Яндекс

/ Heisenbug

LLM'изация тестирования в Яндексе

измеряем эффект от AI в команде из **1000+** QA-инженеров

Владислав Миронов

Руководитель службы,
Яндекс Фантех

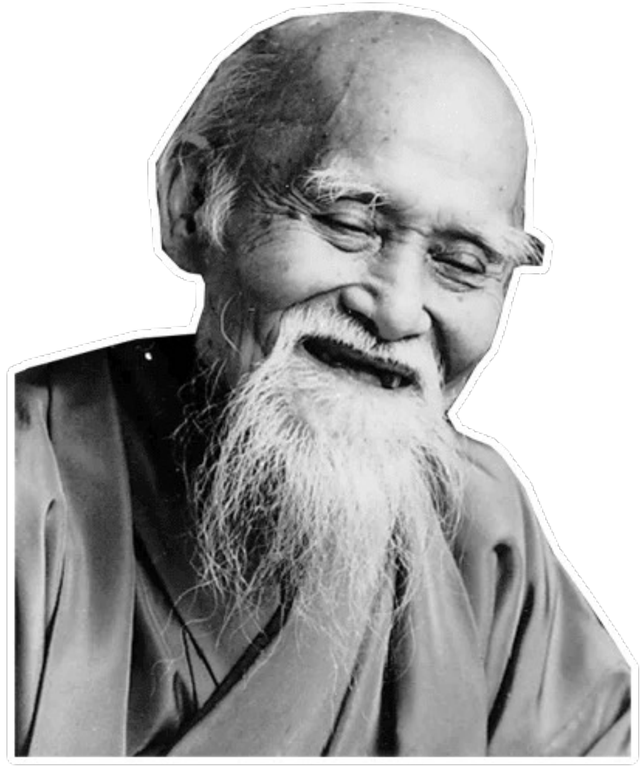
Кто я?

Владислав Миронов

Руководитель службы,
Общая инфраструктура Яндекс Фантеха

- 20+ лет в IT
- ФАНтех (Кинопоиск, Музыка, Афиша, Книги)
- Оптимизация QA процессов с GenAI

Нельзя что-то улучшить, если ты это НЕ измеряешь!



Как обычно измеряют пользу от LLM?

Как в целом оцениваешь результат генерации?

- 👍 - Хорошо
- 👎 - Плохо

Как генерация чеклиста повлияла на ваше время?

- Сэкономила время
- Не повлияла
- Потребовалось больше времени

Насколько результат генерации соответствует твоей задаче?

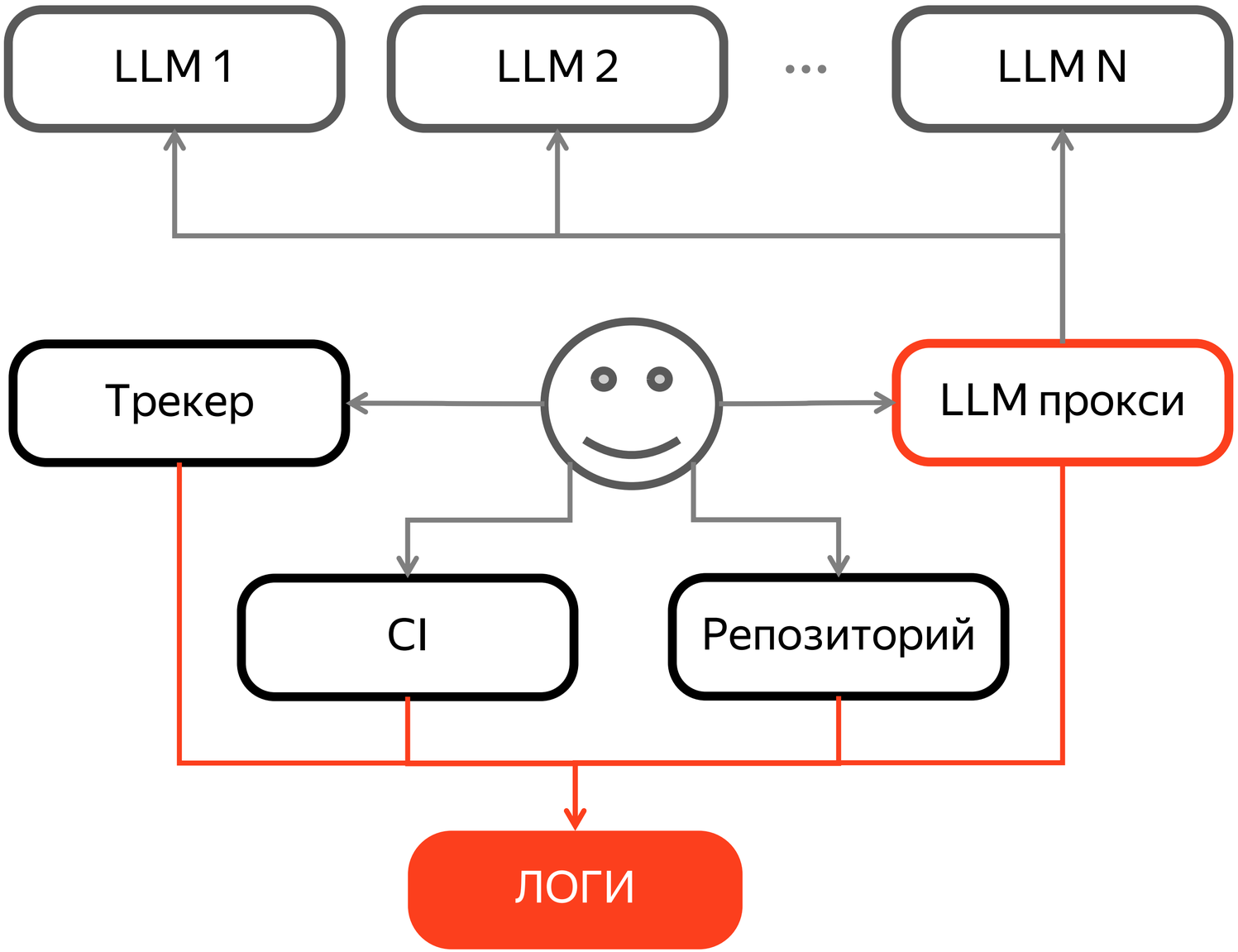
1 — совсем не соответствует · 5 — полностью подходит

- 1
- 2
- 3
- 4
- 5

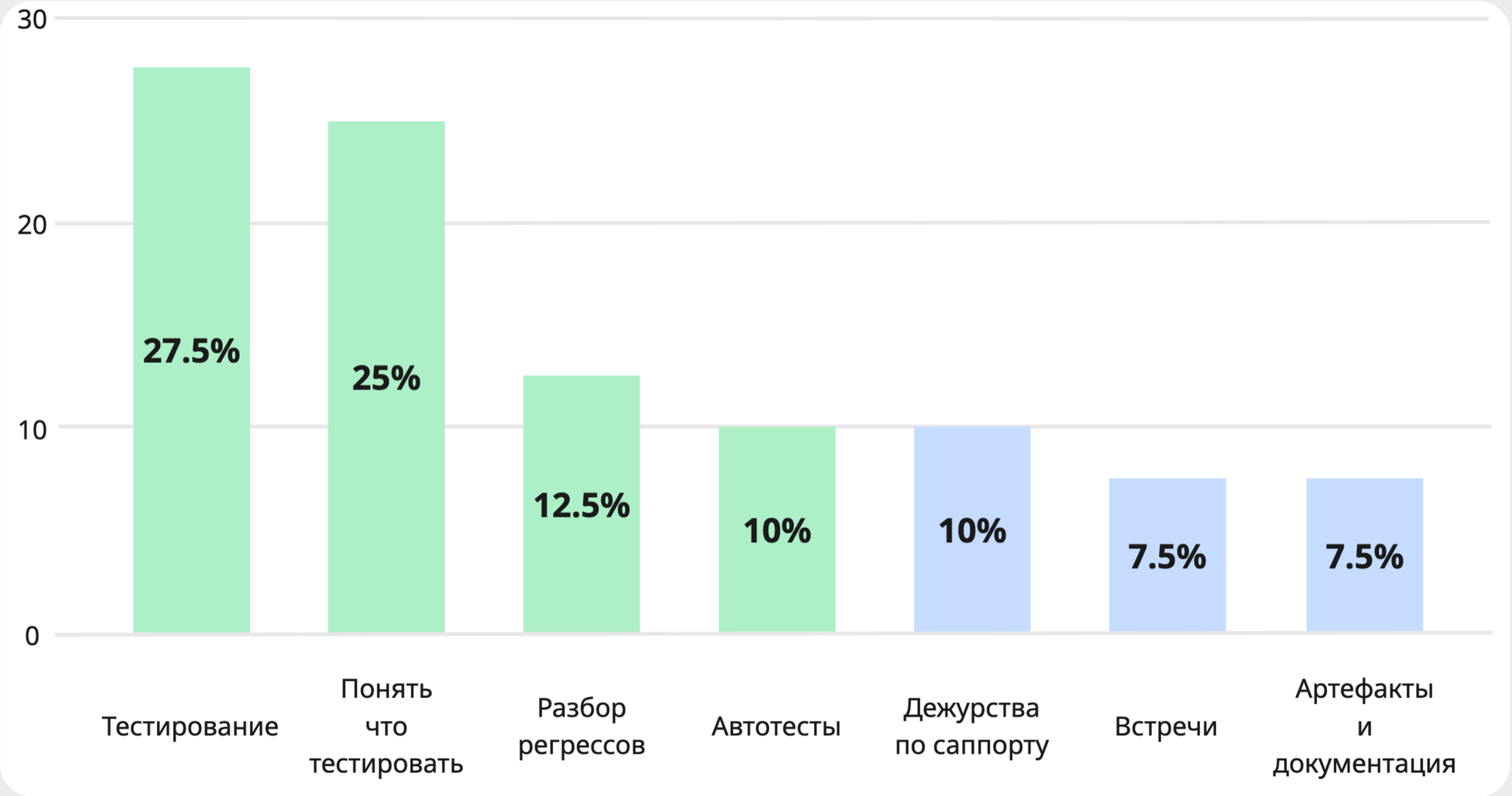
Что можно улучшить в будущем?

Отправить

Как измеряем в Яндексе



Что вообще делают QA?



Какие AI проекты мы запустили

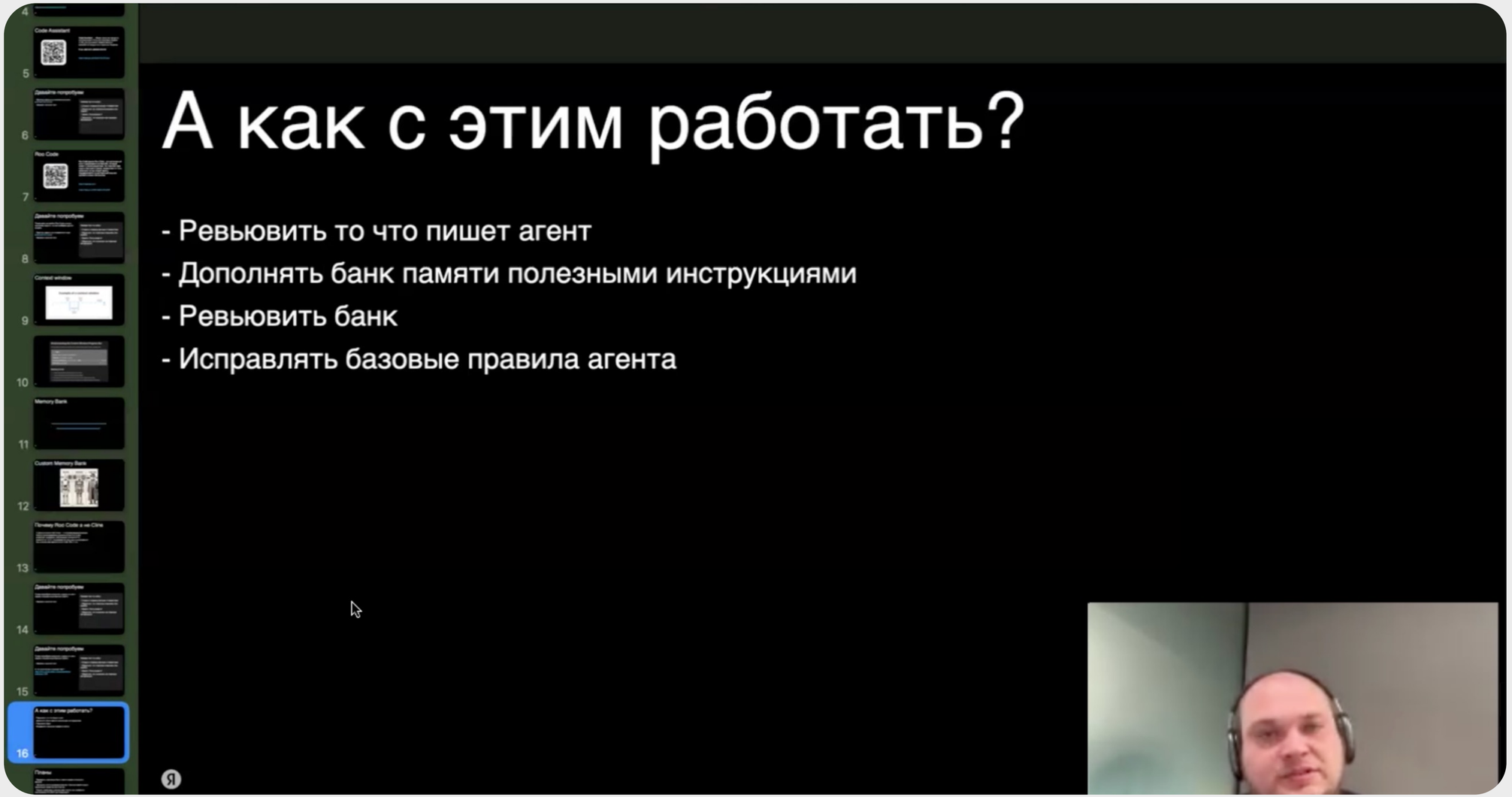
Написание E2E-автотестов

Генерация чек-листов и тест-кейсов

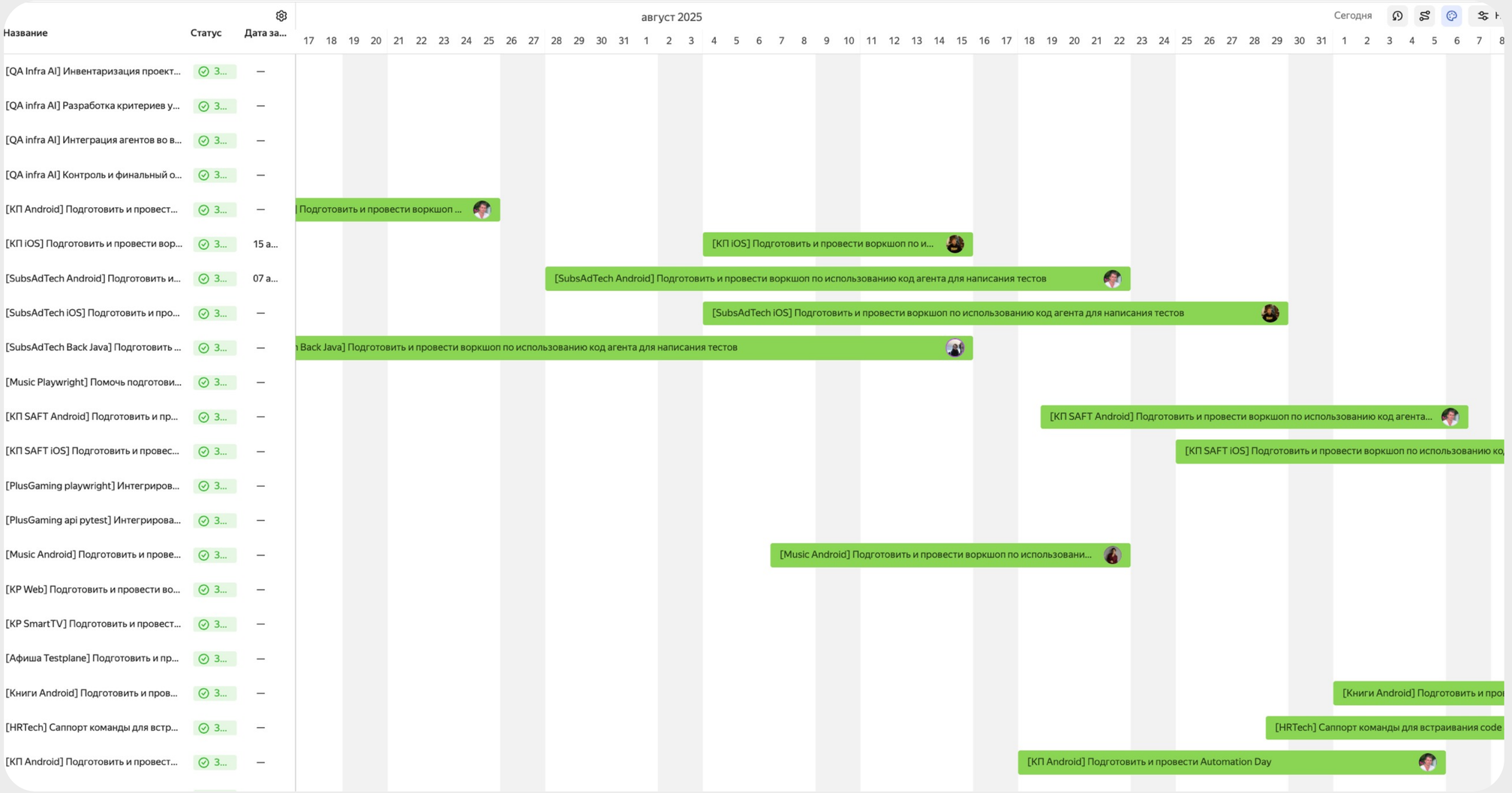
Прохождение регрессов (и не только) AI-агентами

Написание E2E-автотестов — это про вайбкодинг

E2E-автотетсы. С чего мы начали



E2E-автотетсы. К чему пришли



E2E-автотесты. Принятие в командах

До обучения	После обучения
~30%	60-70%

Написание E2E-автотестов. Технологии

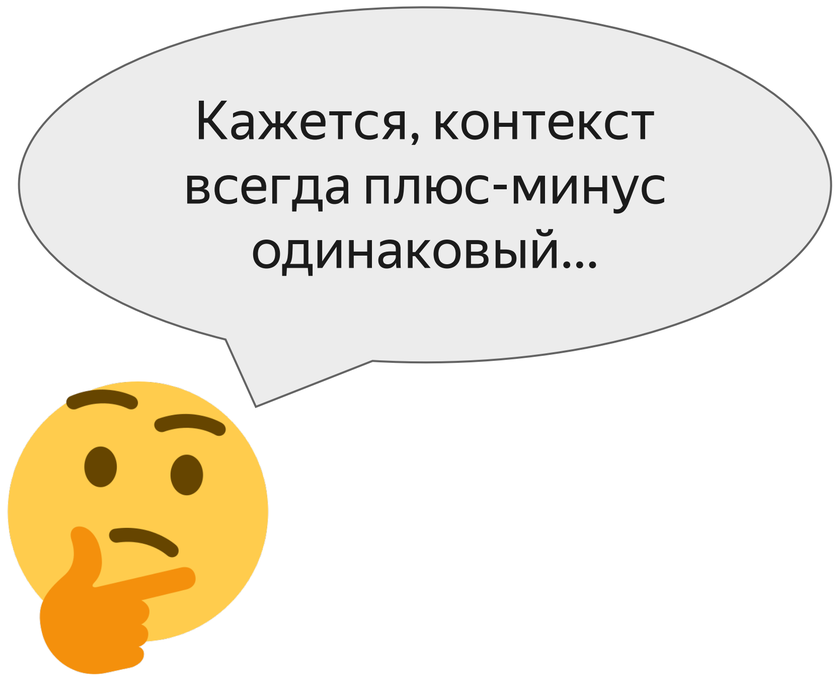
01 Roo Code => Yandex Code Assistant

02 Memory bank

03 MCP

E2E-автотесты. Начинаем с промпта

```
Напиши автотест на кейс
<CTRL-C CTRL-V кейса>
```




```
Ты Senior Automation QA с 100500 лет
опыта работы…

Используй при написании кейса все знания
мира о фреймворке…

В нашем проекте мы делаем следующим образом…

Запомнил все, что было выше?

Ок! Напиши автотест на кейс
<CTRL-C CTRL-V кейса>
```

E2E-автотесты. Подключаем memory bank

Name

..

01-project-overview.md

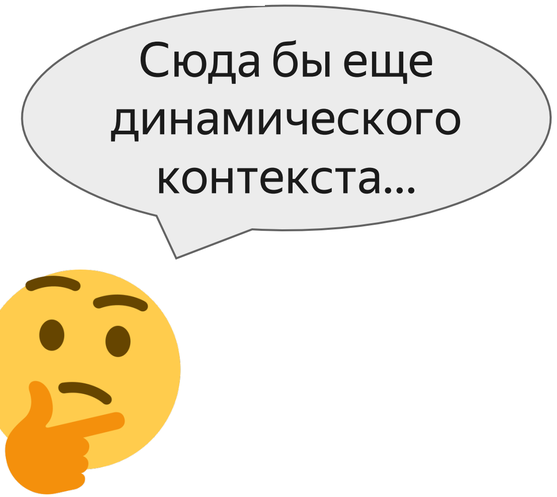
02-testing-patterns-and-style.md

03-common-problems-and-solutions.md

04-framework-api-reference.md

05-test-creation-guide.md

06-tms-and-reporting.md



05-test-creation-guide.md (25477 bytes)

05. Гайд: от Тест-кейса к Автотесту

> **Назначение:** Полный алгоритм создания автотеста для AI.

Требования к стабильности тестов

> Критерии стабильности, которым должен соответствовать каждый автотест.

В пул реквесте с тестом необходимо прикладывать флакер на 150 прогонов. Тест считается стабильным, падений).

Чеклист по тест-кейсам

> Критерии качества тест-кейсов перед автоматизацией.

Перед реализацией автотеста необходимо убедиться, что тест-кейс соответствует лучшим практикам д

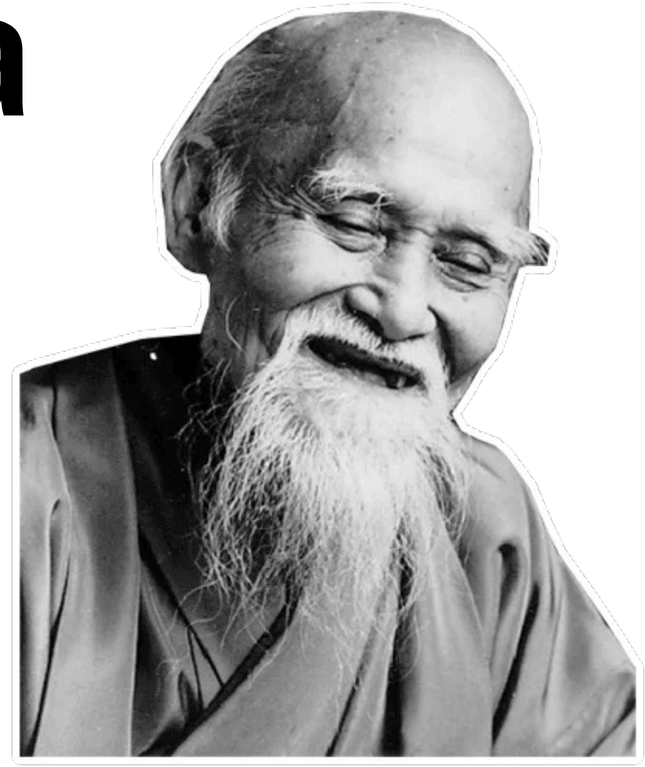
1. **Тест описывает один пользовательский сценарий**. Эвристики:
   - Можно в одном предложении описать суть теста
   - Конечный ожидаемый результат зависит от всех шагов (нельзя исключить ни один)
   - 1 шаг — идеал, 5 шагов — норма, 10+ — возможно требуется декомпозиция
2. **Тест проверяет мобильное приложение, а не бэкенд**:
   - Проверяется 1-2 связанных экрана
   - Между шагами теста не шарится состояние, которое хранится на бэкенде

E2E-автотесты. Подключаем MCP-серверы

- Управляем симуляторами
- Взаимодействуем с приложениями
- Анализируем скриншоты экранов и деревья элементов

- Hint! LLM лучше напишет E2E-тест, если ее попросить сначала пройти его на устройстве

Экономия времени — главная метрика



E2E-автотесты. Измеряем эффективность

Опросы

Как в целом оцениваешь результат генерации?

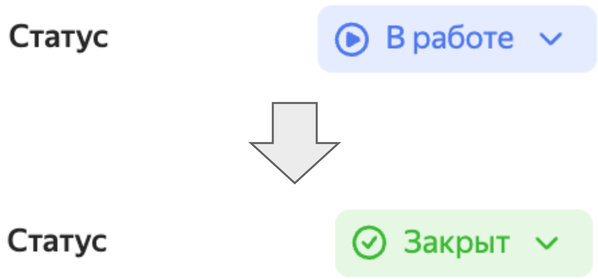
- 👍 - Хорошо
- 👎 - Плохо

Как генерация чеклиста повлияла на ваше время?

- Сэкономила время
- Не повлияла
- Потребовалось больше времени

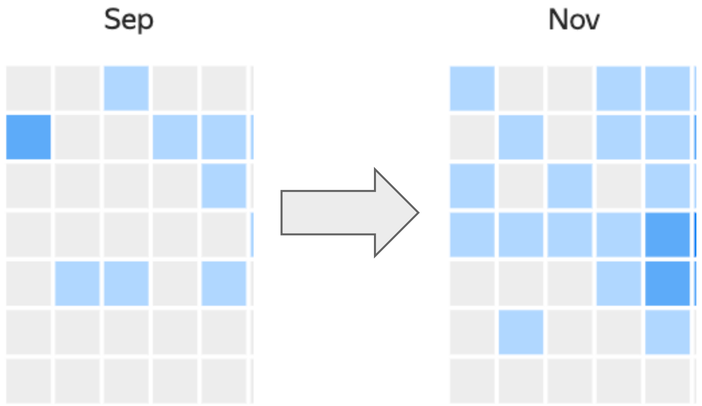
Lead Time

время от взятия задачи в работу до ее завершения



Тех. метрика

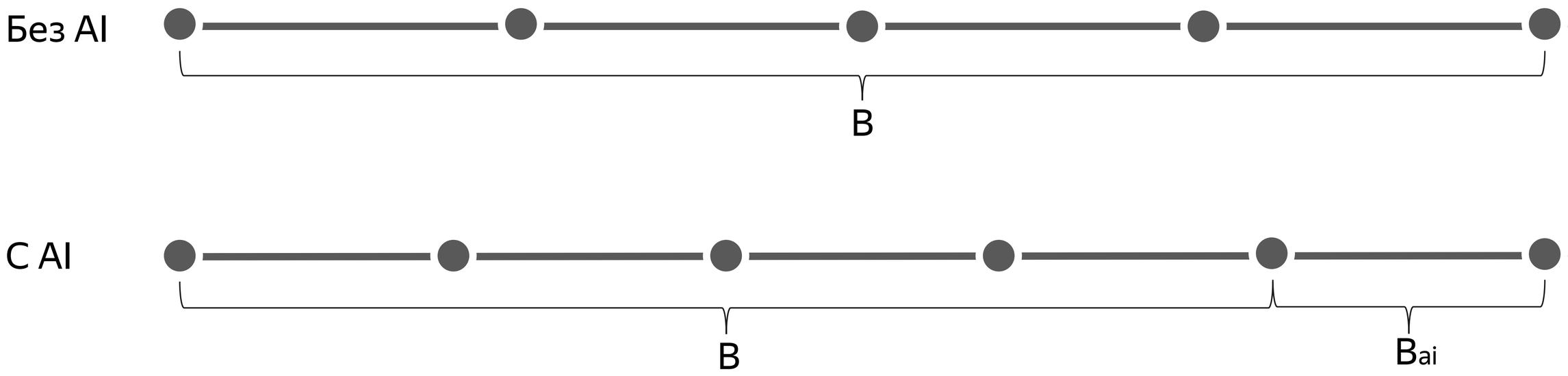
прирост коммитов у тех, кто активно использует AI



E2E-автотесты. Техническая метрика

B - количество коммитов без AI

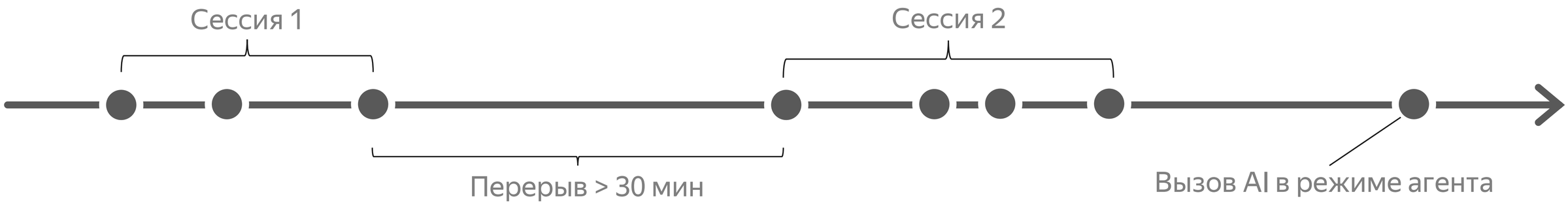
Bai - прирост коммитов



$b = B_{ai}/B$ – **доля** прироста коммитов

E2E-автотесты. Считаем b (методика)

1. Смотрим логи использования LLM у всех пользователей



2. Ищем тех, кто использует AI **более 20 минут** в день

3. Сравниваем долю прироста коммитов у них с теми, кто не использует AI

E2E-автотесты. Считаем b (результат)

1. Инженеров в категориях:
   - С AI : **~150-180**
   - Без AI: **~200-250**

2. Процент сессий с AI в когорте «с AI»: **~60-80%**

3. b = 0.2 (**+20%**)

E2E-автотесты. Сравниваем результаты метрик

Опросы

+38% среднее

Минусы метрики:

- Субъективно
- Были утверждения и про в 2,5 раза!

Lead Time

+30% среднее

Минусы метрики:

- Мало команд
- Параллельные задачи влияют

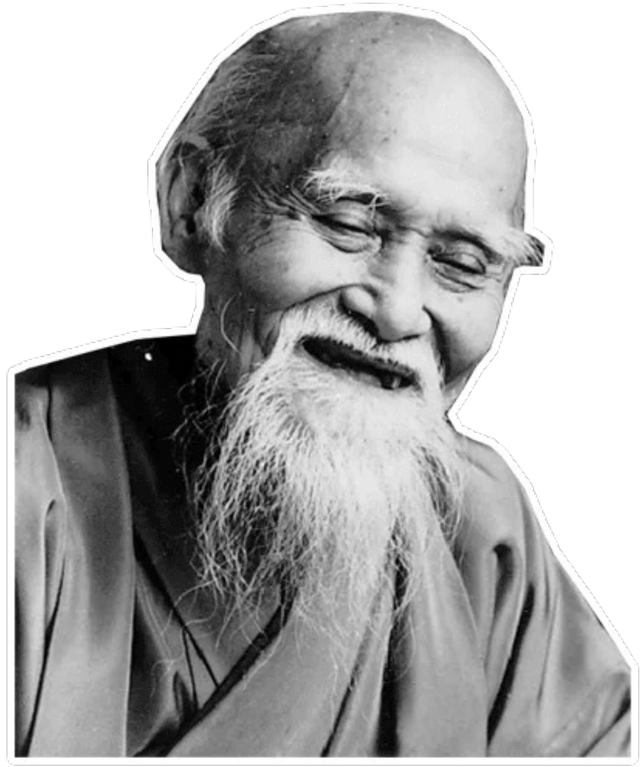
Тех. метрика

+20% среднее

Минусы метрики:

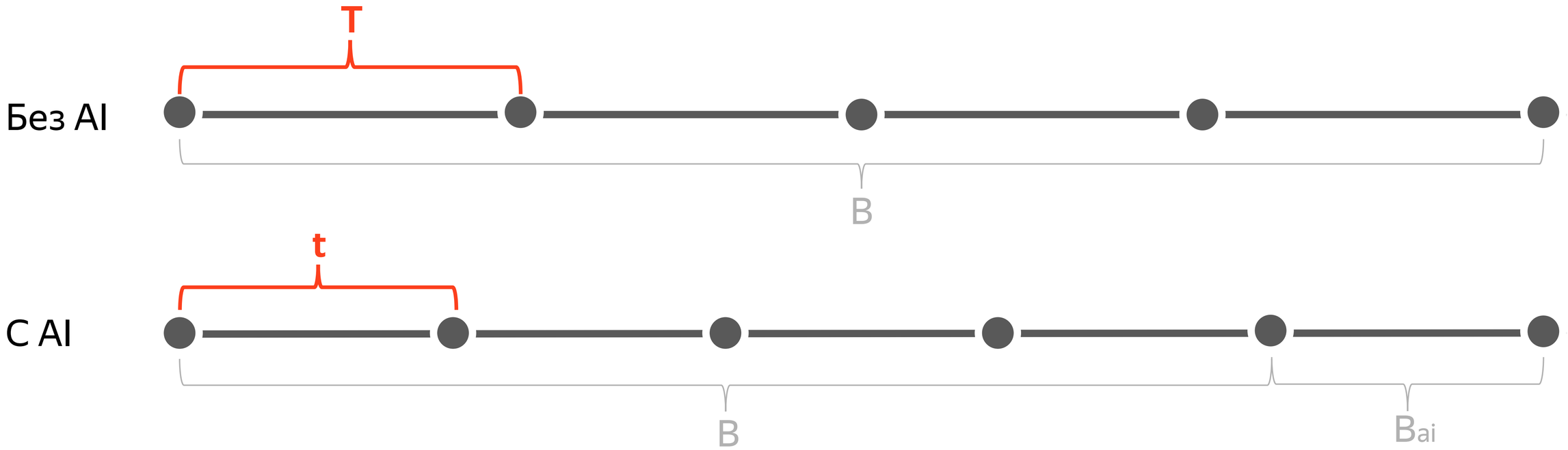
- Активно пишут не все 1000
- Коммиты бывают разные

+20% коммитов—
это НЕ время



E2E-автотесты. Считаем среднее время

T - среднее время на коммит без AI
t - среднее время на коммит с использованием AI



$$T * B = t * (B + Bai) \Rightarrow T = t * (1 + Bai/B)$$

$$\mathbf{T = t * (1 + b)}$$

E2E-автотесты. Экономия времени (ч.1)

1. Экономия на 1 коммите:
   S = T – t
   = t * (1 + b) – t
   = **t * b**

2. Суммируем все коммиты:

   $S_{полное} = \sum t * b$

Что мы должны вычислить?

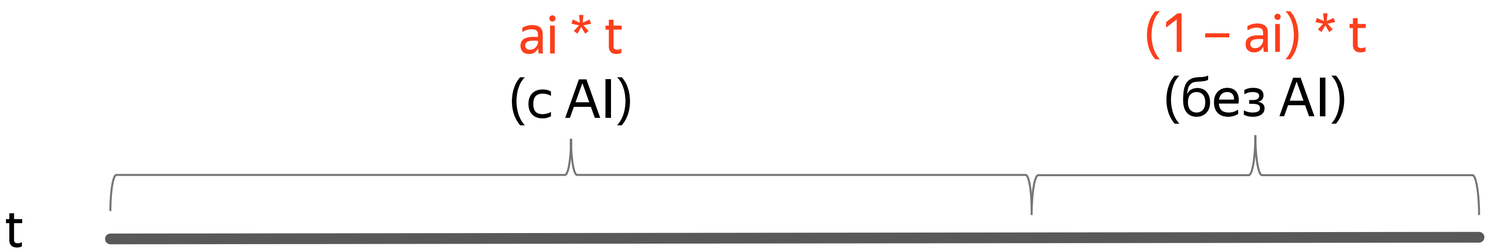
E2E-автотесты. Экономия времени (ч.2)

1. Вычисляем t:
   t = X / ai

2. Подставляем в S:

$S_{полное} = \sum X * (b/ai)$

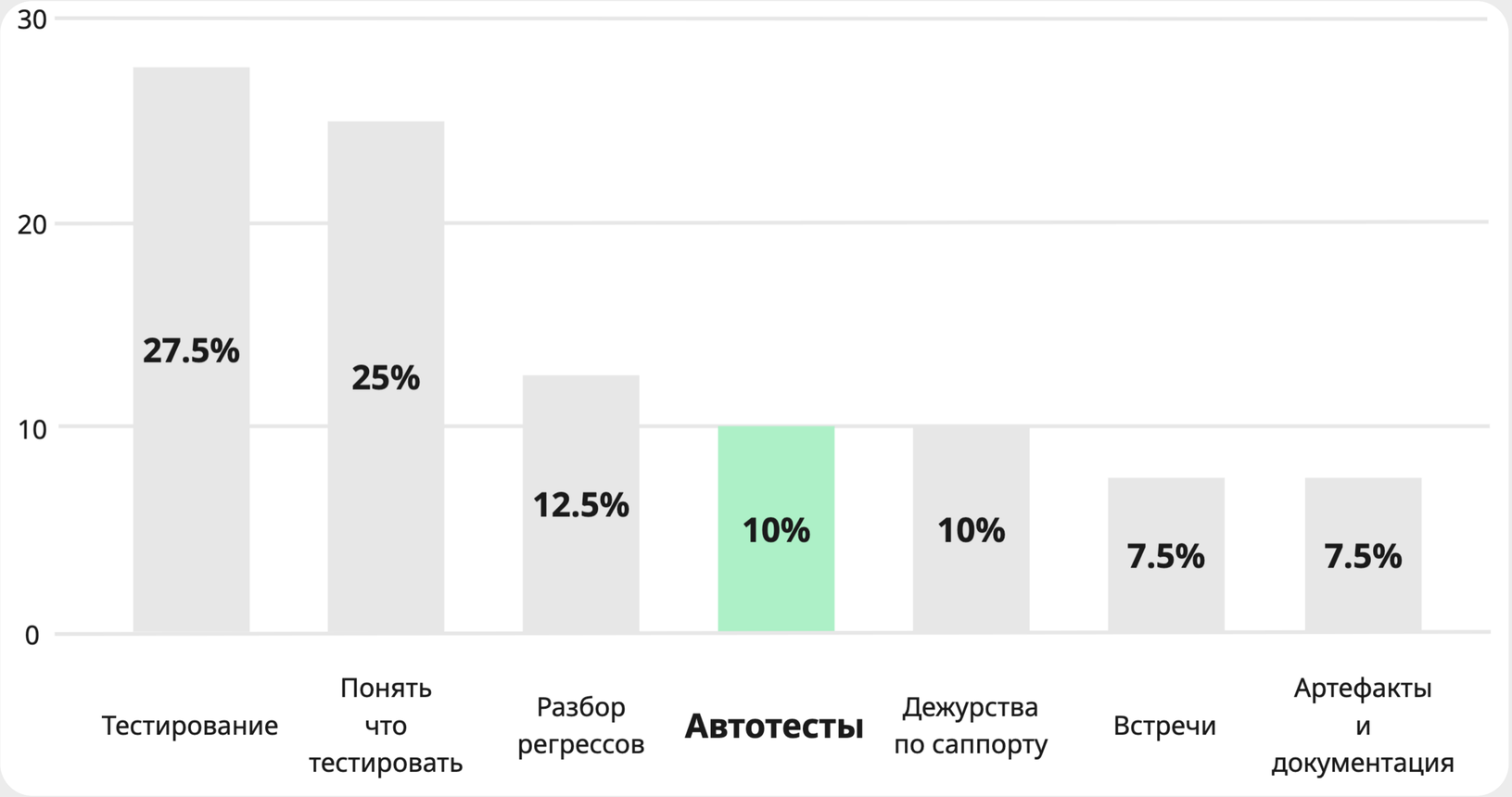
Из чего состоит время подготовки одного коммита?



X — интервал времени, в течение которого инженер использовал AI (знаем из логов)

X = ai * t, где ai — доля сессий с использованием агента (знаем из логов)

E2E-автотесты. Результат предсказуем!



20% от 10% — это 2%!

Почему 2% — это НЕ плохо?

- Качество инструментов улучшается, повышая b
- Время кодинга в QA, вероятно, будет увеличиваться
- Есть AQA! Они кодят значительно больше

К/Ф «Начало», реж. Кристофер Нолан, 2010

E2E-автотесты. Выводы

К/Ф «Властелин колец: Братство кольца», реж. Питер Джексон, 2001

01

Не верьте опросам! И логируйте все, что можете

02

Обучение в малых группах **гораздо** эффективнее

03

+20% эффективности процесса ≠ **+20% времени** инженера

04

+20% эффективности кодинга — все равно здорово ☺

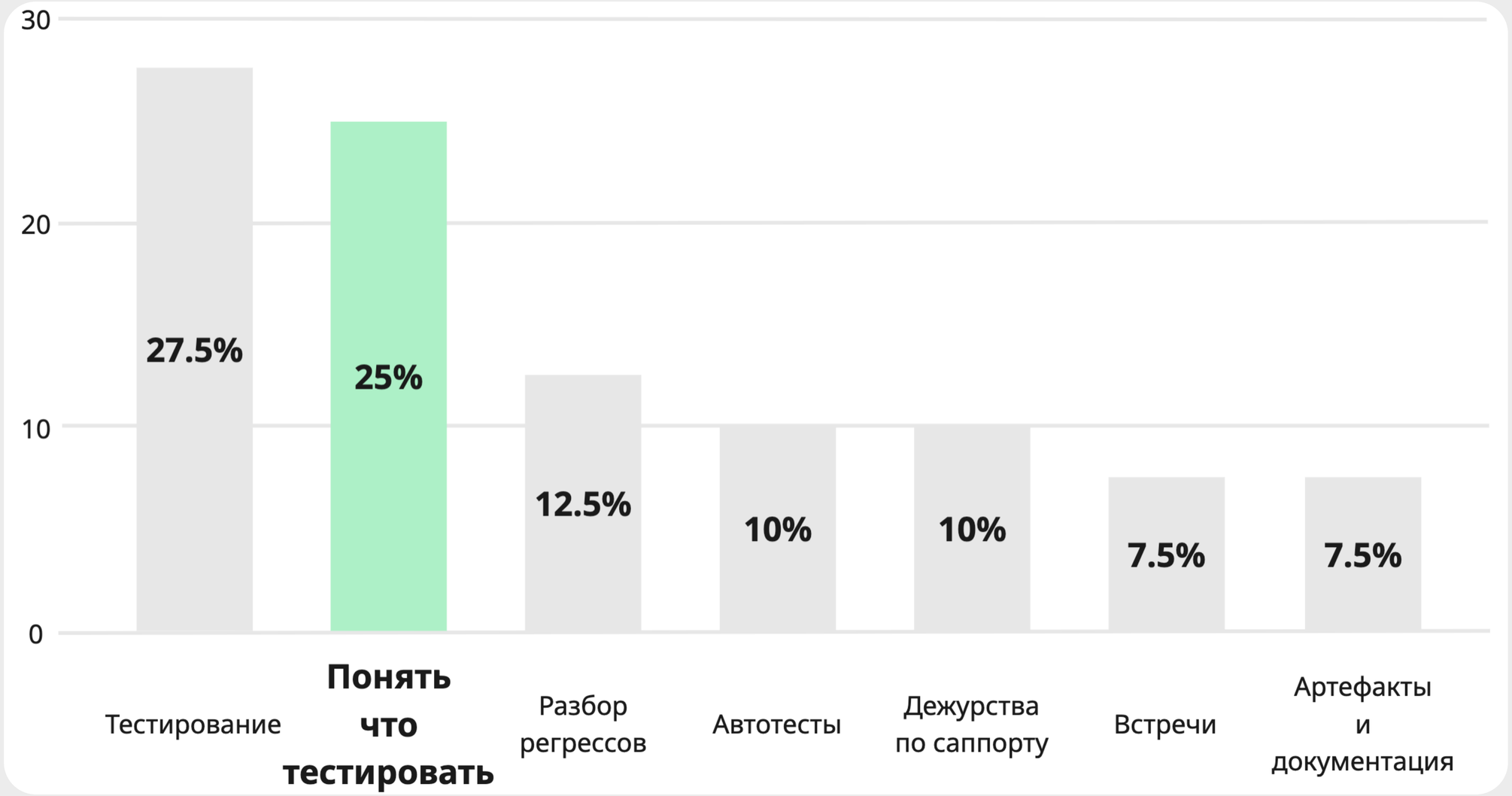
Какие еще AI проекты мы запустили

Написание E2E-автотестов

Генерация чек-листов и тест-кейсов

Прохождение регрессов (и не только) AI-агентами

Чек-листы. На что влияют?



Чек-листы. Как делали?

Источники данных

1. Pull Request
2. Описание задачи
3. Wiki...

Добавлена функциональность отображения полного описания профиля в виде модальной шторки для iOS приложения. Если текст описания пользователя превышает 3 строки, он обрезается, и появляется возможность тапом по тексту открыть шторку с полным текстом. В шторке отображается заголовок с именем пользователя, полный текст описания (с поддержкой кликабельных ссылок) и кнопка закрытия.

> Основные изменения в коде

Чек-лист

Приоритет	Проверка	Ожидаемый результат
Critical	**Основной сценарий: Открытие шторки с длинным описанием**	
	Проверить отображение блока с описанием профиля, если текст занимает более 3 строк.	Текст обрезается до 3 строк. Область текста становится кликабельной (визуальный индикатор не указан, но функционально должна реагировать на тап).
	Тапнуть на обрезанный текст описания.	Снизу экрана открывается модальная шторка.
	Проверить содержимое открывшейся шторки.	В заголовке отображается имя пользователя. В теле шторки отображается полный, необрезанный текст описания. Справа в заголовке отображается кнопка закрытия (крестик).
	Проверить высоту шторки.	Высота шторки автоматически подстраивается под высоту контента (заголовок + полный текст).
Critical	**Основной сценарий: Закрытие шторки**	
	Тапнуть на кнопку с крестиком в шторке.	Шторка плавно закрывается. Фокус возвращается на экран профиля.
	Выполнить свайп-жест вниз по открытой шторке.	Шторка плавно закрывается (стандартное поведение для `.sheet` в iOS).
Critical	**Сценарий: Отсутствие шторки при коротком описании**	
	Проверить отображение блока с описанием, если текст помещается в 3 строки или меньше.	Текст отображается полностью, без обрезки. Тап на тексте не приводит к каким-либо действиям (шторка не открывается).
Medium	**Работа с ссылками в описании**	
	Проверить поведение ссылок в обрезанном тексте описания на основном	Тап на ссылке в обрезанном тексте не открывает ссылку, а вместо этого открывает шторку с полным описанием.

Как собирали обратную связь?

Как в целом оцениваешь результат генерации?

- 👍 - Хорошо
- 👎 - Плохо

Как генерация чеклиста повлияла на ваше время?

- Сэкономила время
- Не повлияла
- Потребовалось больше времени

Насколько результат генерации соответствует твоей задаче?

1 — совсем не соответствует · 5 — полностью подходит

- 1
- 2
- 3
- 4
- 5

Что можно улучшить в будущем?

Отправить

Чек-листы. Результаты опроса

Заполнение опросов

- **250-270** запусков в день
- **4-6** заполненных опросников в день (**2%**)
- Пользуются ли остальными 98% чек-листов?

Оценки

- **90** опросников анализировали
- **74** оценки «хорошо» VS **16** оценок «плохо»
- **~82% удачных** чек-листов

Сохраненное время

- **15 минут** сохраненного времени на **1 удачный** чек-лист
- **0 минут** лишнего потраченного на **1 неудачный** чек-лист

>70 часов

Time = ActiveLists * 0.82 * 15

*ActiveLists — число ежедневных запусков генератора в активных очередях

Перепроверяем через Lead Time

1004 задачи с AI

1694 задач без AI

С AI

Медиана (типичная задача):

1 час 8 минут

Без AI

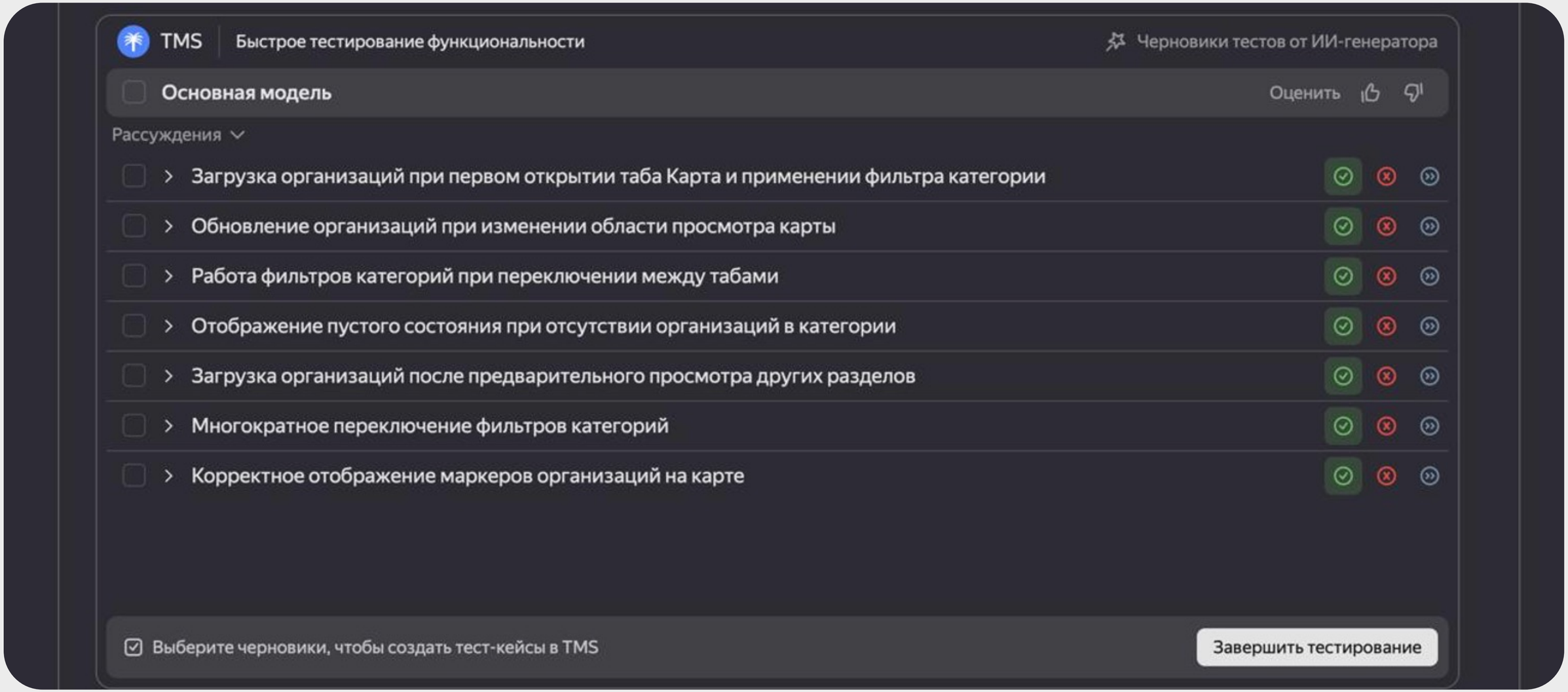
Медиана (типичная задача):

1 час 8 минут

+24%

Возвратов в разработку с AI

Чек-листы. Что дальше с метриками?



Результаты сбора логов

- **>5000** кликов собрано
- **25-30%** чек-листов реально используются
- **353** уникальных пользователя
- Субъективная оценка => объективные данные для анализа

Открывшиеся возможности для анализа

Чек-листы. Выводы

К/Ф «Властелин колец: Братство кольца», реж. Питер Джексон, 2001

01

НЕ верьте опросам!

02

Не вся польза от AI хорошо переводится во **время**

03

AI чек-листы значимо улучшают **качество** тестирования

04

Помимо сбора фидбэка, собирайте логи использования

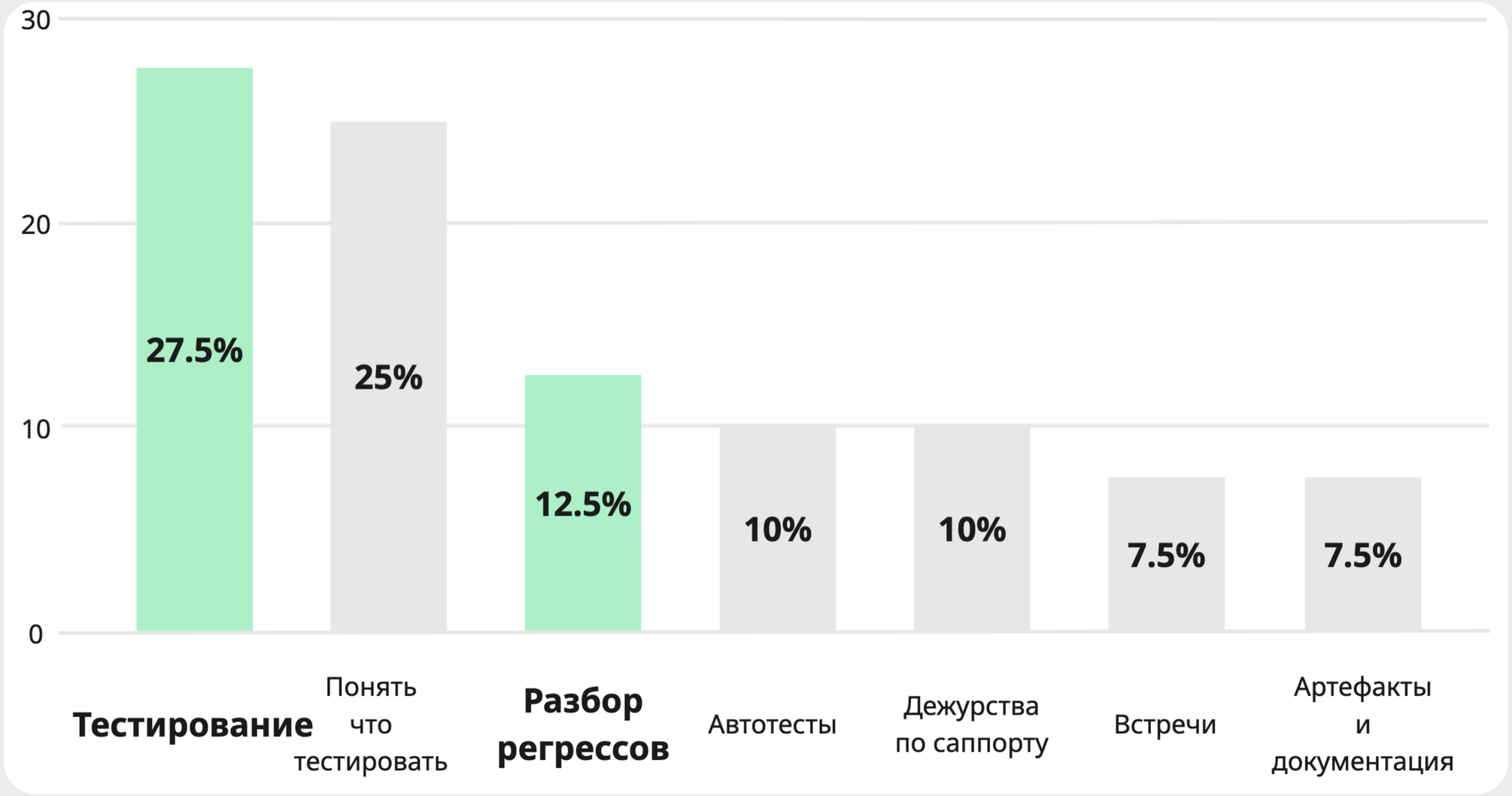
Какие еще AI проекты мы запустили

Написание E2E-автотестов

Генерация чек-листов и тест-кейсов

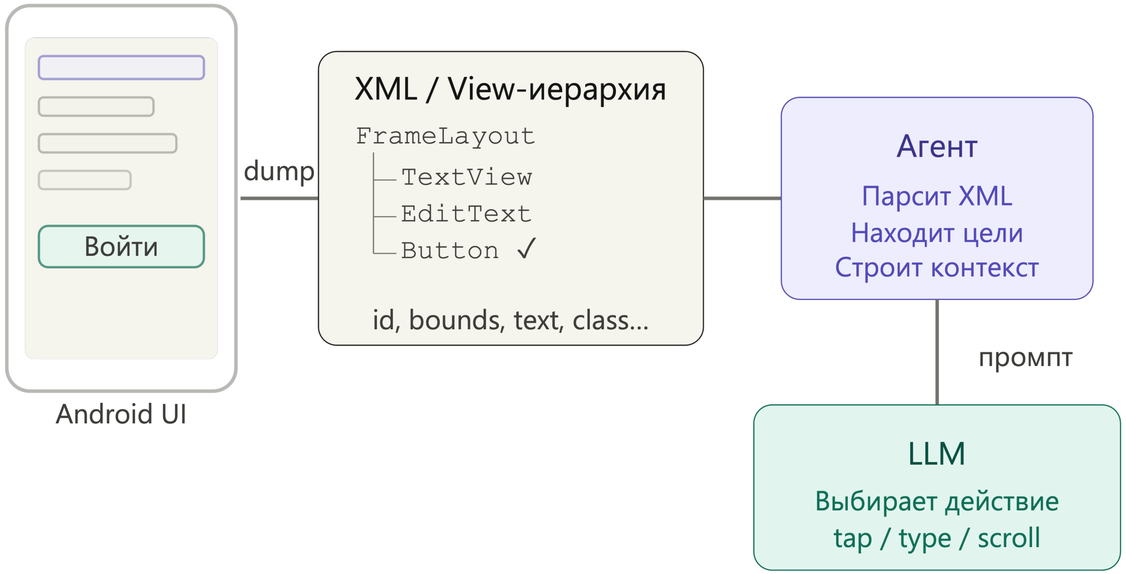
Прохождение регрессов (и не только) AI-агентами

AI-агенты. На что влияют?

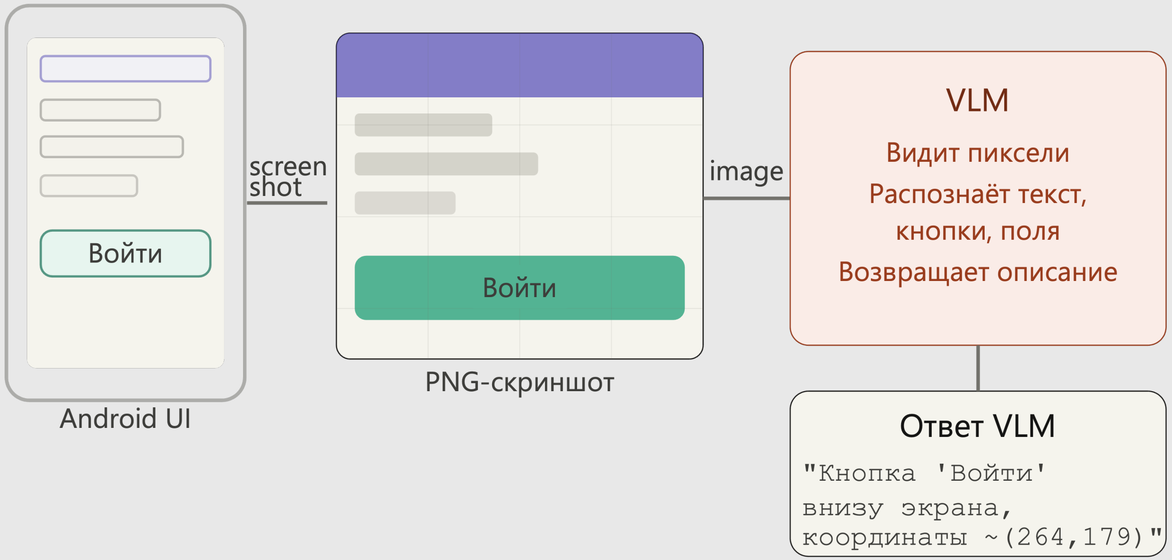


AI-агенты. Что делаем?

Разбор дерева



VLM



AI-агенты. Победила дружба

AI-агенты. Результаты

500 сценариев

- Уравновешены «зеленые» и «забагованные»

На 1/3 меньше багов

- По сравнению с **асессорами** (краудсорсинг)
- Модель тренируется на **«паранойю»**

66% зеленых пройдено

- Остальные нужно допройти человеку

Плюсы

Очевидная экономия времени:

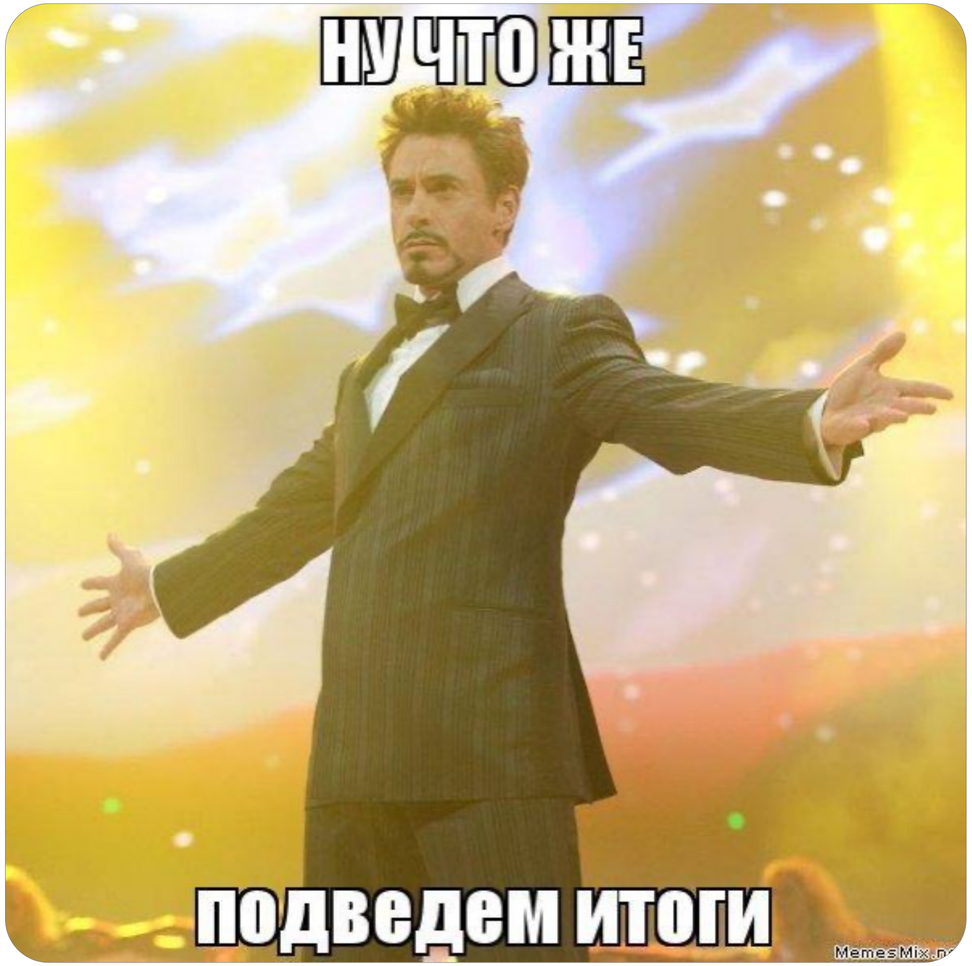
1. До 66% регресса с AI
2. Меньший процент — тоже польза!
3. Главное — контролировать процент багов

Минусы

Есть ограничения:

1. Пока выигрываем у **асессоров** (внешних тестировщиков)
2. Токенозатратно. Надо аккуратно считать экономику

Подведем итоги



01

НЕ верьте опросам!

И собирайте логи
(**x10** данных)

02

Вайбкодинг **на 20%** эффективнее кодинга

(но это не равно сохраненному времени)

03

AI чек-листы значимо (**+24%**) улучшают **качество** тестирования

04

Параноидальный AI-агент может выполнить **до 66%** регресса

Но надо следить за расходом токенов

Яндекс

Вопросы?

Влад Миронов

Яндекс Фантех,
руководитель службы общей инфраструктуры