

Что, если нагружать будут разработчики? Нагрузочные тестировщики больше не нужны?

Александр Иванов

Руководитель группы разработки
инструментов нагрузочного тестирования

Александр Иванов

Руководитель группы
разработки инструментов
нагрузочного тестирования

Yandex Infrastructure



1. А какая вообще от нас польза?

2. Как мы тестируем в Яндексе

3. Работает ли такой подход

4. Так нужны или нет?

5. Ссылки

Барух Садогурский: У нас DevOps. Давайте уволим всех тестировщиков

- В докладе красочно и точно освещена проблема
- Много справедливо и для нагрузочного тестирования
- Доклад классный, его приятно слушать



<https://www.youtube.com/watch?v=8eH3k4BxV6k>

Вопросы бизнеса, на которые отвечает нагрузочное тестирование

1

Выдерживает ли продукт нагрузку после релиза?

Быстро развивать продукт очень круто. Но никто не любит неработающие банкоматы, тормозящие сайты, недоступность контента, недоставленные сообщения

2

Не стало ли хуже?

Сейчас закидать проблемы железом очень просто. Но это всё же деньги. Если не следить за производительностью, не будет ясно, сколько мы платим за ту или иную фичу

3

Можно ли помочь разработчикам?

Исследования и поиск bottlenecks, воспроизведение уже выявленных проблем. Сравнение гипотез. Но! Для бизнеса это всё должно снижать риски и генерировать доход

Тогда почему вообще такие вопросы?

- DevOps
- Микросервисы
- Облака

Тогда почему вообще такие вопросы?

- DevOps
- Микросервисы
- ~~Облака~~

DevOps

- Тут всё про то, чтобы делать изменения быстрее
- И мельче
- И что для нас гораздо хуже, релизить чаще

Микросервисы

- А тут всё про то, чтобы разделять софт на небольшие легко заменяемые компоненты
- И что хуже для нас, растить количество компонентов и связей между ними

Как с ЭТИМ ЖИТЬ...

- «Я буду работать ещё больше» — Боксёр
(Дж. Оруэлл. «Скотный двор»)

Как с ЭТИМ ЖИТЬ...

- ~~• «Я буду работать ещё больше» — Боксёр
(Дж. Оруэлл. «Скотный двор»)~~
- А может, ну его нафиг?

Как с ЭТИМ ЖИТЬ...

- ~~• «Я буду работать ещё больше» — Боксёр
(Дж. Оруэлл. «Скотный двор»)~~
- ~~• А может, ну его нафиг?~~
- Буду тестировать только опасные изменения

Как с ЭТИМ ЖИТЬ...

- ~~• «Я буду работать ещё больше» — Боксёр
(Дж. Оруэлл. «Скотный двор»)~~
- ~~• А может, ну его нафиг?~~
- ~~• Буду тестировать только опасные изменения~~
- Подумаем

Как с ЭТИМ ЖИТЬ...

- ~~• «Я буду работать ещё больше» — Боксёр
(Дж. Оруэлл. «Скотный двор»)~~
- ~~• А может, ну его нафиг?~~
- Буду тестировать только опасные изменения
- Подумаем :-)

1. А какая вообще от нас польза?

2. Как мы тестируем в Яндексе

3. Работает ли такой подход

4. Так нужны или нет?

5. Ссылки

Вводные

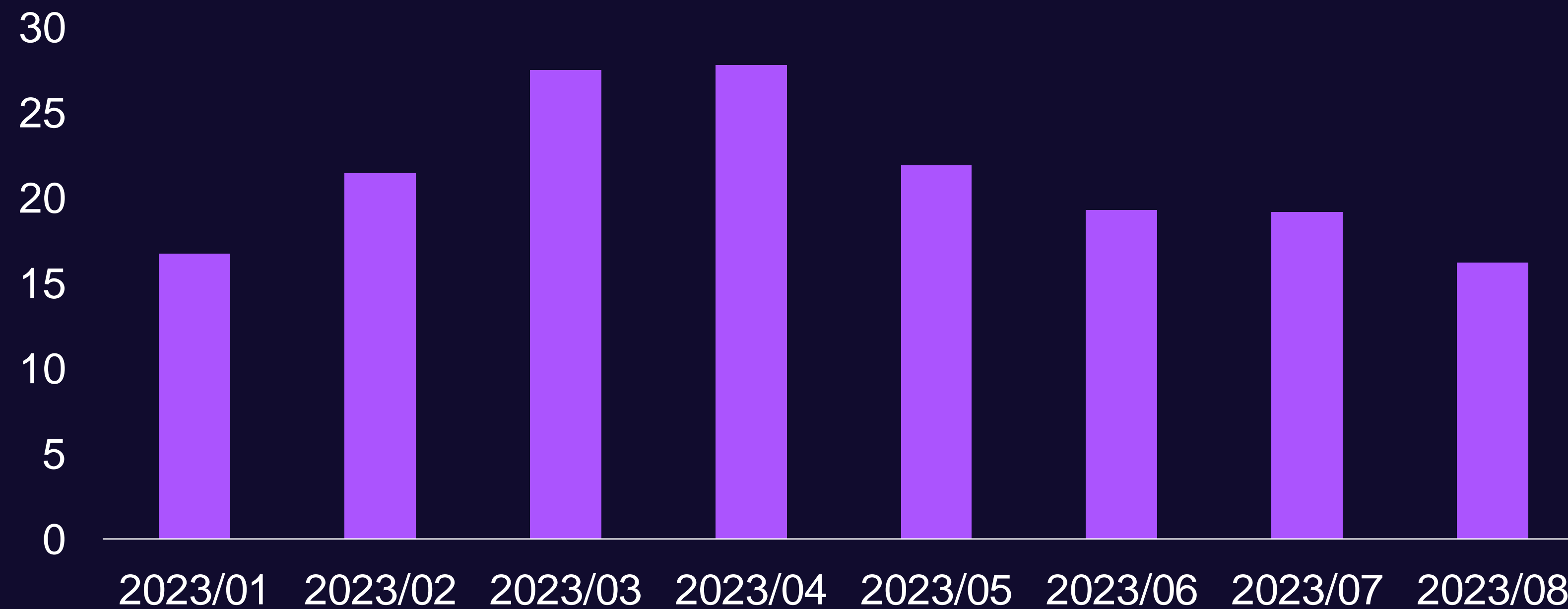
- Много проектов
- Быстрые релизные циклы

Как тестируем сейчас

- Тестируют разработчики
- Автоматизируют разработчики

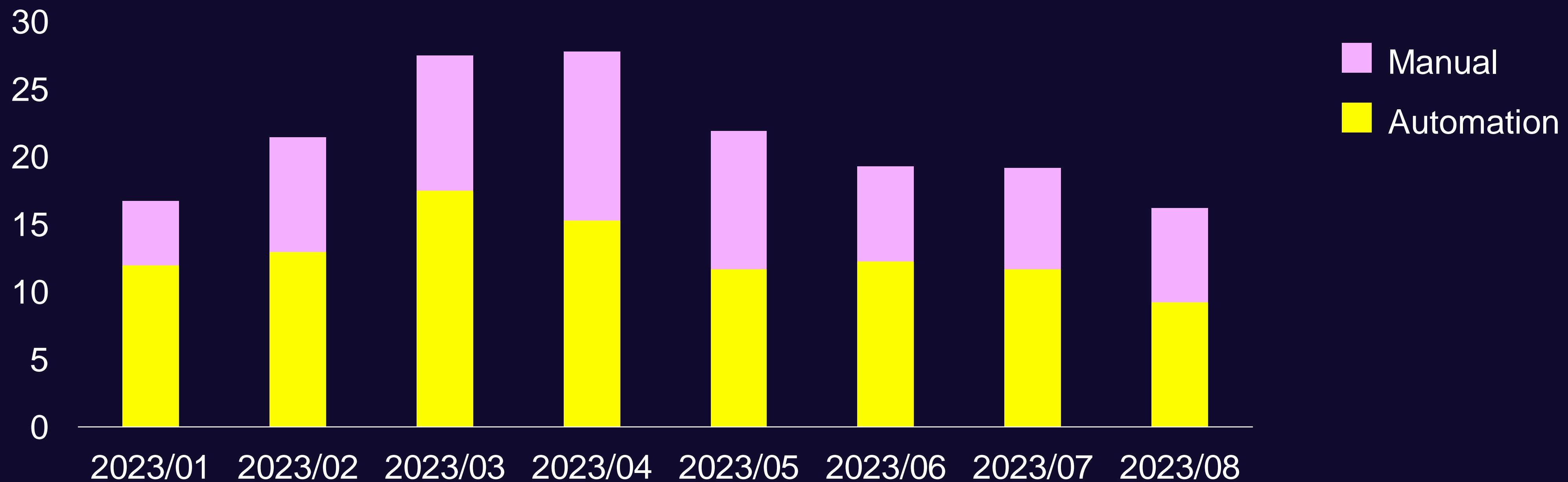
~20к тестов в месяц

Общее количество тестов, к



Высокая доля автоматизации

Распределение по источникам запуска, к



Работает ли это?

- Масштабируемость
- Разумное покрытие
- Высокий процент автоматики
- Переход количества в качество
- Контроль только там, где нужно

1. А какая вообще от нас польза?
2. Как мы тестируем в Яндексе
- 3. Работает ли такой подход**
4. Так нужны или нет?
5. Ссылки

Подход вызывает много вопросов

1

Зачем это всё разработчику? У него и своей работы много

Разработчик и так прилично занят. Часто он и так пишет юнит-тесты. Зачем ему ещё и это?

2

Как гарантировать, что не снизится качество?

Даже если начнёт, то с учётом постоянных дедлайнов наверняка будет соблазн выкатить сырое. Ну, или сделать тяп-ляп

3

Это же целая сложная область. Откуда нужные знания у разработчиков?

Нагрузочное тестирование — довольно сложный процесс.

Нагрузочные тестировщики не просто так едят свой хлеб. Как же разработчик так быстро сможет освоить это?

Экспертиза

Поговорим о сложности

- Подготовка стендов
- Анализ логов
- Создание методик
- Написание эмуляторов
- Профилирование
- Автоматизация
- Анализ статистики
- Знание инструмента

**И это всё?
Ну, нет**

- Опыт
- Знание best practice
- Умение выстроить весь процесс
- Готовность проводить интеграционные тесты

Выносим сложность за скобки

- Инфраструктура по кнопке
- Хранилища логов
- Статьи и лекции с теорией
- Эмуляторы как код
- Примеры автоматизации
- Экосистема, в которой упрощается каждый шаг

Yandex Infrastructure

Мы создаём надёжную и масштабируемую платформу для эффективности разработчиков Яндекса



Многие наши разработки можно попробовать в Yandex Cloud, а YDB и YTsaurus доступны в Open Source



- Строим ЦОДы
- Обеспечиваем доступность по проводам и без
- Надёжно храним данные
- Создаём удобные инструменты разработки
- Масштабируем внутреннее облако



Риски

Поддержка нужного уровня качества

- Нашёл ≠ Исправили

Поддержка нужного уровня качества

- Нашёл ≠ Исправили
- Исправили ≠ Улучшили

Поддержка нужного уровня качества

- Нашёл ≠ Исправили
- Исправили ≠ Улучшили
- Отказ от проведения теста ≠ Плохо

Меняется механизм достижения качества

Имхо

- Мы не рассчитываем, что разработчики хотят или могут
- Мы учим их и показываем, что это им даст. Более того, инструменты и практики делают проведение тестов **ВЫГОДНЫМ**



- Мы не просто перекладываем работу с одних плеч на другие
- Мы повышаем экспертный уровень тех, кто максимально влияет на качество продукта



Подходы дополняют друг друга

**Во многих случаях правильно разделять
ответственность. Если риск неприемлем,
надо добавлять новые уровни контроля**

e.g. Финансы, Медицина

Однако тестировать 100500 релизов — невозможно

Мотивация

А как разработчики отнесутся к новым обязанностям?

- Отрицание
- Гнев
- Торг
- Депрессия
- Принятие

А как разработчики отнесутся к новым обязанностям?

- ~~Отрицание~~
- ~~Гнев~~
- ~~Торг~~
- ~~Депрессия~~
- ~~Принятие~~
- Хорошо, если это будет им выгодно

AdHog-тест по кнопке

- Не надо с людьми общаться!
- Можно проверить любую дичь
- Не рвёт контекст
- Поправить скрипты можно в том же merge request'e

Плюсы

- Уникальность знаний снижается
- Качество может вырасти
- Мотивация появится, если будет выгодно

Плюсы

- Уникальность знаний снижается
- Появляется возможность вырастить качество не теряя скорости
- Растёт уровень разработки

Минусы

- Для быстрого обучения нужны хорошие инструменты, гайды, саппорт
- Чтобы качество росло, нужно менять процессы
- Крутые специалисты стоят дороже

1. А какая вообще от нас польза?
2. Как мы тестируем в Яндексе
3. Работает ли такой подход
- 4. Так нужны или нет?**
5. Ссылки

Конечно!

Ведь можно жить как раньше

- Многим выгодно чётко разделять зоны ответственности
- Даже сейчас CI есть не у всех
- Есть проекты, где разработчики внешние

Конечно!

Ведь изменения — ЭТО ВОЗМОЖНОСТЬ

- Появляются новые роли и возможности роста
- Выстроить новые процессы невозможно без экспертов
- Всегда будут задачи, которые потребуют консультации

1. А какая вообще от нас польза?
2. Как мы тестируем в Яндексе
3. Работает ли такой подход
4. Так нужны или нет?

5. Ссылки

Yandex Load Testing

- Идентичен и внутри компании, и снаружи
- Впитывает наработки и исправляет недостатки
- Подробнее — в докладе по QR



https://www.youtube.com/watch?v=3u6_FDhKk5s

Комьюнити-чат в Telegram

- Измеряйте производительность сервисов и выявляйте bottleneck, прогнозируйте увеличение HW
- Эмулируйте разные профили нагрузки и выявляйте деградации компонентов перед релизами



t.me/YandexLoadTesting

Присоединяйтесь к Yandex Infrastructure в Telegram

Канал о том, как мы развиваем
и поддерживаем IT-инфраструктуру,
на которой работает весь Яндекс



https://t.me/yandex_infrastructure