

Где у кодовой базы руль и куда его поворачивать

...

CLTanuki • 08.06.2023

Метрики

Что собираем

- Продуктовые метрики
- Технические метрики
- Сигналы

Пирамида метрик

Бизнес-метрики

Profit

Экономические метрики

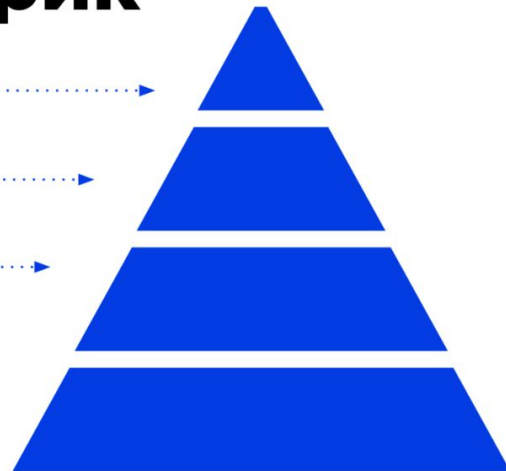
CAC, mARPPU, AOV(средний чек)...

Product metrics

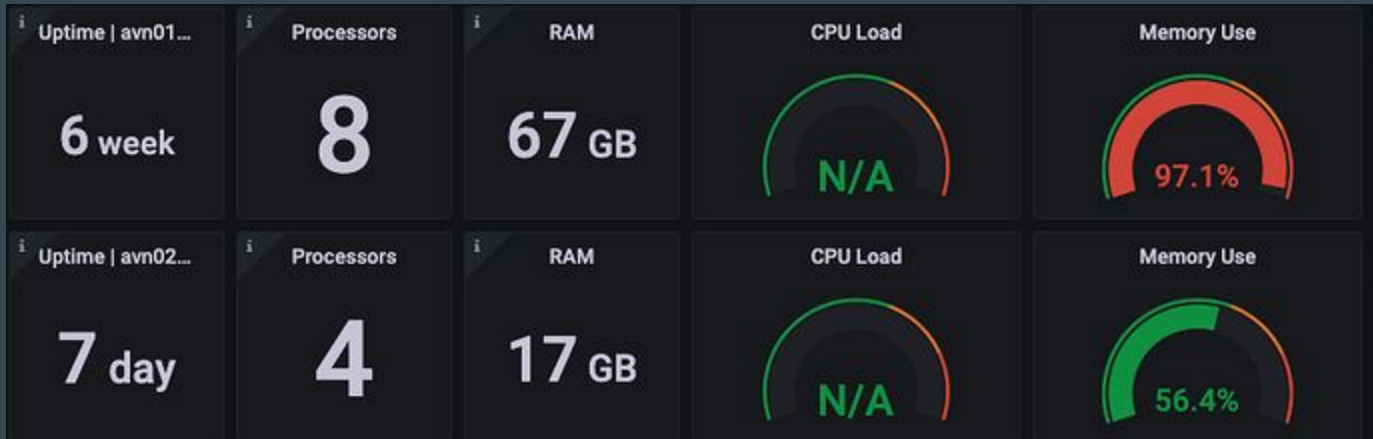
RR, Intensity, NPS, ER...

Specific System metrics

% сложных подборов, достаточность преподавателей, конверсии по воронке, прирост лидов и т.д.







USE - Utilisation, Saturation and Errors

Utilization, Saturation & Errors

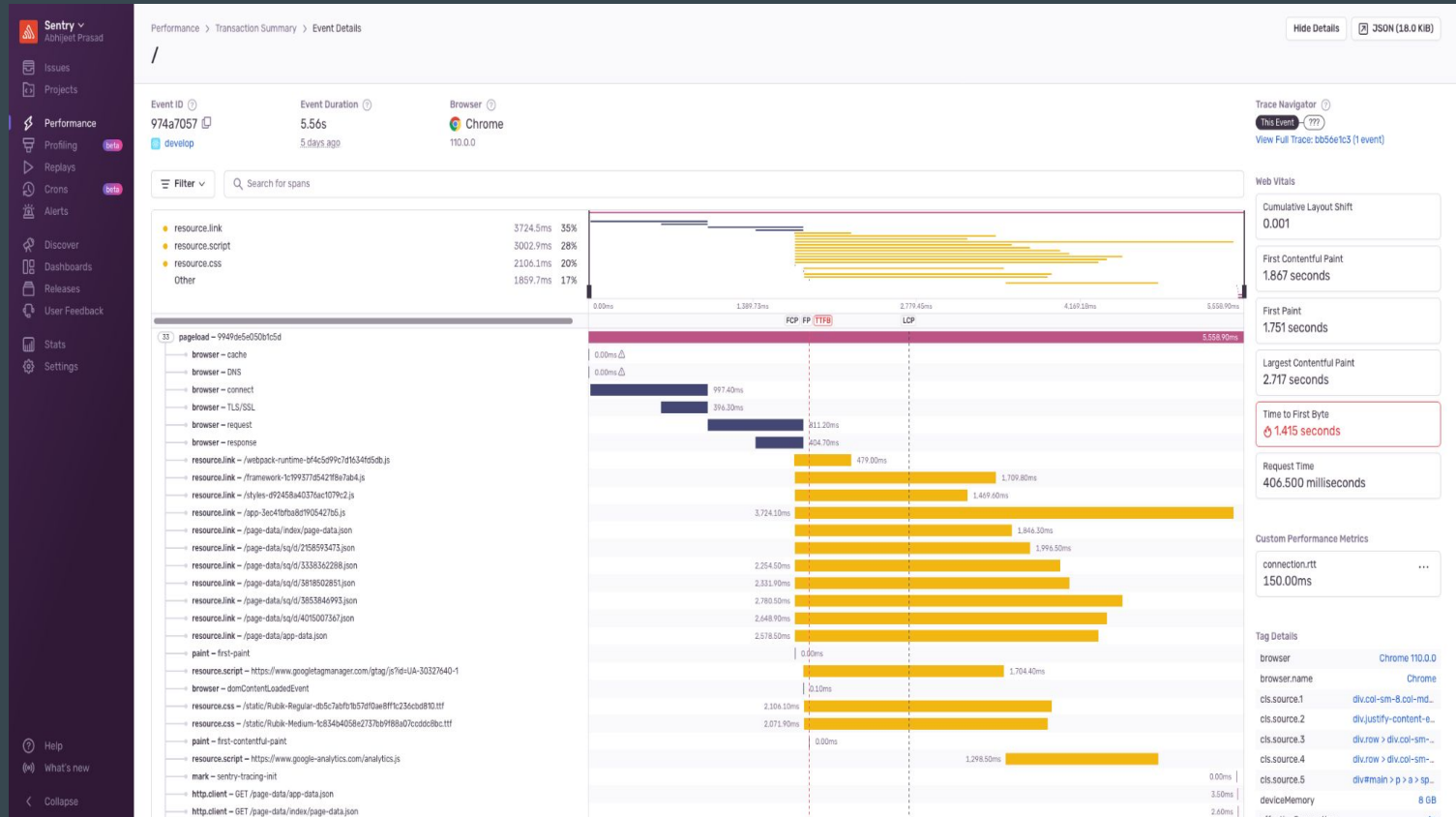
"Для каждого ресурса проверьте использование, насыщенность и ошибки"

Брендан Грегг

Использование: среднее время, в течение которого ресурс был занят обслуживанием работы.

Насыщенность: степень, в которой ресурс имеет дополнительную работу, которую он не может обслужить, часто в очереди.

Ошибки: количество событий, связанных с ошибками.



RED - Rate, Errors and Duration

Rate, Errors & Duration

Rate: количество запросов в секунду, обслуживаемых вашими сервисами.

Errors: количество неудачных запросов в секунду.

Duration: Количество времени , каждый запрос принимает выражаются в виде временного интервала

USERED

U = использование, согласно каноническому определению

S = параллелизм

E = коэффициент ошибок, как метрика пропускной способности

R = Пропускная способность запроса, в запросах в секунду

E = коэффициент ошибок запросов, как метрика пропускной способности или доля от общей пропускной способности.

D = Задержка, время пребывания или время отклика; все три показателя широко используются

Что можем делать

Реактивно

- Тушить пожары
- Успокаивать пользователей
- Терять деньги

Проактивно

- Проектировать изменения
- Внедрять полезные функции
- Уменьшать издержки

Feature flags

USE GIT

Git flow

- Отдельная ветка на каждый чих
- Много контроля
- Высокая ответственность
- Долгоживущие ветки
- Merge hell

Trunk based

- Всё летит в ...
- Много слияний
- Мало контроля (нужны тесты)
- Нагрузка review + CI
- Agile friendly

FeatureManager.cs

```
// Example of consuming feature flags in C#
public class FeatureManager {
    private readonly FeatureFlags _featureFlags;

    public FeatureManager(FeatureFlags featureFlags) {
        _featureFlags = featureFlags;
    }

    public void ExecuteFeatureLogic() {
        if (_featureFlags.IsNewFeatureEnabled()) {
            // Execute new feature logic
        } else {
            // Fallback to default logic
        }
    }
}
```

Флаги значений

[Add Value Flag](#)

Feature key	Наименование	Описание	Активен	Операнд	Приложение	Версии
AGE_KEY	Возраст	Требование возраста	+	And	655dd7c4-85c5-44a7-bb37-a503ca83e302	Edit Удалить
EATIN_MODE	Режим заказа на стол		+	Or	31059241-da65-4006-8ca1-5de497da1f64	Edit Удалить

Логический операнд:

And ▾

Приложение:

Demo app ▾

Версии:

1.0.0 ▲



Наименование:

userAge

Тип отношения:

< ▾

Значение:

62

Наименование:

userAge

Тип отношения:

> ▾

Значение:

21

ARC

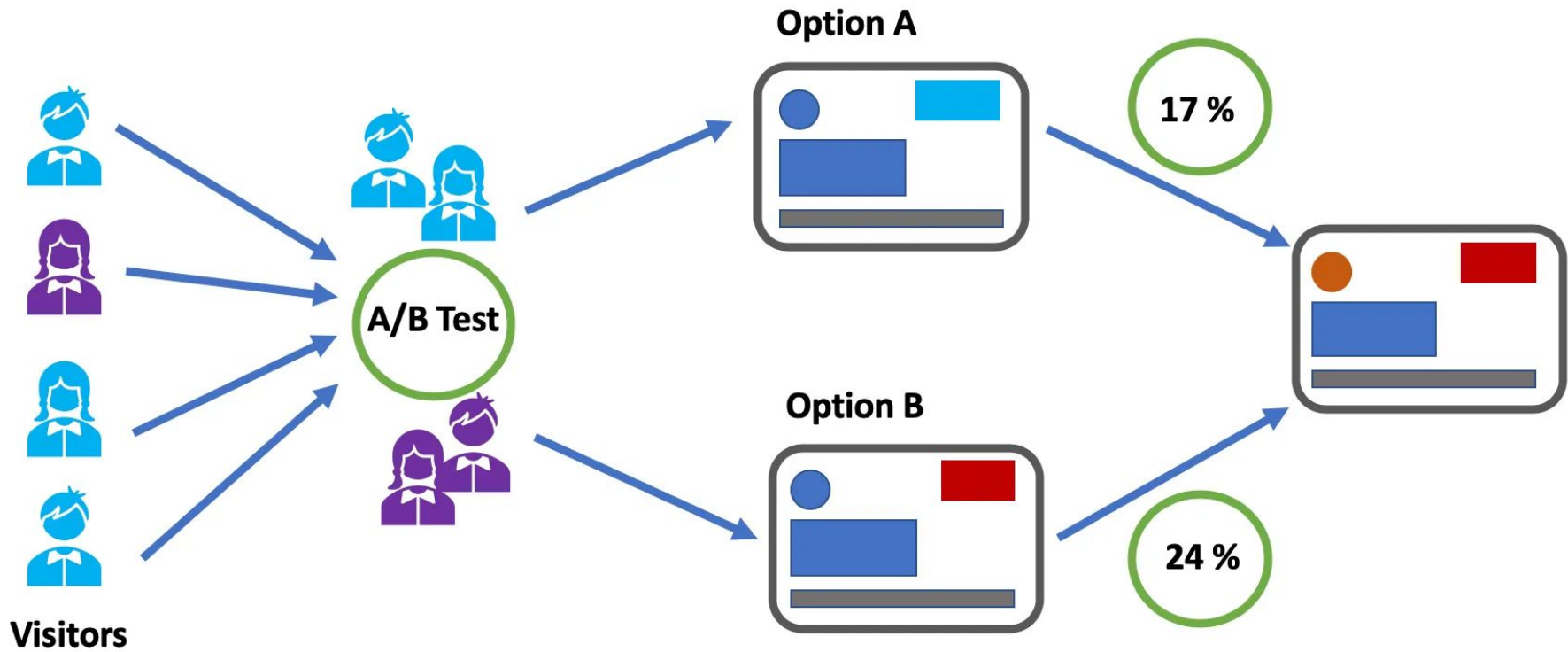


SOURCE





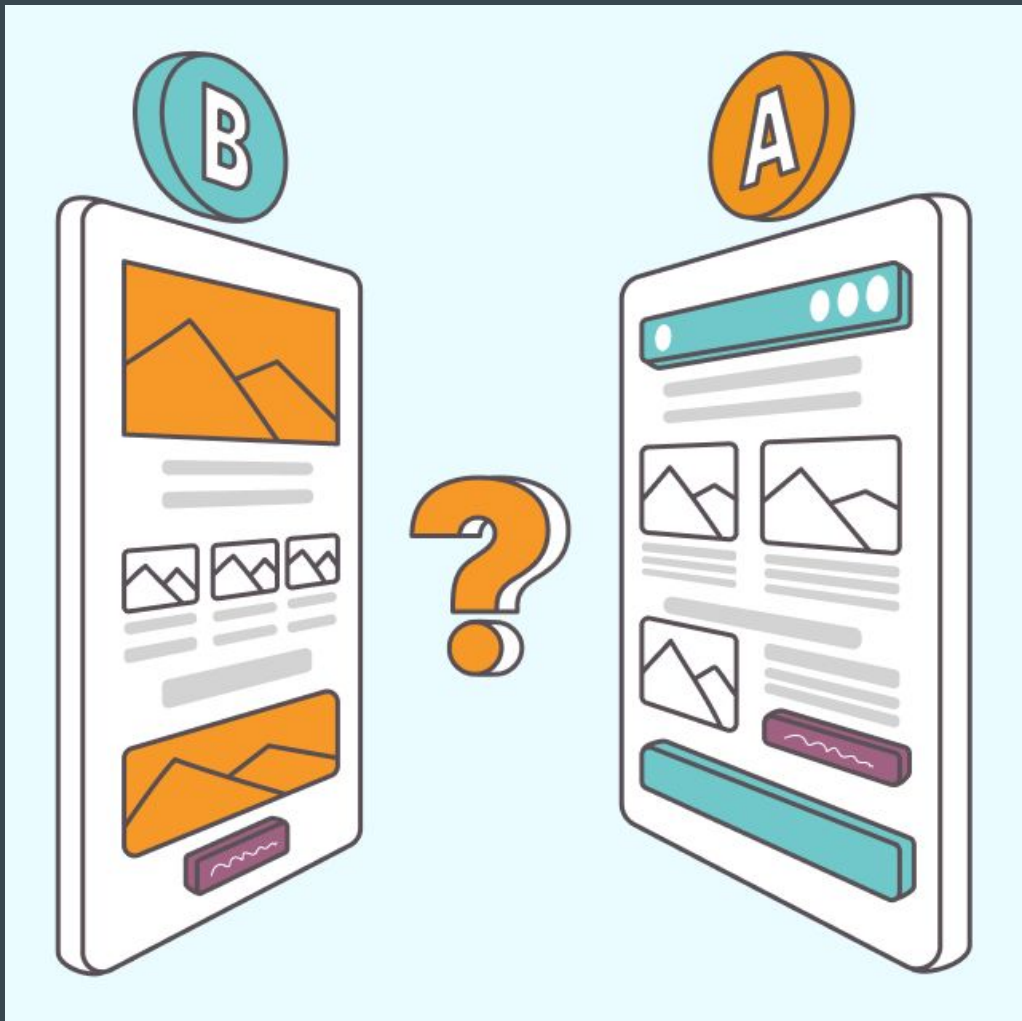
A\B тестирование



Нулевая гипотеза

предполагает, что результаты, А и В на самом деле не отличаются и что наблюдаемые различия случайны. Мы надеемся опровергнуть эту гипотезу.

Альтернативная гипотеза — это гипотеза о том, что В отличается от А, и вы хотите сделать вывод об её истинности.







К прочтению

- <https://logrocon.ru/news/monitoring>
- <https://habr.com/ru/companies/skbkontur/articles/341326/>
- <https://habr.com/ru/articles/233911/>
- <https://education.yandex.ru/journal/kak-provesti-a-b-testirovanie-6-prostykh-shagov>
- <https://www.reg.ru/blog/polnyj-gajd-po-b-testam/>