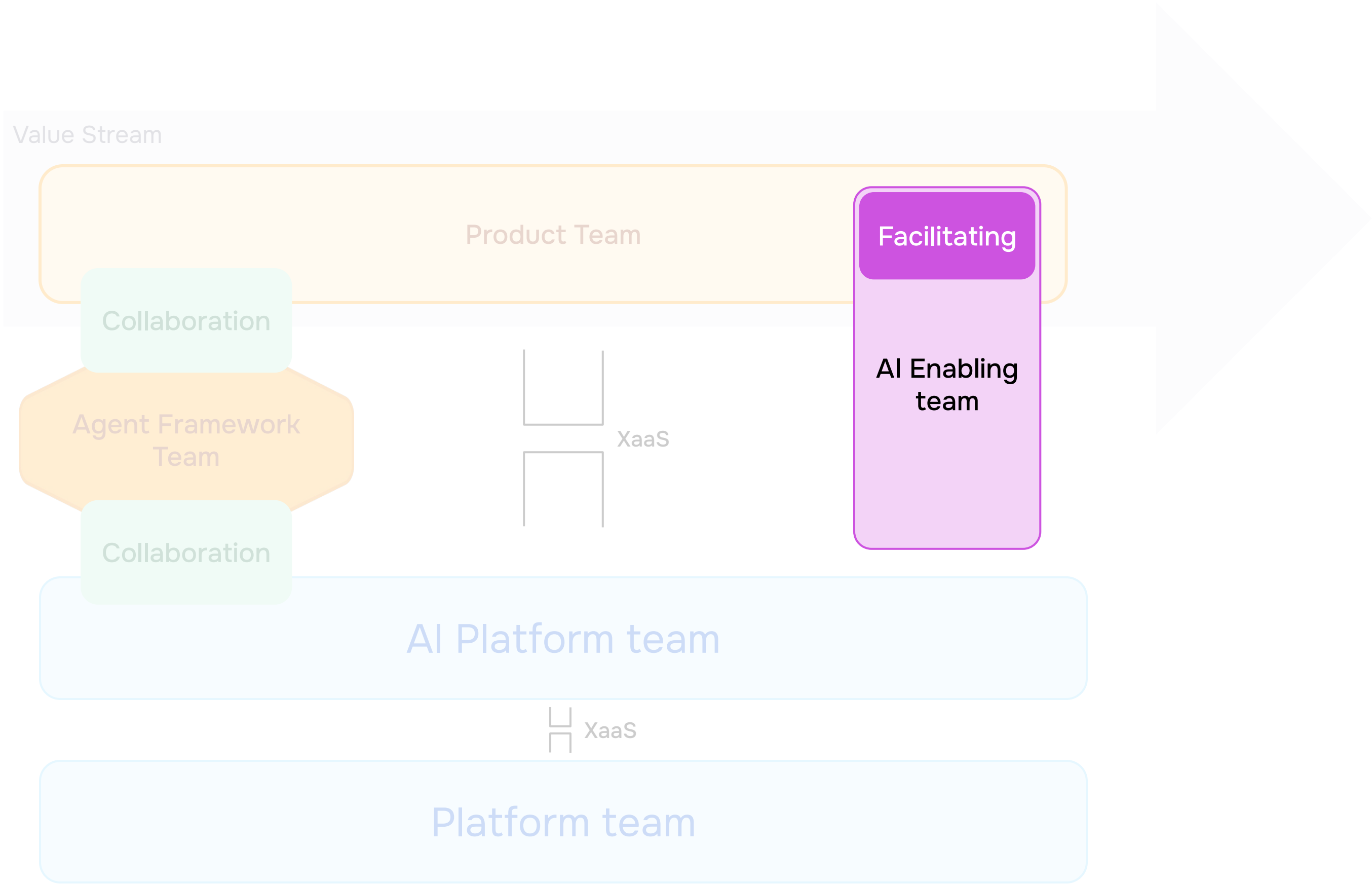
Ролевая модель



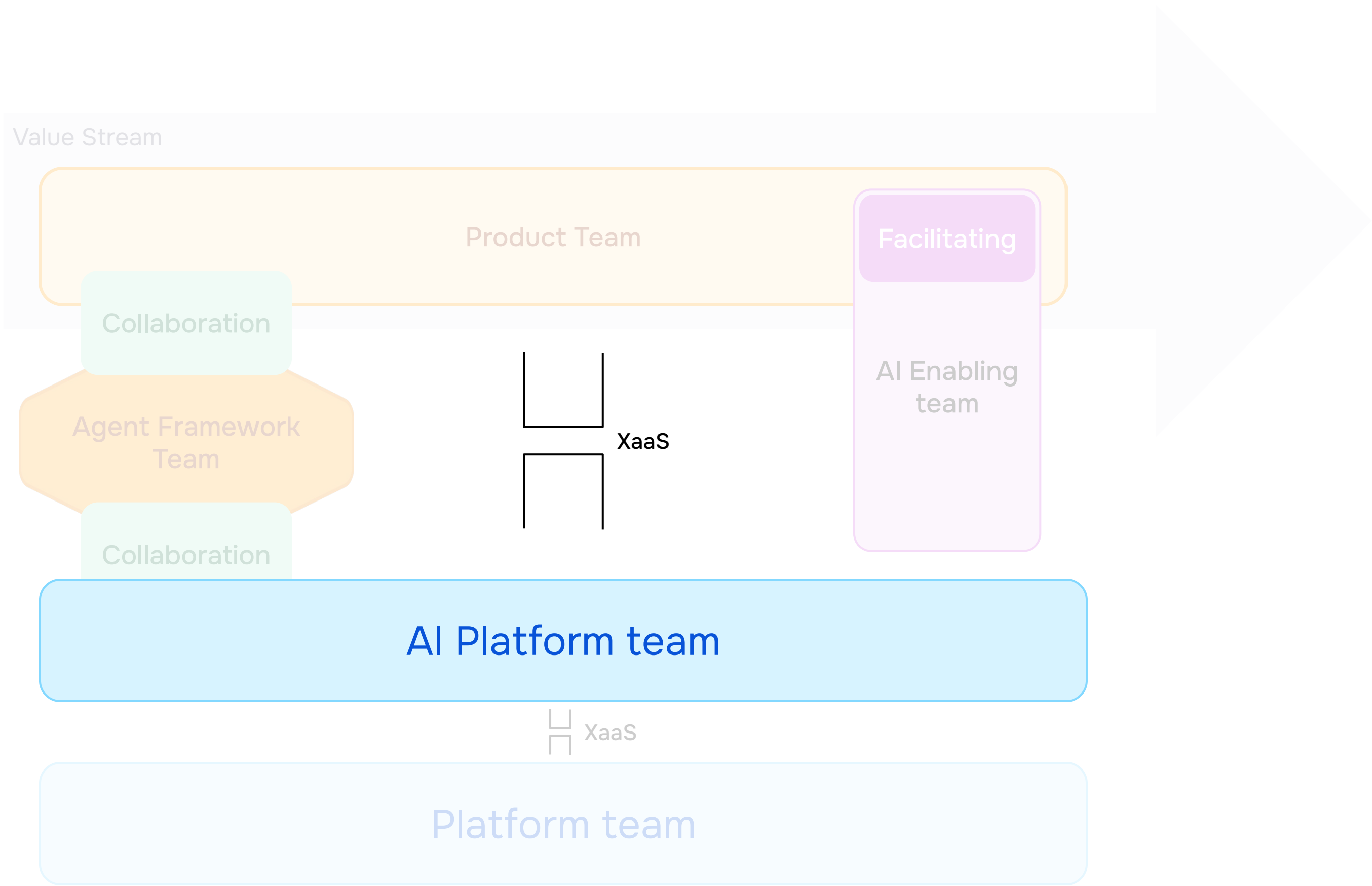
Ролевая модель



Ролевая модель



Ролевая модель



Полезные материалы

Вебинар

История одного проекта:

как встроить безопасность
по командным топологиям

Василий Шмыр
техлид по практикам DevOps



Смотреть

Распределение задач между человеком и агентами

Этап	Человек	Агенты + автоматизация
Намерение	• Формулирует продуктовую задачу • Определяет ожидаемый результат	• Помощь в формулировании требований • Валидация их корректности • Декомпозиция требований
Проектирование	• Валидация итогового решения	• Проектирование архитектурного решения • Декомпозиция на задачи • Валидация напротив продуктовых ограничений и требований
Реализация	• Валидация кода в критичных случаях и по запросу агентов	• Генерация кода и тестов • Валидация кода и тестов
Верификация	• Приёмочное/исследовательское тестирование	• Прогон тестов • Рефакторинг кода • Проверки Governance
Релиз	• Подтверждение релиза • Принятие решения об откате	• Развёртывание

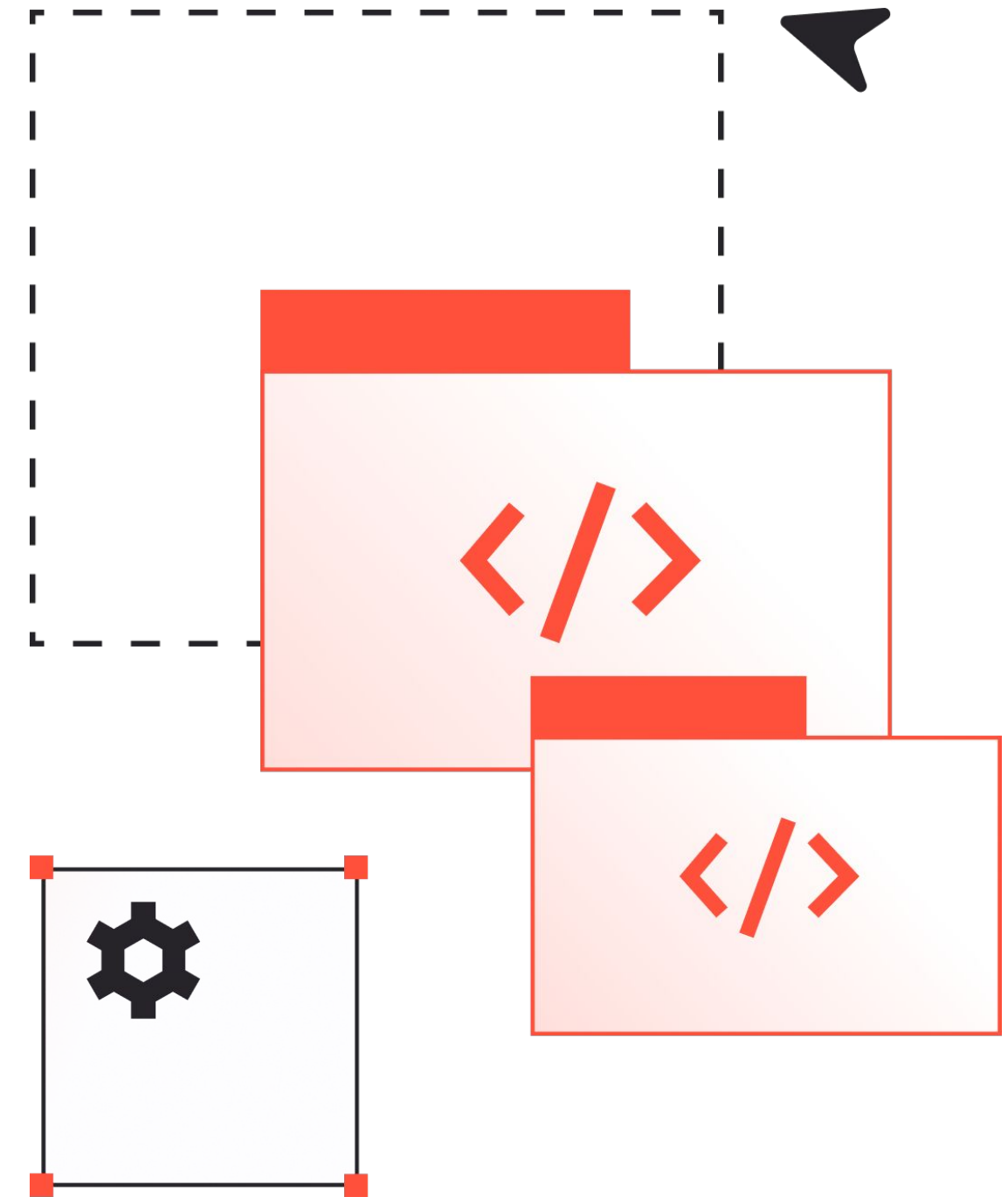
Метрики



Эффективность процесса
(DORA и др.)



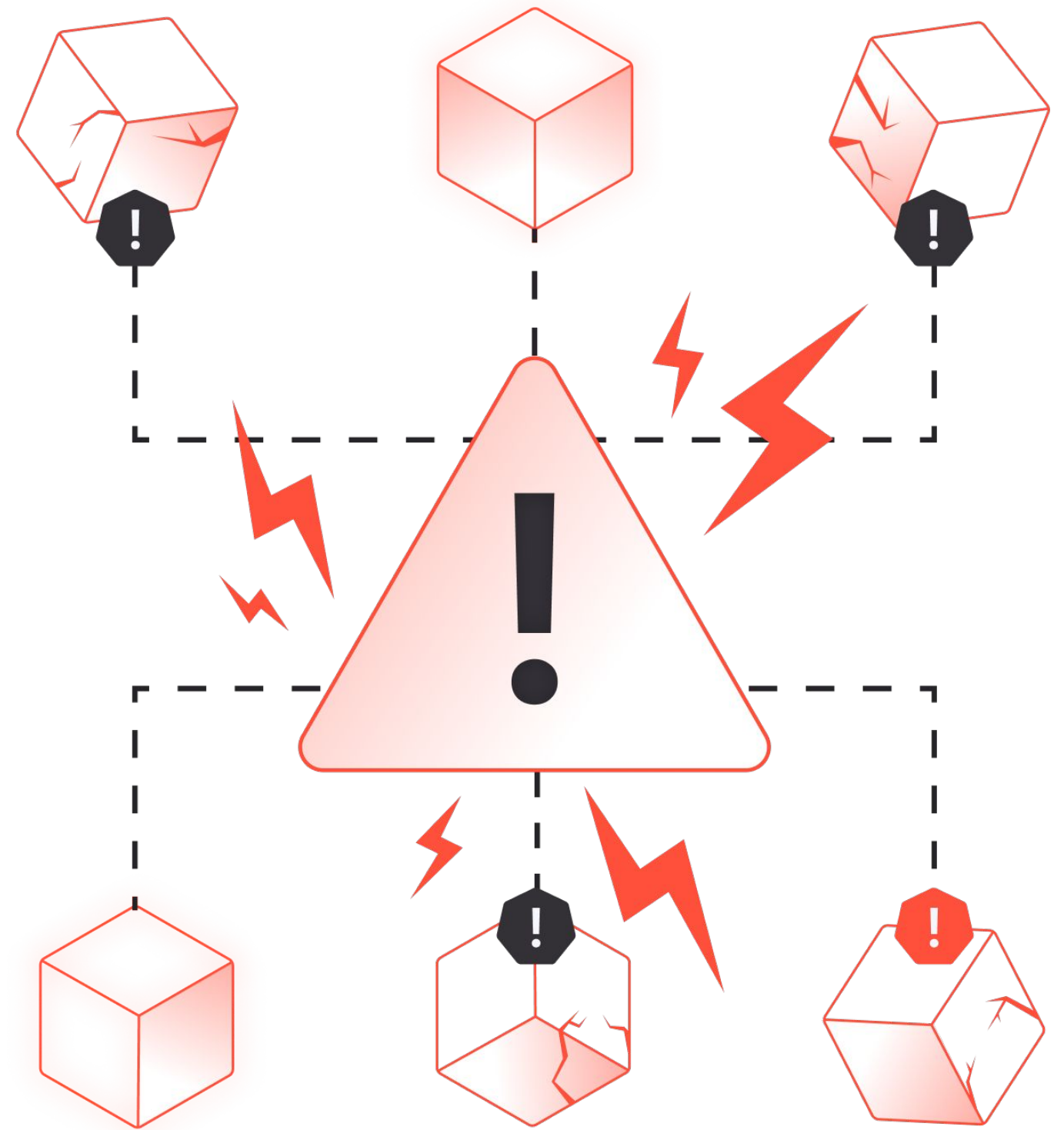
Эффективность отдельных
этапов



Риски



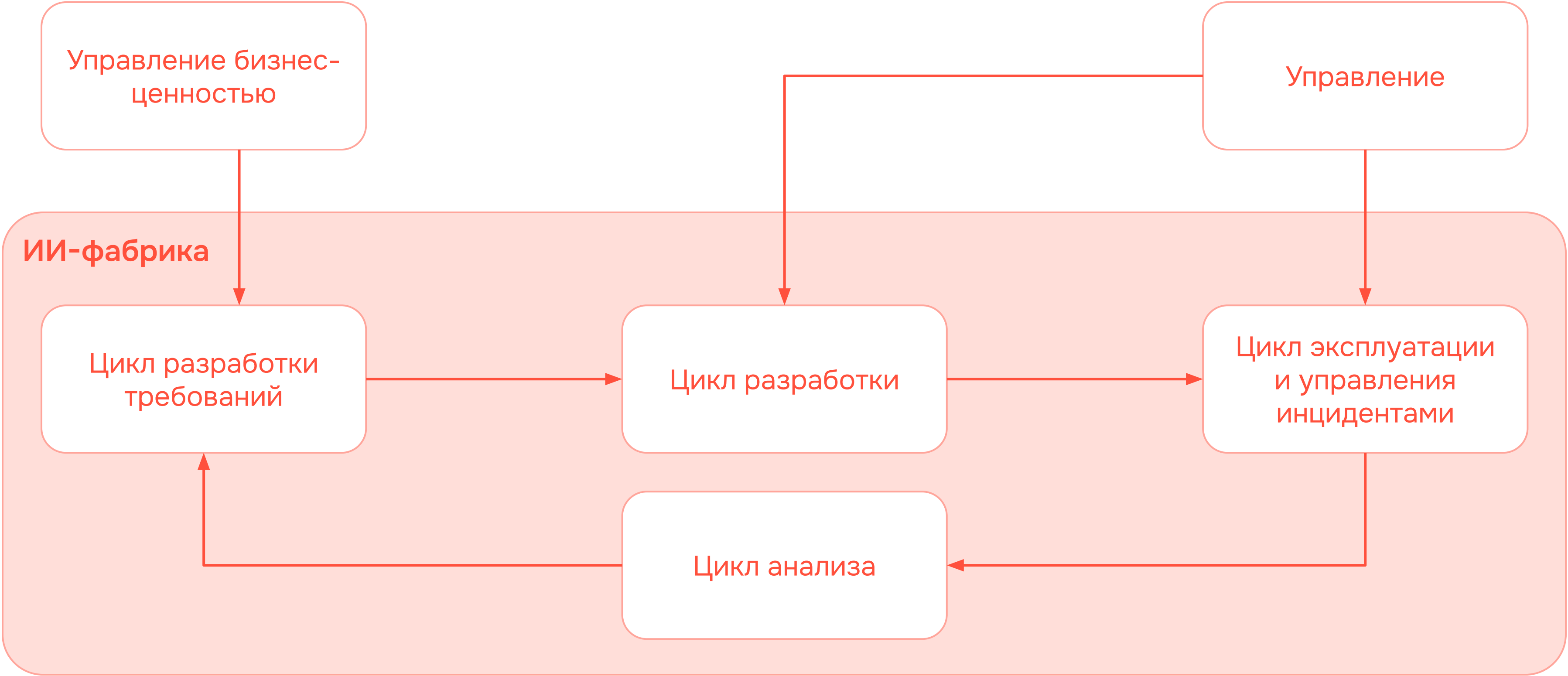
Перестаём контролировать процесс



Результат

- ✓ Конвейер масштабируется на организацию
- ✓ Эффективность разработки увеличилась
- ✓ Созданы механизмы непрерывного улучшения

ИИ-фабрика



Три направления работы

Столп 1



Компетенции пользователя

- Доменные знания

Столп 2



Capabilities

- Самообучение
- Тестирование гипотез

Столп 3



Стандарты и политики

- Расширение Governance

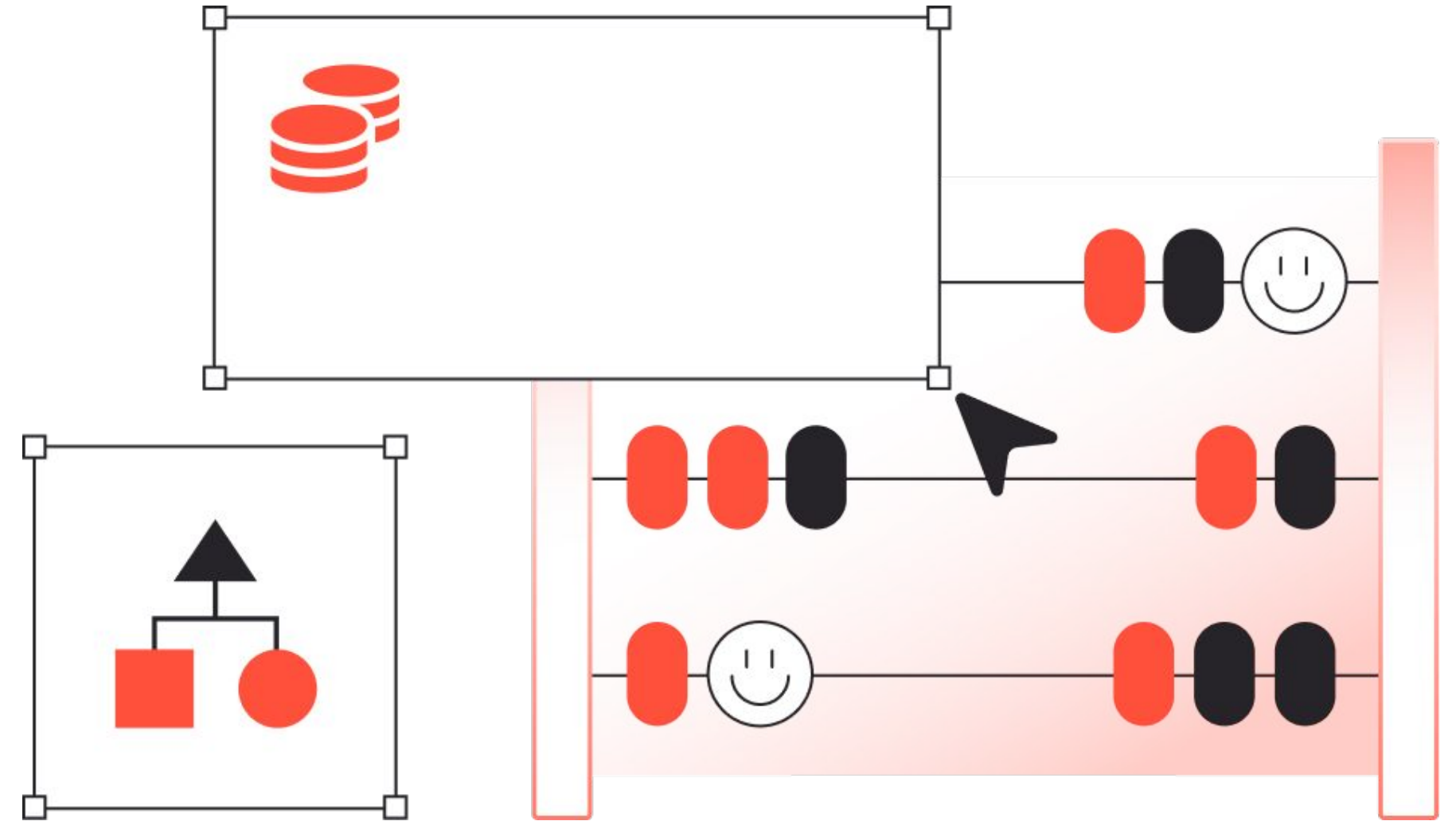
Метрики



Эффективность процесса



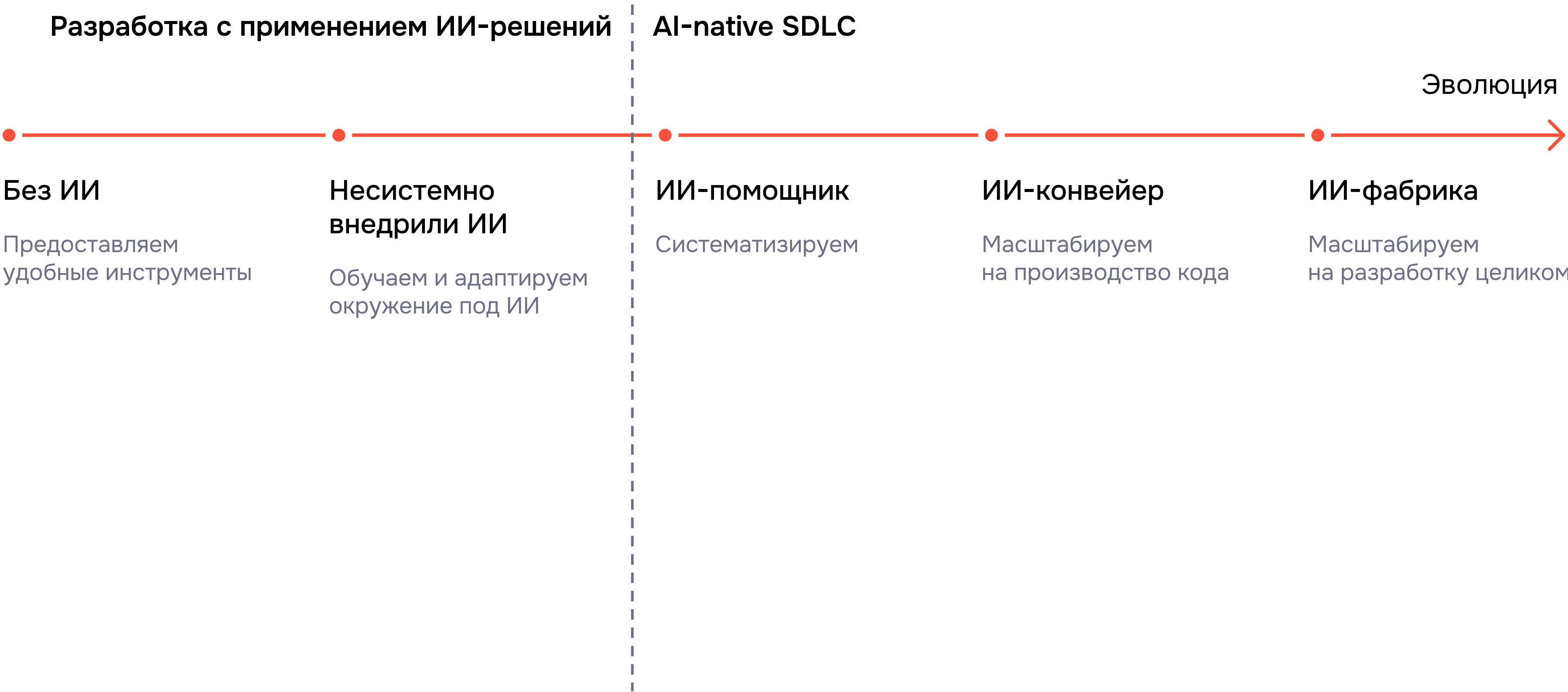
Продуктовые метрики



Результат

- ✓ Линейно масштабируемое производство

Резюме



Эволюция

Этапы	Столп 1 	Столп 2 	Столп 3 
	Компетенции пользователя	Capabilities агента	Стандарты и политики
ИИ-фабрика	Доменные знания	- Самообучение - Тестирование гипотез	Расширение Governance
ИИ-конвейер	Работа с intent'ом	AI Platform	Governance
ИИ-помощник	Prompt engineering	MCP и Tools	Верификация
Несистемно внедрили ИИ	Наработка интуиции	Индекс по знаниям	Архитектура продукта
Без ИИ	Фундамент	Корпоративное ИИ-решение / AI Platform	

Готовы перейти к AI-native SDLC?

Получите экспертную диагностику
процессов разработки:

- Анализ текущих процессов, инструментов и практик разработки
- Оценка эффективности использования AI и уровня внедрения
- Формирование целевой модели AI-native SDLC
- Подготовка дорожной карты перехода



Записаться
на диагностику

 input@express42.com 

 express42.com 