

Будущее ИТ: как ИИ трансформирует отрасль



Максим Цепков

IT-архитектор и бизнес-аналитик

Навигатор и эксперт по миру Agile,
методов самоуправления и моделей soft skill

 [mtsepkov](https://t.me/mtsepkov)  mtsepkov.org

SQA days | Москва
30-31 октября 2026

О чем будет рассказ?

Это концептуальный обзор изменений, конкретики про модели – не будет

- Что изменилось в разработке: теория против практики
- Каковы возможны новые роли для тестировщика?

ИИ-агенты и будущее

- ИИ-агенты – как их встраивать для конкретных операций
- ИИ-агенты будут работать в бизнесе – их надо обучать и тестировать
- Какие еще изменения несет будущее?



Про создание тестов ИИ-агентами было много кейсов на SQAdays. Часть из них – в моей статье [«Об организации труда ИИ-агентов»](#), там же есть кейсы с других конференций (но материал устаревает)

Немного обо мне

Создание и внедрение больших корпоративных систем (30+ лет)

- Знание практик операционного управления и ведения проектов в коммерческих и государственных организациях и банках
- Опыт управления проектами в IT: от инженерного подхода и PMBOK – к современным Agile-методам (с 2007 года)
- Опыт перестройки организаций при внедрении систем

Навигация в менеджменте цифрового мира

- Agile и самоуправление: бирюзовые организации, холакратия и социократия ([книга, статьи и выступления](#))
- Модель [спиральной динамики](#) (с 2013) и другие [модели soft skills](#), [модели личности](#) ([книга](#)) и [самоопределения](#) ([книга](#))

Корпоративная культура Китая – лидера новой волны технологических революций ([книга](#))

На моем сайте mtsepkov.org – выступления, блог и много другого





ИТ-разработка: логика изменений

Как приходят новации?

Типовой сценарий освоения новой отрасли

- Новая успешная практика: ИИ-сопилот разработчика, вайбкодинг бизнеса
- Для нее достают старую теорию: SDD, Disrupt PDCL – она работает плохо
- Сквозь теорию прорастает успешная практика, она – другая

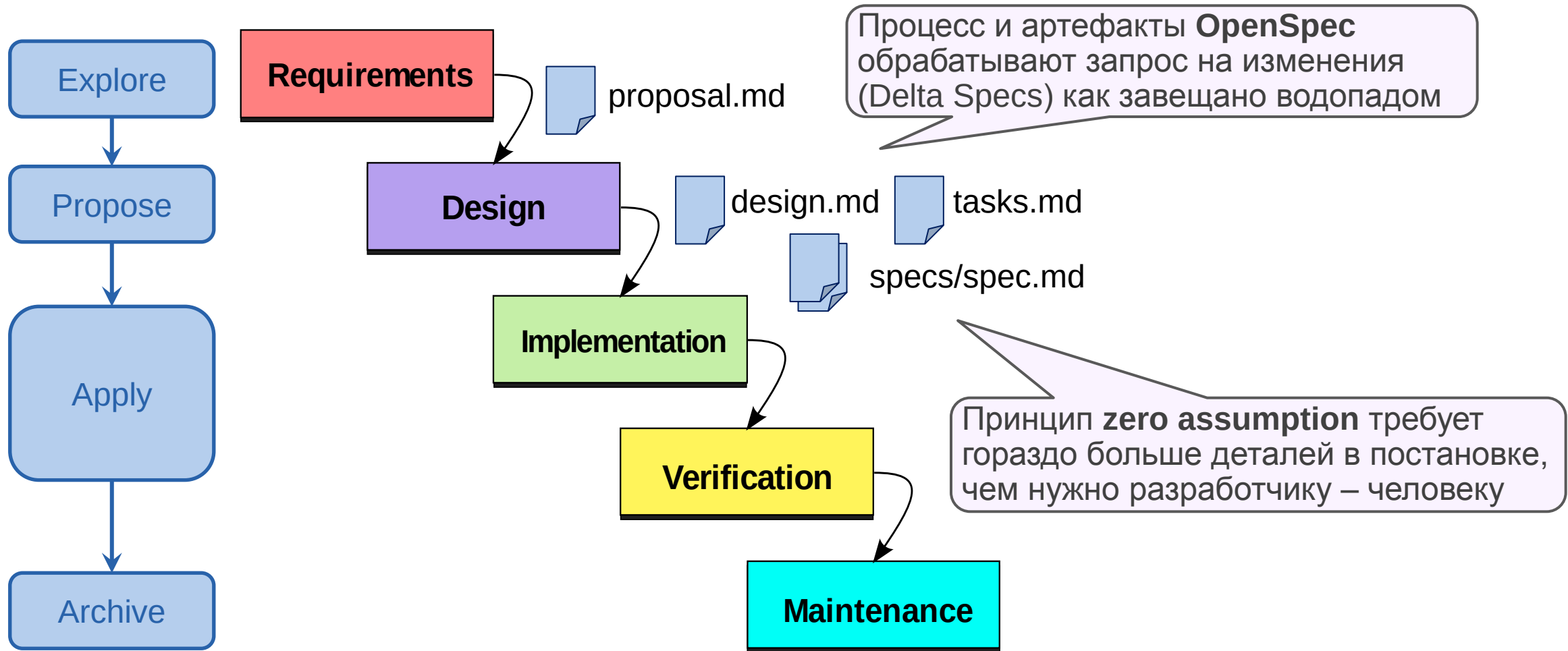
А что происходит на уровне компании?

- Небольшим компаниям проще: драйвер приносит новое в коммуникации
- В крупных – три варианта: инициатива, организованные пилоты или внедрение по площади, у каждого – свои ограничения и преимущества
- **Внедрение по площади не ускорит бизнес** – узкое место будет последним



Стоит понимать, что происходит в компании и отрасли в целом – и **строить свой путь активно**, а не быть жертвой ситуации

SDD – очередная инкарнация водопада/RUP



Модель водопада – http://en.wikipedia.org/wiki/Waterfall_model

Проблема внедрения SDD – проверка спецификации

Если в спецификации технический дизайн, то для получения хорошего результата нужно много деталей – и **она становится плохо проверяемой**

- Ее точно не может проверить бизнес и она ему не нужна
- Ее тяжело проверяют сами разработчики
- Проверка не гарантирует, что будет сделано то, что нужно бизнесу
- Проверка не гарантирует, что в коде будет так, как требует спецификация
- У разработчиков теряется мотивация – они учились писать код, а не проверять технический дизайн

Опыт Райффайзен: **спецификация должна снижать неопределенность намерения, а не давать технический дизайн** ([Андрей Усенко на AI native](#))

Альтернативы SDD – есть!

TDD/BDD: тест-кейсы вместо технического дизайна

- Acceptance criteria и тест-кейсы проверять легче
- Бизнес их понимает и согласен обсуждать сложные случаи
- Тест-кейсы могут описывать агенты, нужны критерии хороших сценариев
- Вместо дизайна – задаем архитектуру и шаблоны, дальше – агенты
- В работе агентов полезно проверять **точки выбора и принятия решений**

Виталий Левченко «[SDD, или Как кратко ускорить разработку с LLM](#)»
(Codefest-2026): в названии SDD, но по сути в описании – архитектурные принципы и тесты, по ним спецификацию и код итеративно доводят агенты



Тестировщик может стать прикладным разработчиком: acceptance criteria, тест-кейсы и пирамида тестирования ему знакомы, если он писал автотесты – разбирается в коде, а навыки аналитика надо освоить, если их нет

ИИ меняет обработку типовых задач – рельсы

- Таких задач много: сайт-визитка, типовой интернет-магазин, типовой документооборот, типовая CRM небольшого бизнеса
- Раньше делали базовое решение с точками кастомизации, ее делала пара аналитик и разработчик, иногда докручивая базу
- Теперь можно иначе, с помощью ИИ
 - Создаем harness для скелета решения
 - Аналитик может вайбкодить специализированное решение
 - А разработчик смотрит, где в решениях возникают проблемы и докручивает harness
- Есть ряд кейсов, например, на Merge-2026 рассказывали: в компании enterprise разработки из 10 проектов разработчики остались на двух, остальные вайбкодят аналитики с прежним качеством и темпами, руководитель держит руку на пульсе и докручивает harness
- **Сложность таких решений ограничена, но гибкость – увеличилась**



Место тестировщика
надо искать

Разработка через прототипы

Если бизнес не готов формулировать, что нужно – надо делать прототипы

- Прототипы показываем и доводим, их разработка ускоряется кратно
- По мере стабилизации – формируем архитектурные описания и тесты

Кейсы:

- Андрей Сотник и Родион Лунев «[AI-помощник. Как эксперимент вышел из-под контроля](#)» (Saint Teamlead-2026) – изготовление прототипов в X5-tech
- Вадим Герасимов «[От хаоса к системе: Почему LLM не отменяет анализ, а требует его эволюции](#)» (AnalystDays-2026) – создание небольшой CRM

? Место тестировщика в таком процессе надо искать...

Бизнес сам вайбкодит прототипы, ему – нравится!

- Вайбкодинг доступен бизнесу и он приносит прототипы вместо идей
 - С точки зрения бизнеса, задача – сделана, осталось запустить на прод
 - Бизнес не готов обсуждать постановки, а поведение в сложных случаях – готов
 - Тестировщик может попробовать подхватить такую разработку от бизнеса
- **Круглый стол по вайбкодингу на секции маркетинга на Merge-2026:**
 - Легче самому за пару вечеров сделать, чем две недели объяснять разработке
 - ИИ понимает лучше и не сопротивляется, когда требуешь переделок
- **Никита Худов** на Alnative-2026 [«Агенты как часть штата с первого дня: роли, найм и управление при запуске продукта»](#)
 - Два основателя создали продукт и запустили вайбкодингом агентов за три месяца
 - Такая разработка стоила бы 6-8 млн, а обошлась в 100-300 тысяч, включая их время
- Бизнес ждет, что ИТ сможет подхватить сделанное и развивать дальше, это изменит процесс, хотя пока не осознано – тут есть место тестировщику

Бизнес не сможет вайбкодить все

- Развитие этой идеи – **Disrupt-PDCL** от Сбера: разработчики делают платформу, на которой бизнес сам делает приложения
- Применение – Андрей Неведин из Райффайзенбанк [«Context is a must: как мы системно управляем контекстом»](#) (Saint Highload-2026)
- Подход переносит на ИИ-агентов старую идею CASE-средств: разработка создает платформу, на которой бизнес делает приложения
- У идеи те же ограничения: платформа будет слишком сложной, и при этом все равно не поддержит необходимые кейсы, решения для которых надо искать, **решая задачу совместно с бизнесом**

Вайбкодим проект, а что с тестами и документами?

- Описание архитектуры необходимо, но может появляться не сразу
- Acceptance criteria и тест-кейсы тоже необходимы, но тоже не сразу
- Агенты хорошо порождают описания и тесты, если есть чек-листы качества
- **Задачи тестировщика:** сформировать пирамиду и политику тестирования, сделать mcr, чтобы агент тестировал, обучить его ставить инциденты..
- Должен быть способ в частном решении контролируемо нарушить правила, чтобы провести эксперимент или расширить функционал
- Нужен отдельный процесс: **мониторинг качества решений**, которые обеспечивает harness, для **его развития** – доработка шаблонов и правил

ИИ в разработке – лишь начало изменений

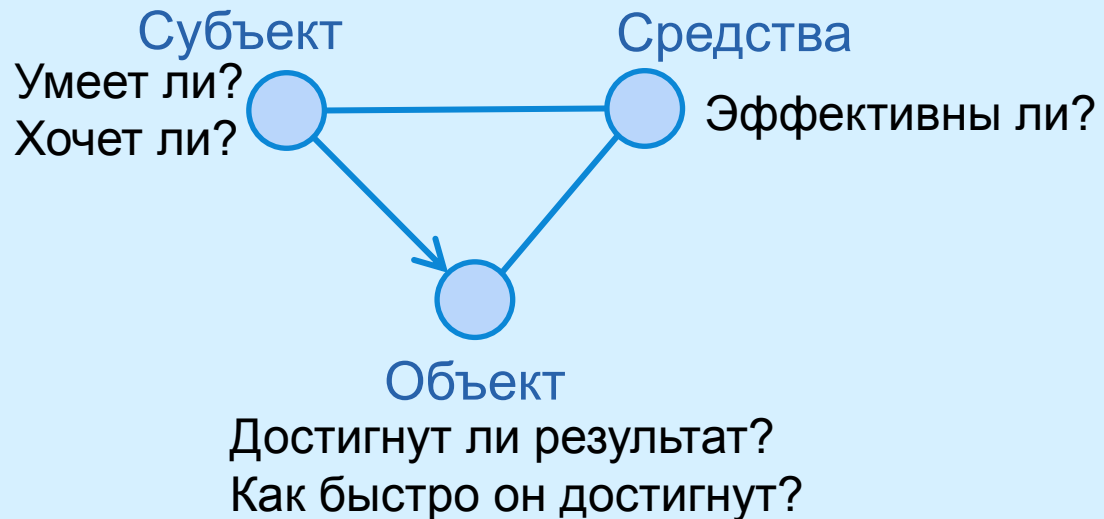
- Разработка перестала быть ограничением – надо работать с новой точкой
- Ограничением стала работа владельца продукта по созданию гипотез
- Если гипотеза проверяется за 3-5 дней вместо 2-3 месяцев, **надо перестраивать весь процесс** и роль ИТ в нем
- Раньше бизнес говорил: «куча идей, разработка не успевает», теперь вопрос к **качеству идей**
- Работу владельца продукта ИИ-помощники ускоряют меньше – сейчас нарабатывают **технологии передачи discovery агентам**: гипотеза, проверка проблем, целесообразность решения
- Для проверки бизнес может и будет вайбкодить прототипы – это быстрее, чем доносить идею до команды, и надо **уметь прототипы подхватывать**



Система разделения труда

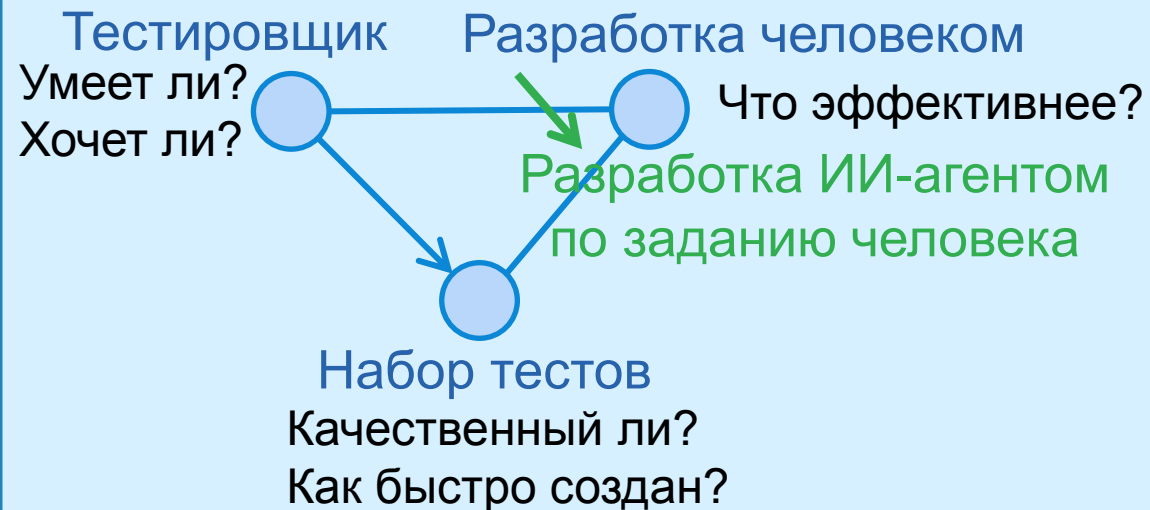
Субъект – Объект – Средства

Выполнение операции



Вопросы определяют метрики, с помощью которых мы проверяем результативность и эффективность выполнения операции

Создание тест-кейсов



Сменив средство – передав разработку тест-кейсов ИИ, мы повлияли на мотивацию и способность тестировщика – как изменилась скорость и качество?

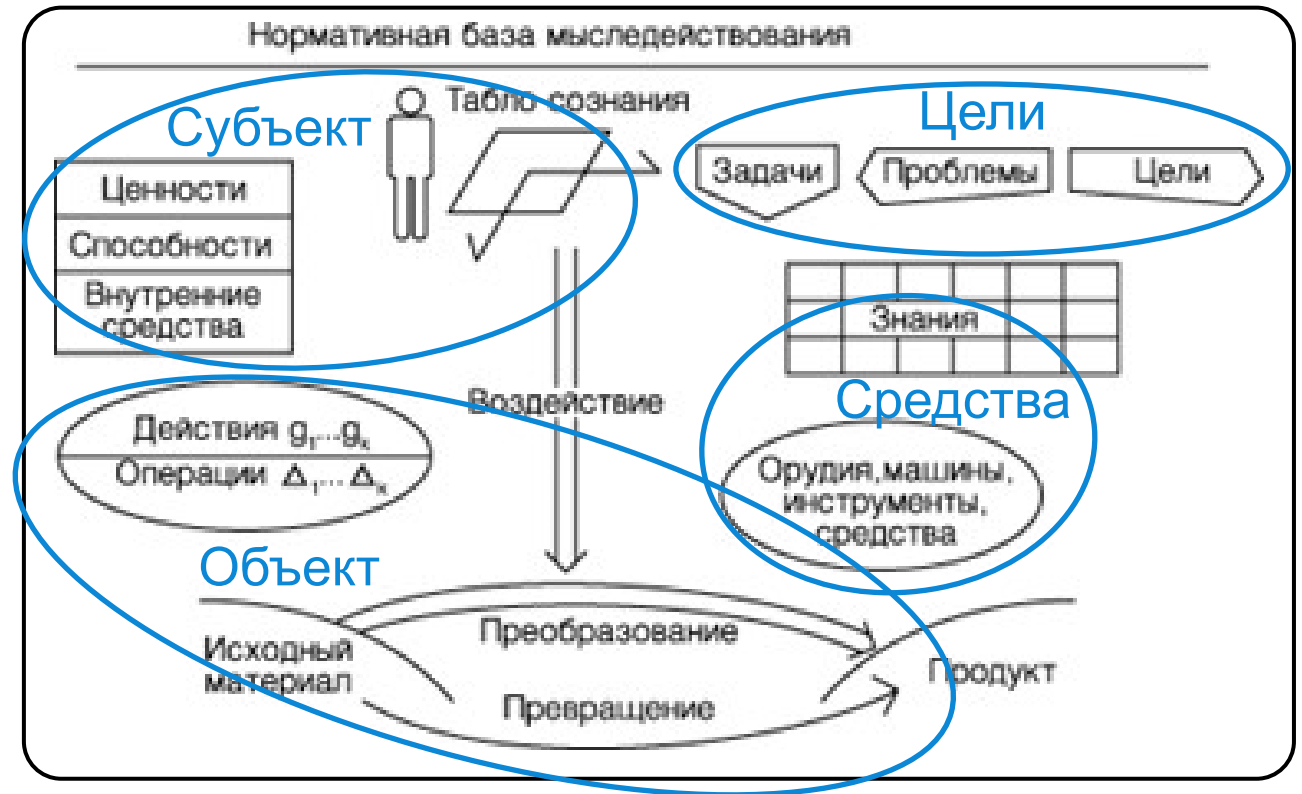
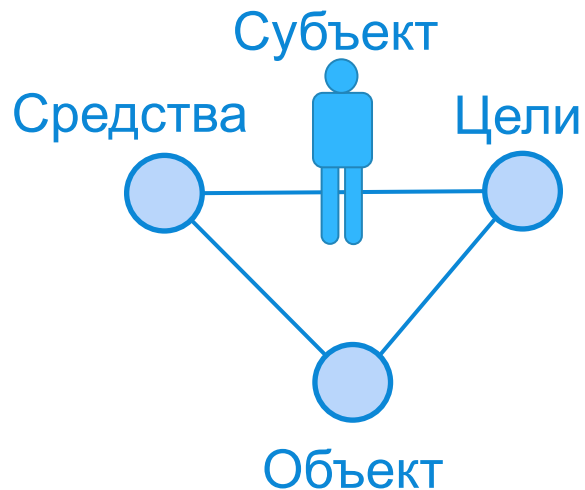


Средство – не только LLM, среда разработки или фреймворк, но и **знания**, каким именно образом их использовать.

А для ИИ-агента – обеспечение его знаниями о проекте и методах работы (harness)

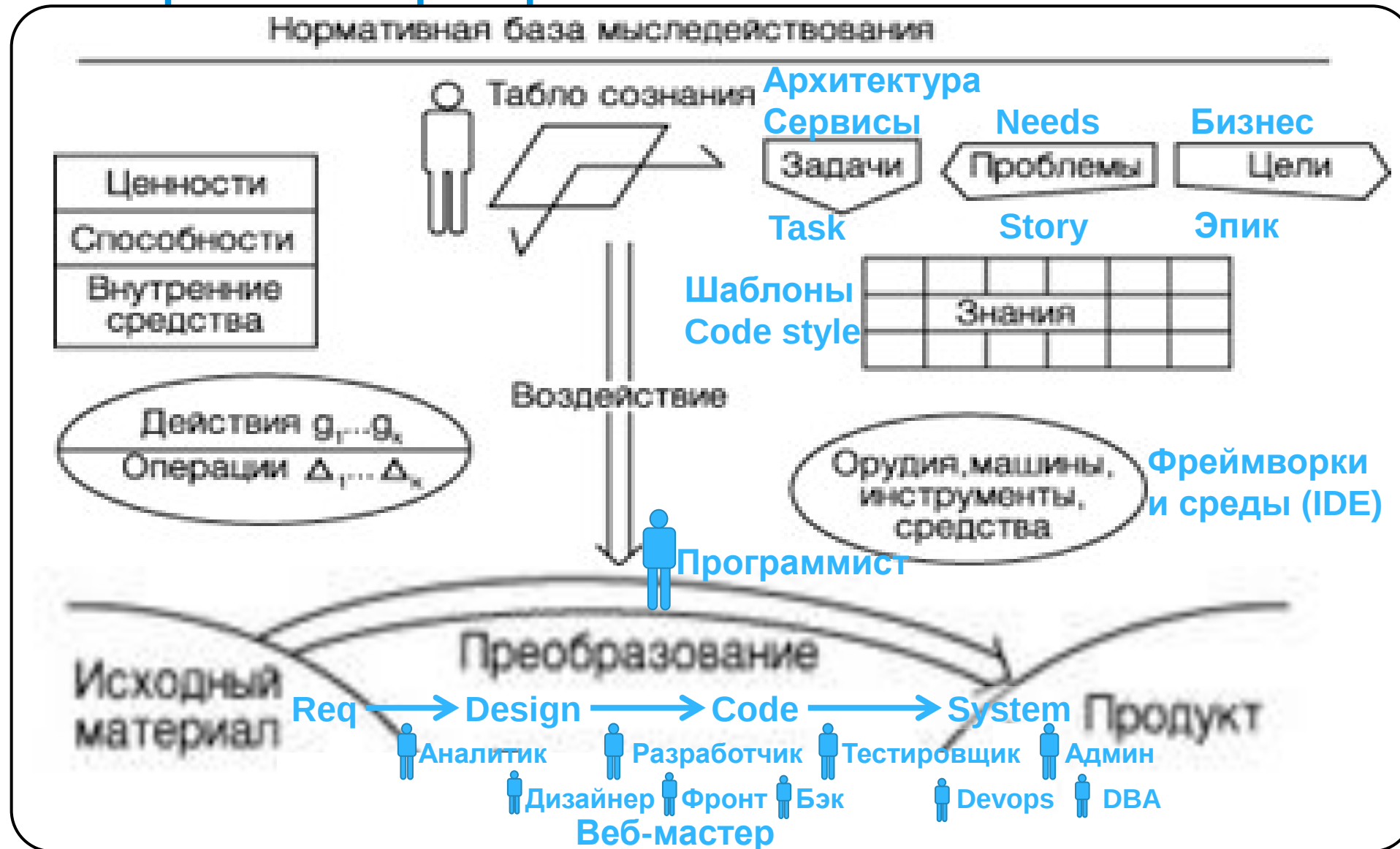
Схема Акта Деятельности (СМД-методология)

Различает структурные элементы Деятельности, это позволяет ее понимать и конструировать



Развернута от тройки **Цели – Объект – Средства** в лекции 3 серии **Петра Георгиевича Щедровицкого** в МИСИС PUZZLES: Как современный управленец формирует картину мира: [pdf \(fondgp.ru\)](http://fondgp.ru), [youtube](https://www.youtube.com), [rutube](https://rutube.ru)

История ИТ-разработки

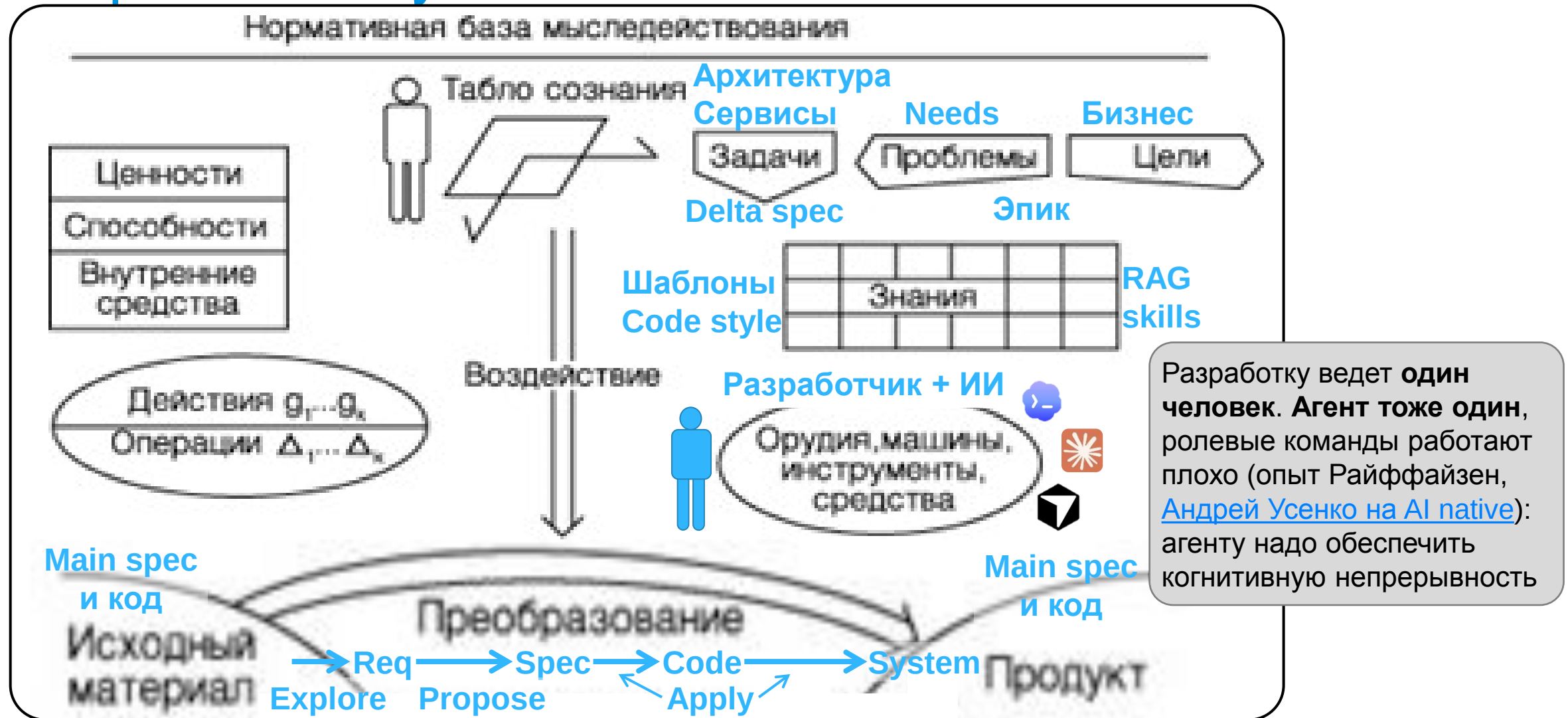


Два разделения труда: division и sharing

- Мы можем делить труд на операции, а можем объединяться для их совместного выполнения
- Объединение важно, если **есть большая неопределенность** в решении, тогда бизнес (аналитик) и разработчик совместно ищут решение
 - Вместо сложной системы шаблонов документов – каталог образцов для копирования
 - Вместо конструктора отчетов – средство легко вывести SQL-запрос на интерфейс
- Кроссфункциональные команды и T-shape – про sharing вместо division
- Универсализм ИИ-агентов позволяет делать **tiny-team**: бизнес + технарь, только для них надо обеспечить bus factor больше 1
- Набор компетенций сильно изменяется

Тестировщикам предстоит искать свое место в новой разработке

Разработка с участием ИИ



Меняется процесс – меняются компетенции людей

- Проверка спецификации – совсем не то, что написание кода: писатель редко бывает хорошим редактором и наоборот
- Компетенция **быстрого кодирования прототипов** раньше была отдельной, теперь большинство работают в близком режиме
- Агенты работают долго (5-10-20 минут), на это время **человек переключается** на другую задачу – это тоже новое требование **для всех**
- Требования должны быть разумны: заранее оценить, что делает ИИ, а что лучше сделает разработчик – отдельная компетенция



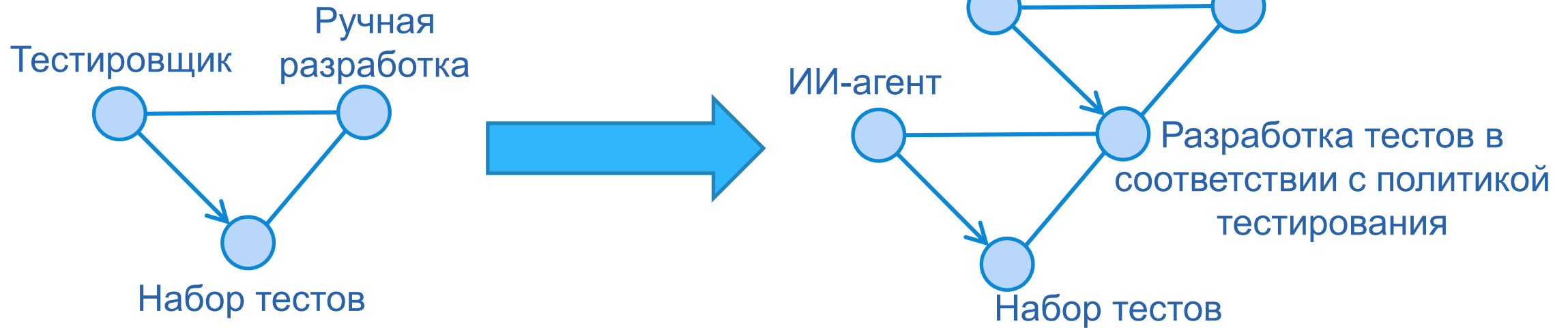
Помните: человек не может все.

Если бы разработчик или тестировщик мог быть предпринимателем, руководителем компании – он бы не работал разработчиком

Лог работы позволяет совершенствовать процесс

- Работа сотрудников с агентами дает лог реальной деятельности, гораздо более подробный чем раньше в task-трекере, мессенджерах и почте
- Такой лог дает возможность **анализа и совершенствования процесса** и конкретных сотрудников, показывая точки возможного развития
- Об этом – интересное выступление Авенира Воронова [«AI-копилоты управления разработкой»](#) (Saint Teamlead-2026)
- Это – новое направление в системе вертикального разделения труда
- Но опыт говорит, что использовать будут не продуктивно, для контроля

Вертикальное разделение труда



- Создание тестов выполняет ИИ-агент по политике тестирования
- Политику пишет тестирующий, используя HADI-цикл: оценивает текущую работу и ставит эксперименты, накапливая массив задач для отладки

Вертикальное разделение труда – создание знаний для обеспечения выполнения отдельных операций, а также разделение работы на операции

В ИТ это погружено внутрь команд, а не выделено на позицию технолога



Организация труда ИИ-агентов

Включение ИИ-агентов

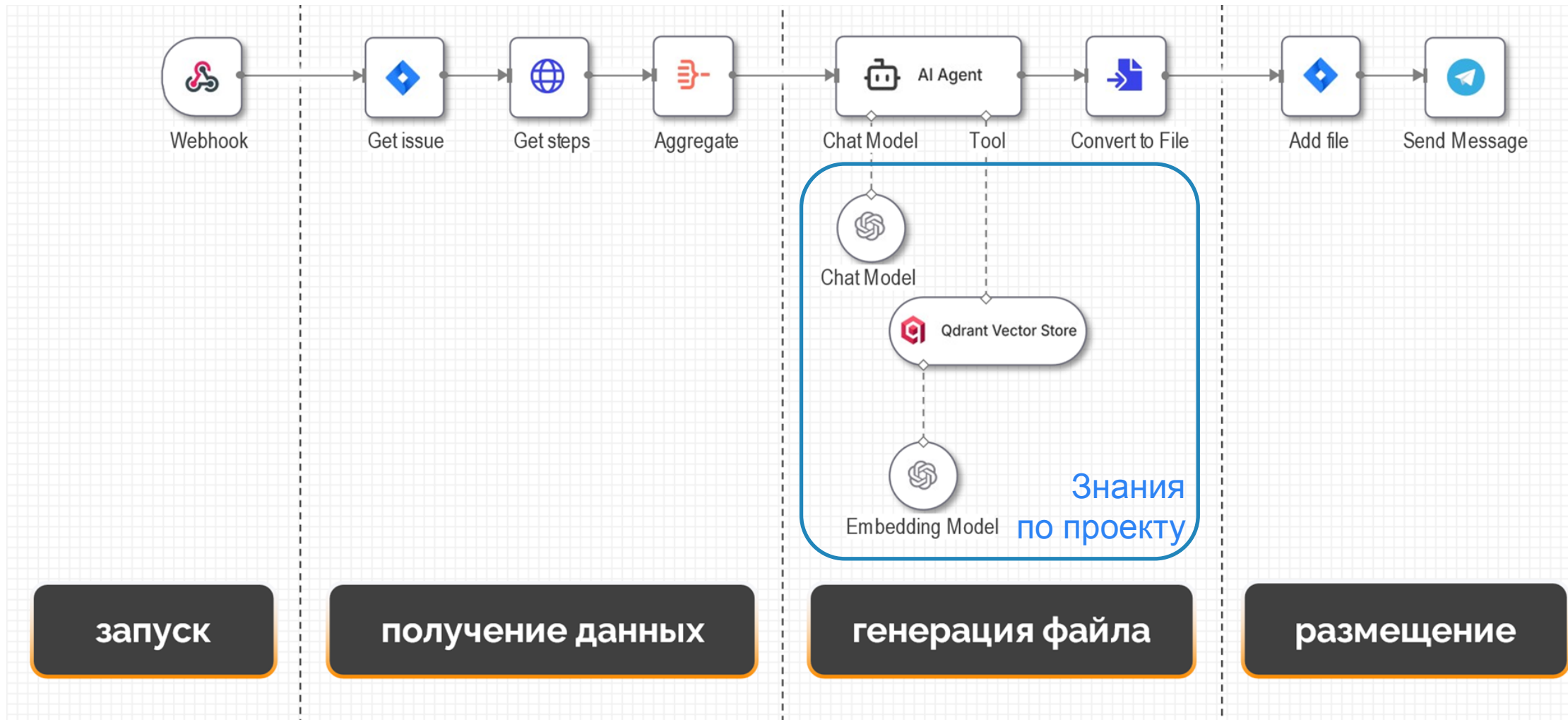
- ИИ-агенты классно делают точечные задачи – но процесс становится сложнее, в нем больше шагов
- **Описания pipeline как workflow задач в Jira** – недостаточно, нужны средства типа **n8n** с деталями и высокой формальностью
- Агентов надо снабдить информацией, это – не только принципы и архитектура, но и **живое знание о ходе проекта** и его коде
- Доступа к коду и записям встреч – недостаточно, надо проектировать **систему артефактов** и поддерживать структурные описания в ней
- Нужен процесс **мониторинга качества решений**, которые обеспечивает harness и его развитие (HADI-цикл), но затраты на это необходимо ограничивать – для всего этого **у тестировщиков есть навыки**
- Все это касается не только разработки: бизнес будет использовать агентов в своих основных процессах, и их работу так же надо поддерживать

Quality gate или барьер для разработки?

- ИИ-агенты хорошо делают проверки по чек-листам
- Проверки могут обеспечить качество кода, соответствие архитектурным принципам, требования безопасности, качество документации и так далее
- Проверки по чек-листам частично выносим в автомат, кодируемый агентом (правила для linter и т.п.) – это дает жесткость проверок
- Важно предусмотреть гибкие механизмы для особых случаев – раньше это обеспечивала коммуникация с ревьюером
- Нужен отдельный процесс: **мониторинг качества решений**, которые обеспечивает harness и его развитие – доработка шаблонов и правил

ИИ-агента надо снабдить знаниями

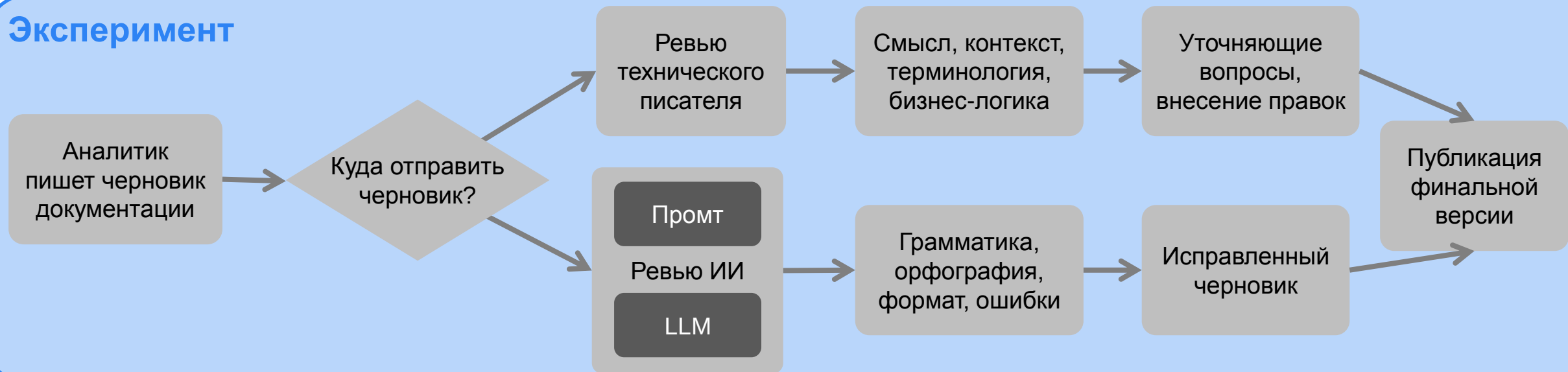
Александр Ткачев «No-code/Low-code в тестировании: ускоряем процессы с помощью n8n» (SQAdays-2026)



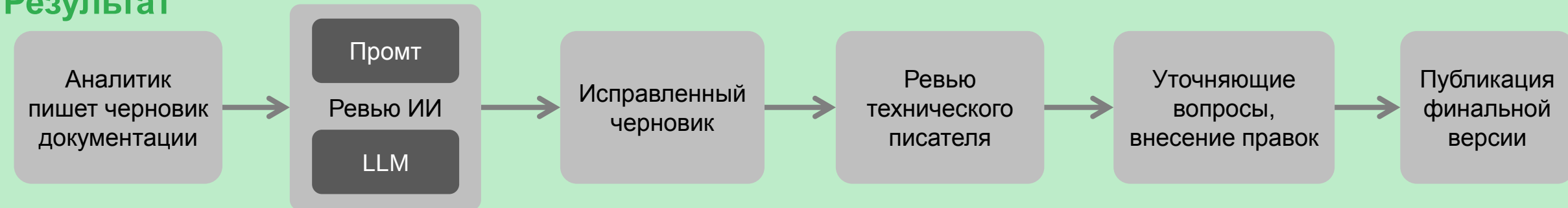
Надо адекватно оценить ВОЗМОЖНОСТИ

Камила Мазаева «Кому доверить ревью API – техпису или искусственному интеллекту» (TechWriterDays-2026)

Эксперимент



Результат



Социократия: вся деятельность – эксперименты



- Цель – снимать напряжения, а не делать задачи
- Напряжение – источник мотивации для действий
- Драйвер трактует напряжение как возможность создания ценности или препятствие, мешающее потоку ценности
- Решение – не идеально, оно лишь **достаточно хорошо**, чтобы уменьшить напряжение и **достаточно безопасно**, чтобы попробовать



А что нас ждет в будущем?

ИИ-агенты для бизнеса

- Бизнес уже активно использует ИИ для обслуживания клиентов: поиск, рекомендательные системы, классические чат-боты – добавляется LLM
- Агентов необходимо подбирать, погружать в контекст, обучать, развивать – эти задачи аналогичны задачам для сотрудников, но средства отличаются
- При применении в бизнесе часто важна стоимость: специализированные легкие модели дешевле общих, которые могут их обучать
- О легких моделях для бизнеса – интересное выступление Павла Кана на ArchDays-2025 «ИИ своими силами: реальный кейс разработки без миллионных инвестиций» ([мой конспект](#), там ссылки)
- Агенты и люди образуют сложную систему – надо вести мониторинг качества, совершенствовать шаги и систему в целом
- Часть задач тестировщикам знакомы, другие – предстоит осваивать...

Отдельные агенты образуют сложную систему

[Дарья Шатко. «Как мы внедрили LLM-судей в автоматизациях клиентского сервиса» \(AIconf-2026\)](#)

е2е-пайплайн чат-бота в клиентской поддержке



Будущее: ИИ-агенты в жизни

- Если ты взаимодействуешь с ИИ, он знает твою жизнь: здоровье, типовые активности и задачи, и может использовать это, решая запросы
 - Подобрать обувь или зубную щетку и пасту с учетом индивидуальных особенностей
 - Планировать отпуск, выходные или вечер отдыха, опираясь на твою историю
 - Подстроить информации по профессиональным вопросам
 - Обеспечить контекстом, когда ты подхватываешь задачи коллеги
- Если выбирает и покупает ИИ-агент, маркетинг надо строить иначе
- Надо строить мост из нескольких стадий обработки, соединяя данные
 - Проектировать, строить и держать контекст твоей жизни, работы команды и компании
 - Выделять задачи для ИИ-агентов и организовывать взаимодействие в глобальной сети
 - Маркетплейсы и агрегаторы пробуют затянуть в свою сеть, а люди сопротивляются
 - Вместо универсальных таск-трекеров и CRM можно делать специализированные
- ИТ-разработка опережает в освоении ИИ и может быть драйвером

Об этом хорошие выступления у **Гарретта Джонстона**, мои конспекты:

[Живая компания-2025](#) (там есть ссылки на запись), [ПИР-2024](#), [ПИР-2025](#), [ПИР-2026](#)

Китай показывает пример развития ИИ

- Китай активно создает ИИ-модели, **выкладывая в открытый доступ**, [программа развития до 2035 года](#) вдохновляет
- Там не боятся, что ИИ лишит работы, он **помощник и партнер**
 - Xiaomi: роботы будут делать нашу работу, и у нас будет больше времени
 - Миссия SenseTime: пусть ИИ возглавит наш прогресс
 - Нет проблем в том, чтобы дружить с ИИ-помощником, хотя он не человек
- ИИ не заменяет человека, а дополняет – это **дает поэтапное развитие**
 - Обучение танцам человек повторяет за учителем, а аватар показывает разницу
 - Роботы будут ухаживать за больными и стариками – уже **идет апробация**: робот помогает чем может и умеет звать на помощь, его умения можно увеличивать
 - Поставлена задача снижения цены робота до 20 тысяч долларов (сейчас 40-100)
- ИИ не глуп – он учится, и с ним надо обсуждать рабочие вопросы, чтобы улучшить решение, и чтобы сам **ИИ научился и лучше помогал**



[Книга на ridero](#), [выступление на SQAdays](#) весной 2026, другие [материалы по Китаю](#)

ИИ меняет мир – соответствуйте!

- ИИ перестал быть просто помощником человека, а стал **членом команды**
- Как человеку, **ИИ нужен онбординг** – контекст проекта и принятый способ работы, его надо оценивать и развивать – работа с **harness** и HADI-цикл
- Меняется организация разработки и бизнеса, компетенции и роли – разделение труда между ИИ-агентами и человеком: **ищите свое место**



AI? Not! Super Intellect!

**Мир меняется, масштаб изменений мира
соразмерен появлению персоналок и интернета**



Максим Цепков



<http://mtsepkov.org>



Tg: [@MaximTsepkov](https://t.me/MaximTsepkov)



Mx: [@channel_tsepkov](https://www.whatsapp.com/channel/channel_tsepkov)

maks.tsepkov@ya.ru



Вопросы и
обратная связь