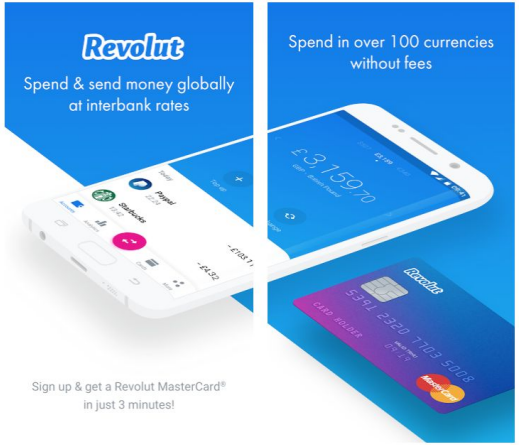
А/Б тестирование, вам понравится!

Revolut, Konstantin Volkov

From Revolut with Love 😍



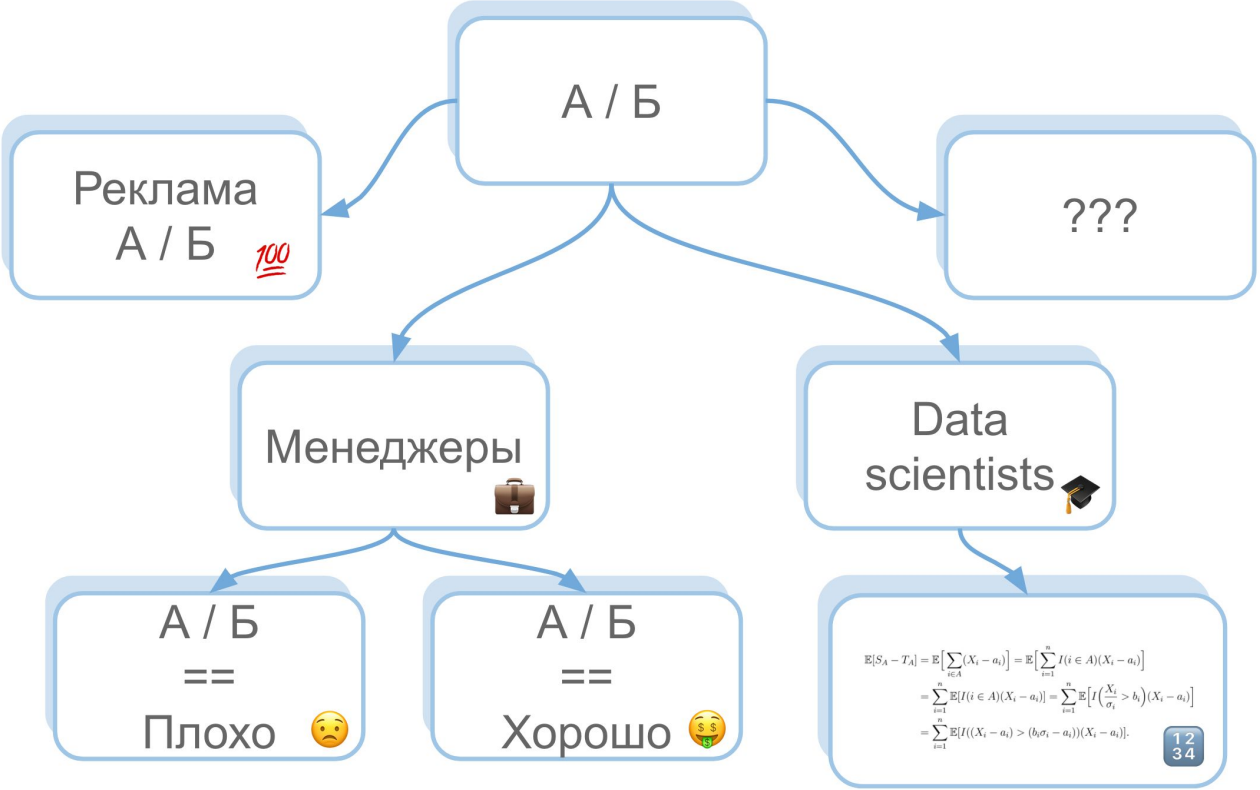
The ~~King's~~ User's Speech

Revolut Community

Sign Up | Log In

- Announcements
- Ideas
- Feedback
- Bugs
- Travelling
- Other Stuff
- Revolut for Business

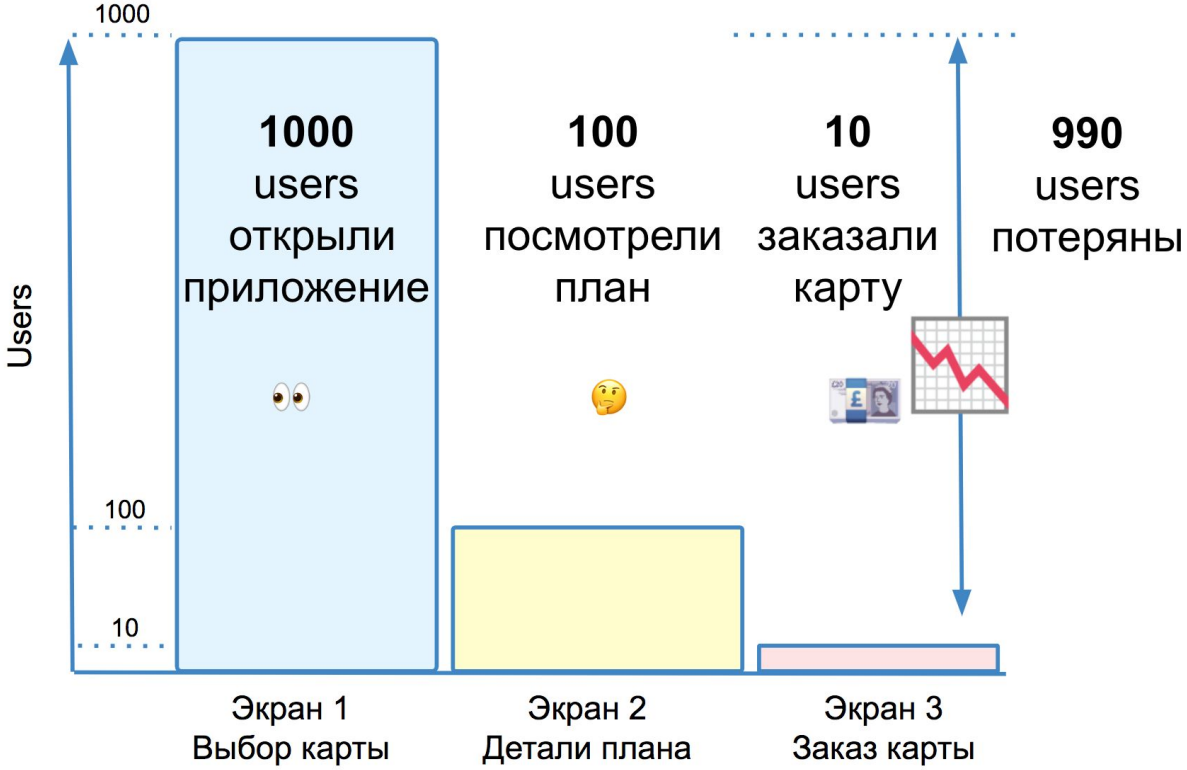
Что вы возможно уже знаете о А/Б тестировании ❓



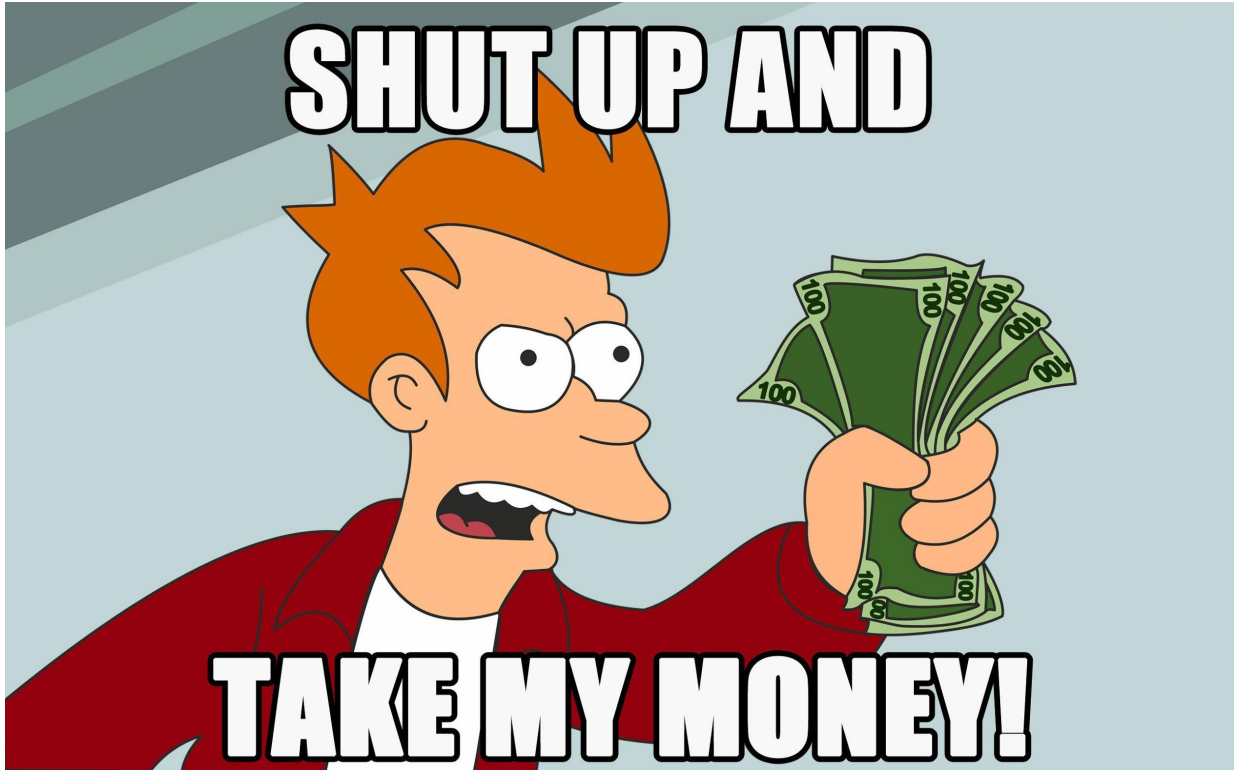
План действий

TO DO	IN PROGRESS	DONE
Что такое А / Б тесты ?		Архитектура
Как разработать А / Б тесты ?		покрытие тестами 80%
Как встроить А / Б тесты в проект		сбор аналитики

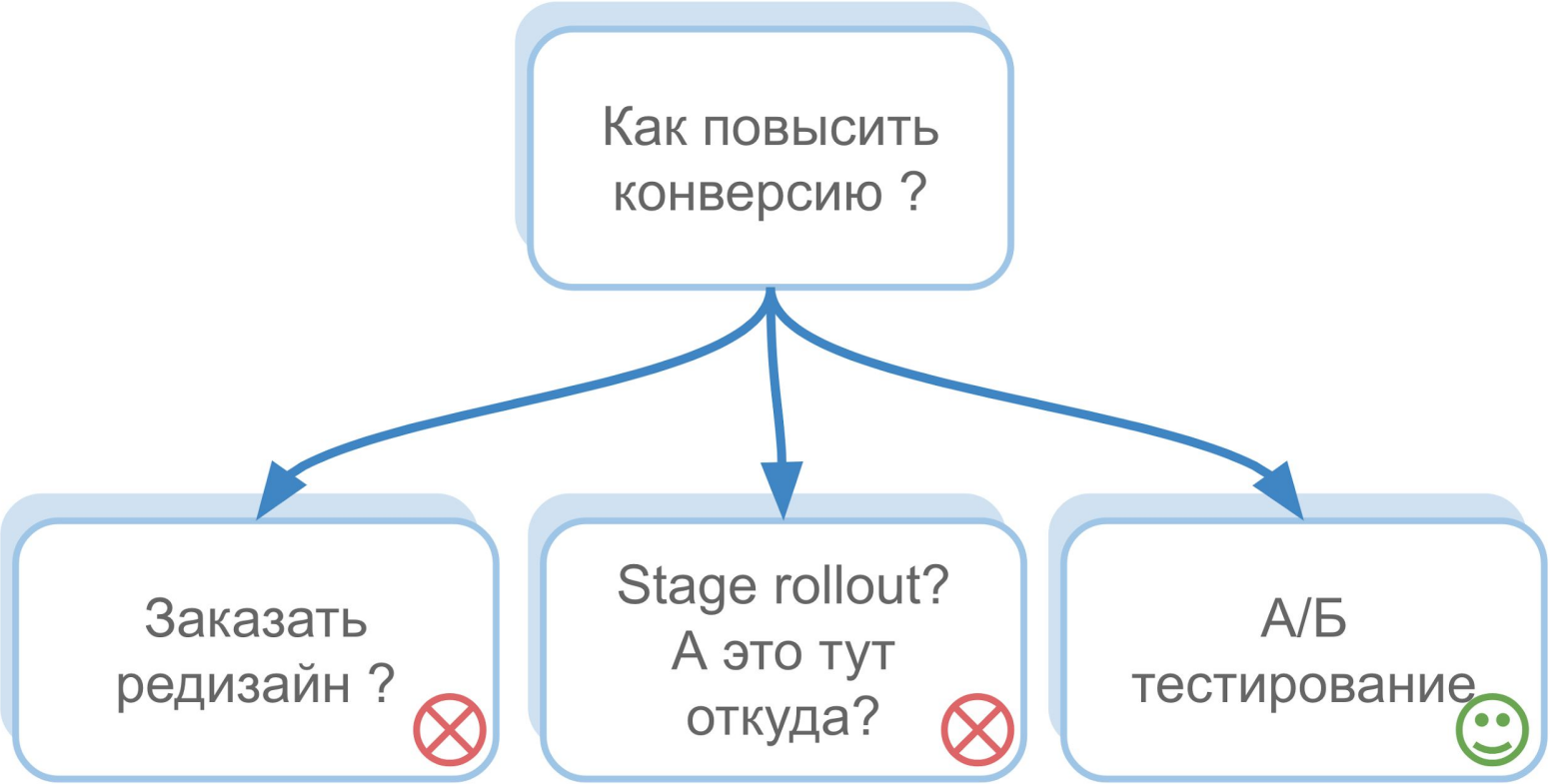
Что не так с этими пользователями



Мнение инвестора



Так что же делать? 🤔



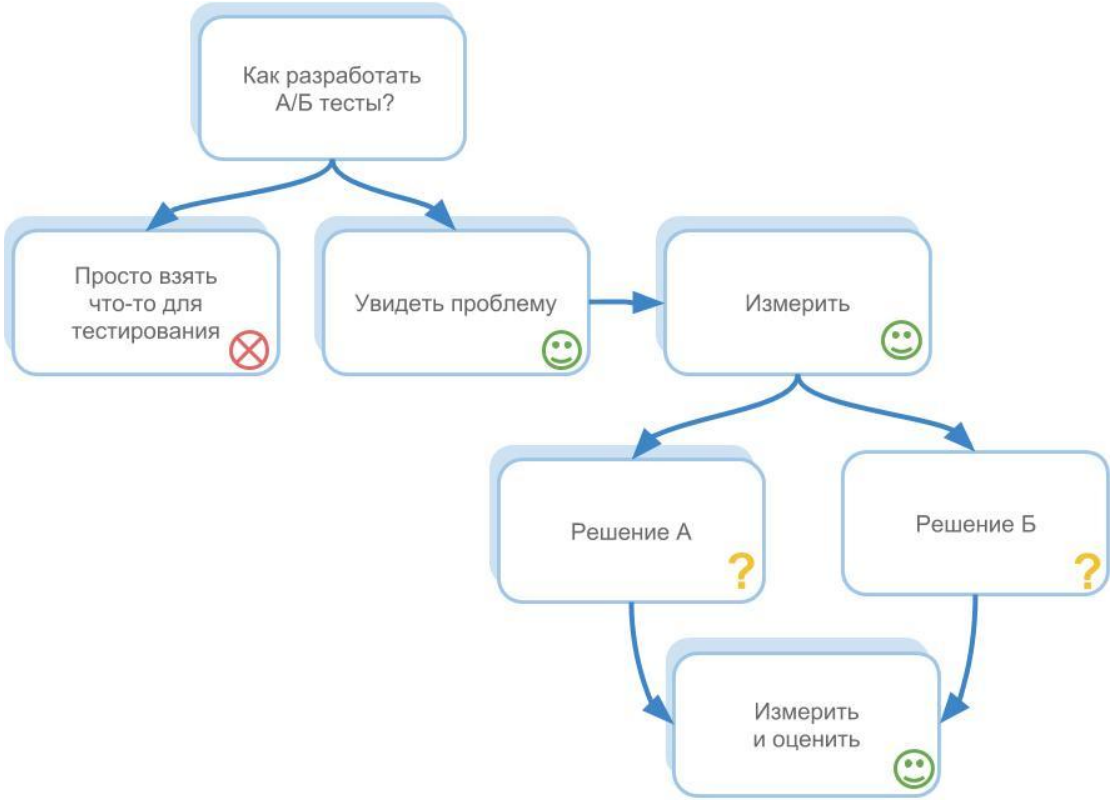
Ок, а как делать А / Б тесты?

О чем думает бизнес?

Цели бизнеса:

Разработка гипотезы

Требуется качество и скорость реализации



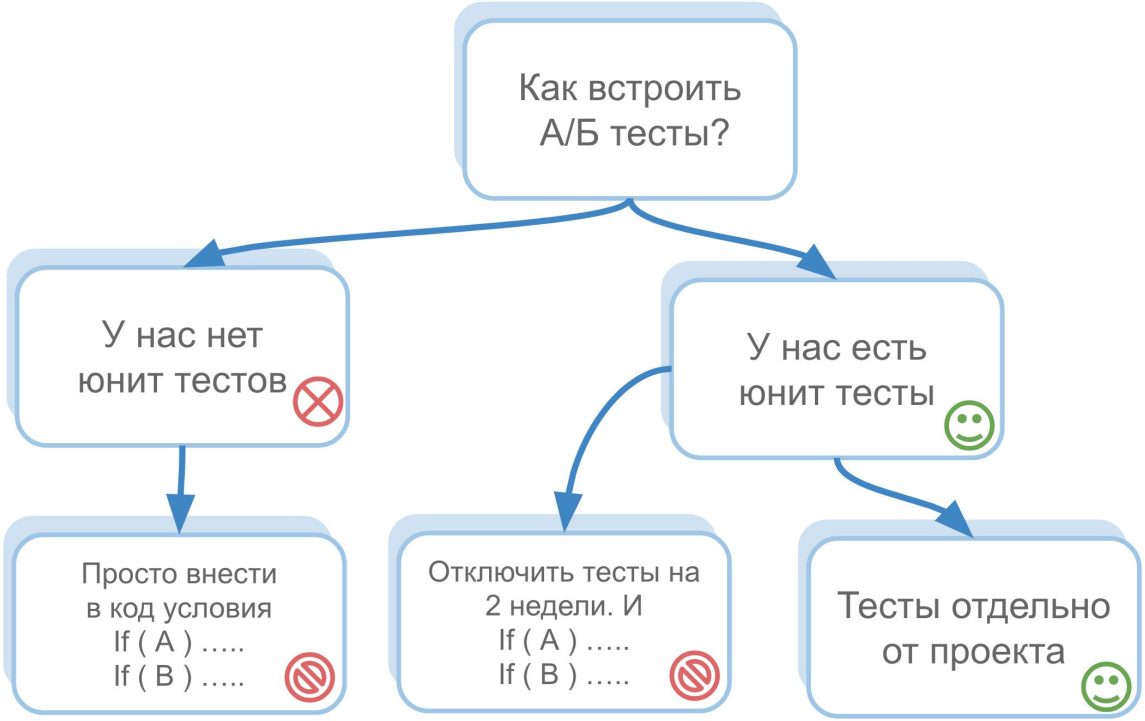
О чем думает разработка?

Цели разработки:

Сохранить текущие Unit tests

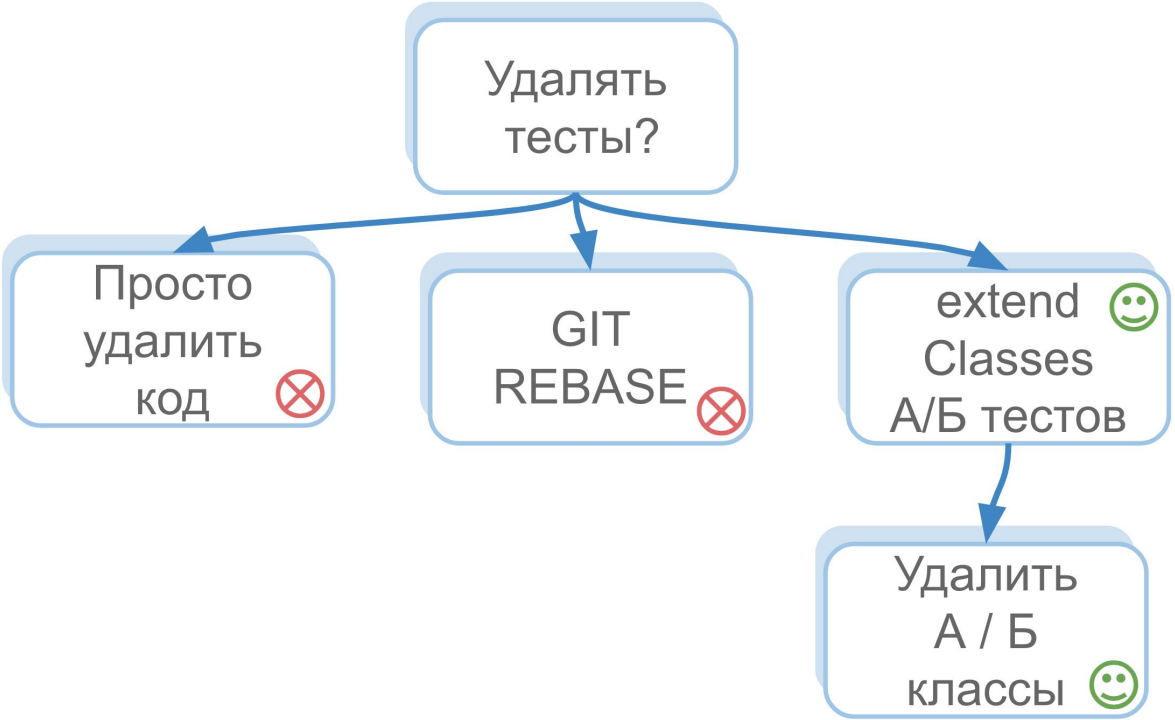
Не мешать работе остальной команды

Корректная реализация



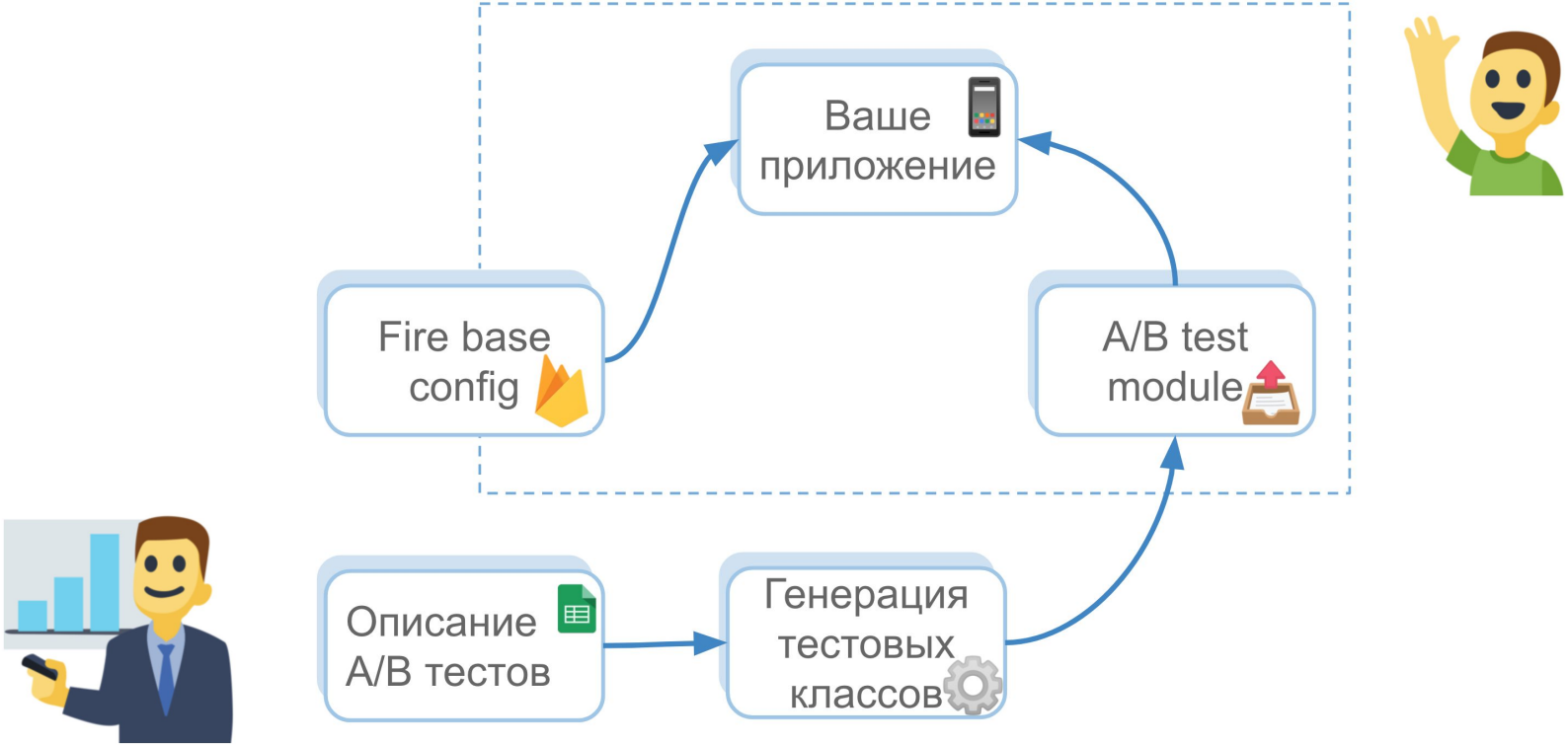
И еще ...

После того как все закончится нужно оставить проект в чистоте.



какие будут
ваши предложения?
memesmix.net

У нас есть решение для Всех



Как работаем Firebase Remote Config?

Make sure you have the latest RemoteConfig SDK installed LEARN MORE

Experiment basics
ShowPromoExperiment
10% of users matching 2 targeting criteria

2 Variants

ADD PARAMETER

		shouldBeSh…	title
33.33%	Control group	true	(no value)
33.33%	Variant A	false	(no value)
33.33%	Variant B	true	You can use R card a

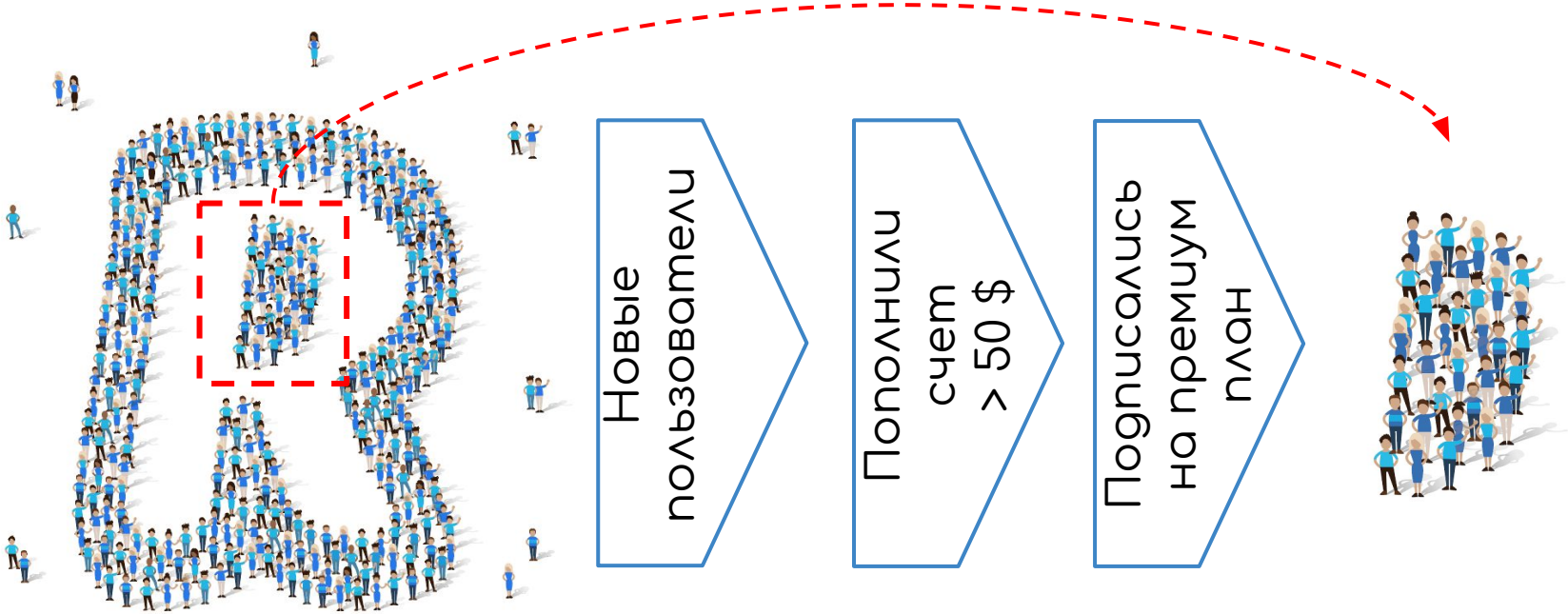
ADD VARIANT

PREVIOUS NEXT

Goal

CANCEL REVIEW

А / Б формируем выборку пользователей? 🚀



Как составить таблицу с описанием А / Б теста?

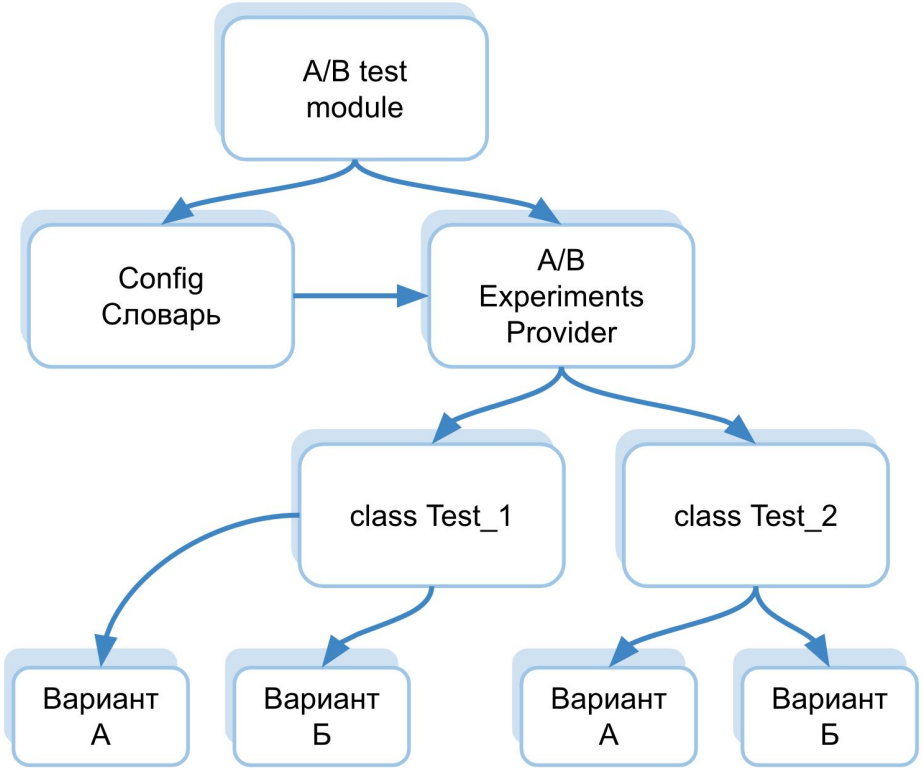
	A	B	C	D	E	F	G
1	**Version**	**Experiment Name**	**Variant Neme**	**Variant description**	**Value Name**	**Value Type**	**Value**
2	**v1.0**	**ShowPromoExperiment**	**Default**	Поведение без изменений	shouldBeShown	Boolean	FALSE
3					showDelay	String	null
4			**A_Experiment**	Показывать промо экран сразу	shouldBeShown	Boolean	TRUE
5					showDelay	String	0
6			**B_Experiment**	Показывать промо экран через день	shouldBeShown	Boolean	TRUE
7					showDelay	String	1d
8		**ButtonTextExperiment**	**Default**	Поведение без изменений	color	String	#000000
9					text	String	Subscribe
10			**A_Experiment**	Красная кнопка "Subscribe!"	color	String	#FF0000
11					text	String	Subscribe!
12			**B_Experiment**	Синяя кнопка "Confirm"	color	String	#0000FF
13					text	String	Confirm

Из чего состоит модуль с тестами?

Config - хранилище конфигурации в формате ключ/значение

Provider для доступа к тестам, который,
который использует конфигурацию для принятия решения

И набор тестовых классов, каждый из которых содержит набор вариантов



Как работает автогенерация?

1) Получение доступа к Google Sheets API. На вход передается cert файл.

2) Инициализация идентификатора таблицы (часть URL вашей таблицы)

3) Инициализация запроса к таблице (специальный язык запросов в Google sheets API)

4) Обработка массива данных. И генерация классов для iOS и Android

```
SCOPES = 'https://www.googleapis.com/auth/spreadsheets.readonly'

CLIENT_SECRET_FILE = 'client_secret.json'

GOOGLE_SHEET_ID = 'YOUR GOOGLE SHEET ID IS HERE'

def main():

    service = getService(CLIENT_SECRET_FILE, SCOPES)

    spreadsheetId = GOOGLE_SHEET_ID

    rangeName = 'abtest!A2:M'

    parseTableToClass(service, spreadsheetId, rangeName)

if __name__ == '__main__':

    main()
```

{{Mustache}} справляется с iOS

```
class {{experimentName}} : Experiment {
    override func name() -> String {
        return \"{{experimentName}}\"
    }
    override func variants() -> Dictionary<String, Variant> {
        return [
        {{#variants}}
        \"{{experimentVariantName}}\" : Variant(name: \"{{experimentVariantName}}\", description:
\"{{experimentVariantDescription}}\", values:
        [{{#experimentValues}}
        VariantValue(name: \"{{experimentValueName}}\",value: \"{{experimentValue}}\"),
        {{/experimentValues}}]),
        {{/variants}}
        ]}
    override func defaultVariant() -> Variant {
        return Variant(name: "DEF_VARIANT", description: "def variant", values:[
        {{#defaultValues}}
        VariantValue(name: \"{{experimentValueName}}\",value: \"{{experimentValue}}\"),
        {{/defaultValues}}])}
}
```

{{Mustache}} справляется с Android

```
enum class {{experimentName}} : BaseTest {
    {{#variants}}
    {{experimentVariantName}} {
        override fun getExperimentVariantDescription(): String =
\"{{experimentVariantDescription}}\"
        override fun getExperimentValue(): String = \"{{experimentVariantConfigValue}}\"
        {{#experimentValues}}
        override fun {{experimentValueName}}(): {{experimentValueType}} = \"{{experimentValue}}\"
        {{/experimentValues}}
    },
    {{/variants}}
    DEF_EXPERIMENT {
    override fun getExperimentVariantDescription(): String = \"default variant\"
    override fun getExperimentValue(): String = \"DEF\"
    };
    override fun getExperimentKey(): String = \"{{experimentName}}\"
    {{#defaultValues}}
    open fun {{experimentValueName}}(): {{experimentValueType}} = \"{{experimentValue}}\'
    {{/defaultValues}}
    };
```

Встраиваем A / Б тесты в проект

Dependency Injection A/B	AB_Activity : Activity A/B	Activity / Fragment / View UTest	Router UTest	AB_Router : Router A/B
	AB_Presenter : Presenter A/B	Presenter / ViewModel UTest		
	AB_Interactor : Interactor A/B	Interactor UTest		
	AB_Repository : Repository A/B	Repository UTest		

А что дальше?

Слушайте ваших пользователей

Ищите простые способы решения проблем

Автоматизируйте процесс

Мало написать тесты - важно их сохранить, чтобы они служили вам долго

Чуть не забыл

Спасибо !

Ссылка на исходники:
https://github.com/kvrus

Модуль + тестовое приложение