

avito.tech

Москва — 2021

Как pull request'ы крадут ваше время!

Екатерина Батеева

Екатерина Батеева

7 лет в тестировании

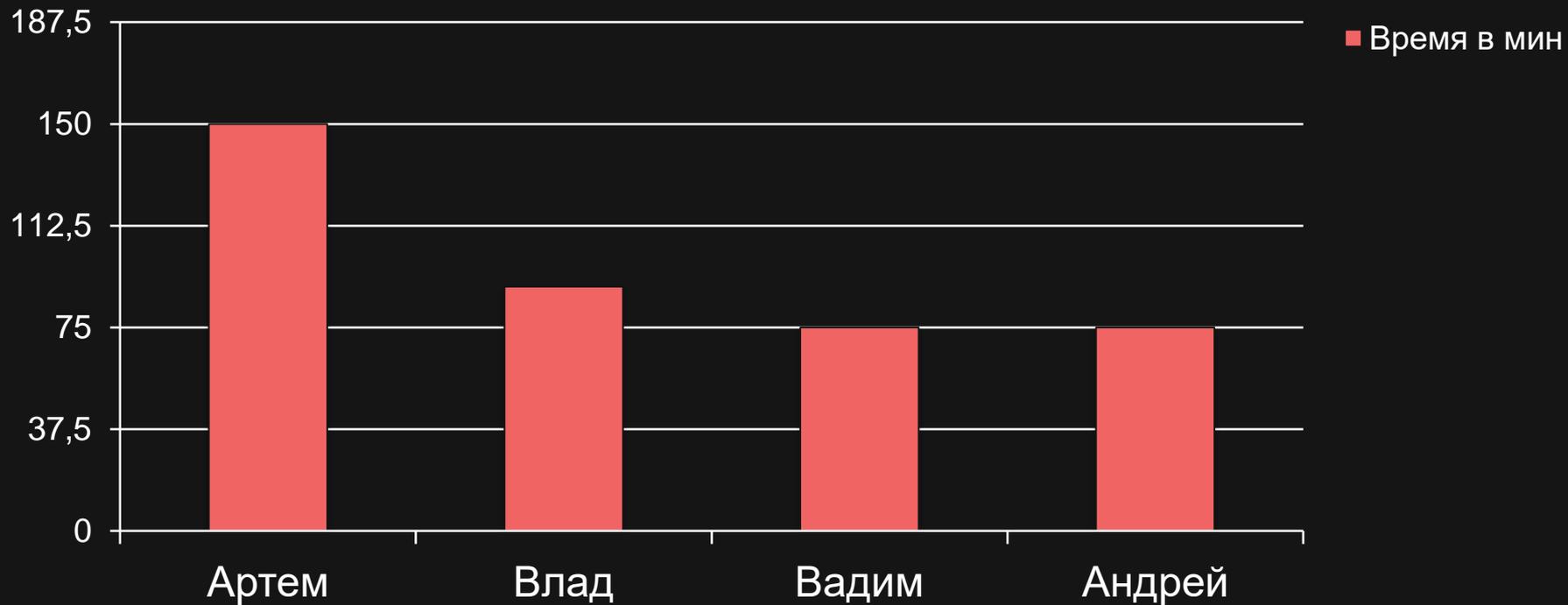
2,5 года в iOS разработке



МИФ 1. Посмотреть PR всего 15 минут



Топ reviewer'ов за день



Посмотреть PR всего 15 минут



Ограничьтесь
часом

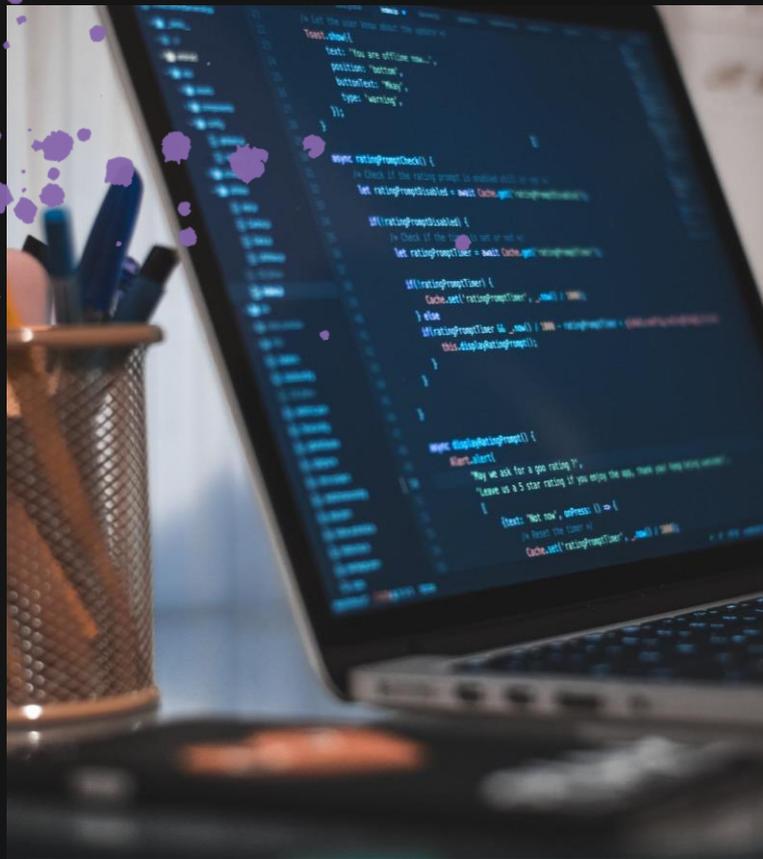


< 400 строк



25% времени

МИФ 2. Нам нужно много правил



Нам нужно много правил

01. Сколько пробелов в отступе
02. На какой строке открывающая скобка
03. Лишние пустые строки и прочее...

Нам нужно много правил



swift-format

Swift-format

Пробелы на любой вкус

```
"indentation": {
```

```
  "spaces": 2
```

```
}
```

Swift-format

Каждая переменная на своей строке

```
"lineBreakBeforeEachArgument": true
```

Swift-format

Комментарии не пройдут

```
"NoBlockComments": true,
```

Swift-format

```
struct DeliveryAddress: Codable {  
    var streetAddress: String  
    var region: String;var postalCode: String  
    var country: String  
  
    init( streetAddress: String,  
          region: String, postalCode: String, country: String    ){
```

Swift-format

```
struct DeliveryAddress: Codable {  
    var streetAddress: String  
    var region: String;  
    var postalCode: String  
    var country: String  
  
    init(  
        streetAddress: String,  
        region: String,  
        postalCode: String,  
        country: String  
    ) {
```

Swift-format

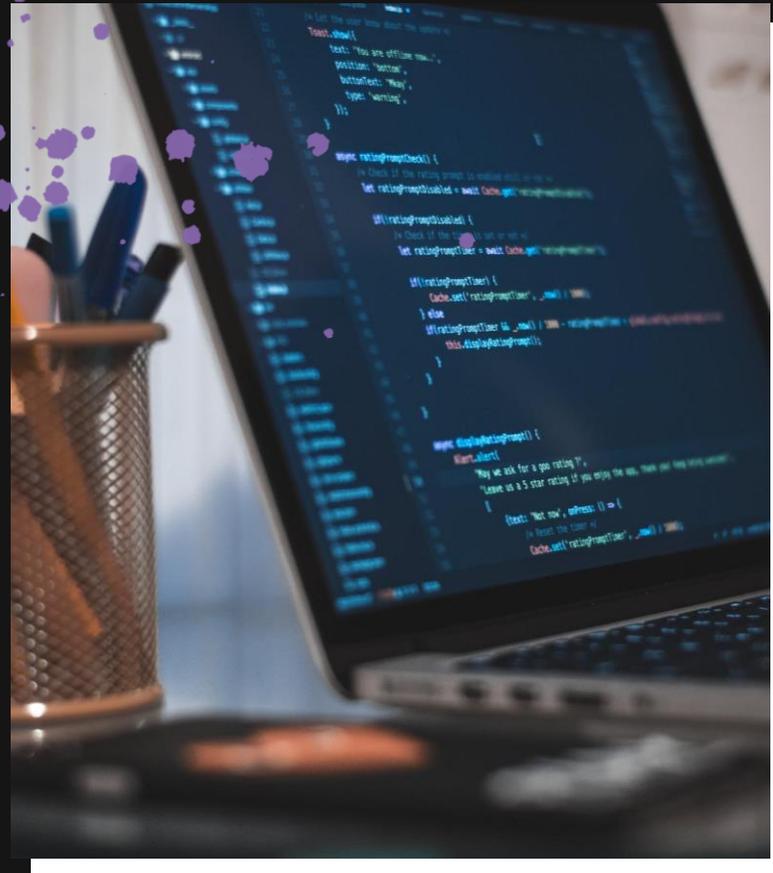
Плюсы

- ▶ Не тратим время на проверку форматирования
- ▶ Библиотека от Apple

Минусы

- ▶ Непонятны проблемы
- ▶ Большой проект обрабатывает долго

МИФ 3. Линтер нас спасет



255

Правил

Проверка линтером

5

Начните с
этого

Ну или хотя бы с 1

0

Warnings —
Ваша цель

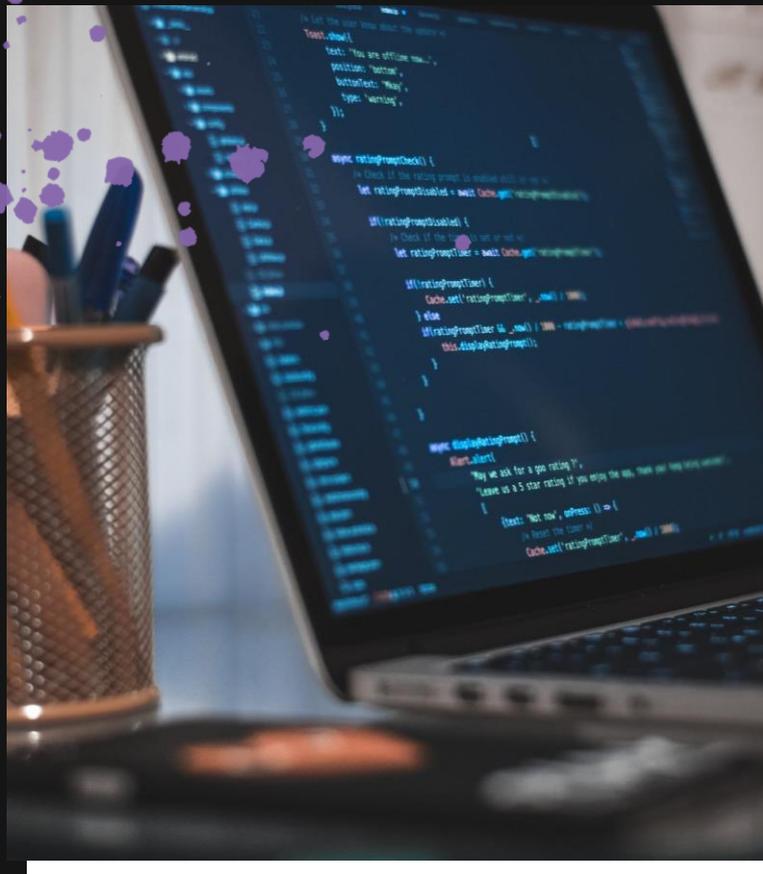
Локально отключать -
нежелательно

Топ-5 правил линтера

- ▶ `file_length`
- ▶ `function_body_lenth`
- ▶ `warning_threshold`
- ▶ `empty_xctest_method`
- ▶ `weak_delegate`



МИФ 4. Unit-тесты не нужны

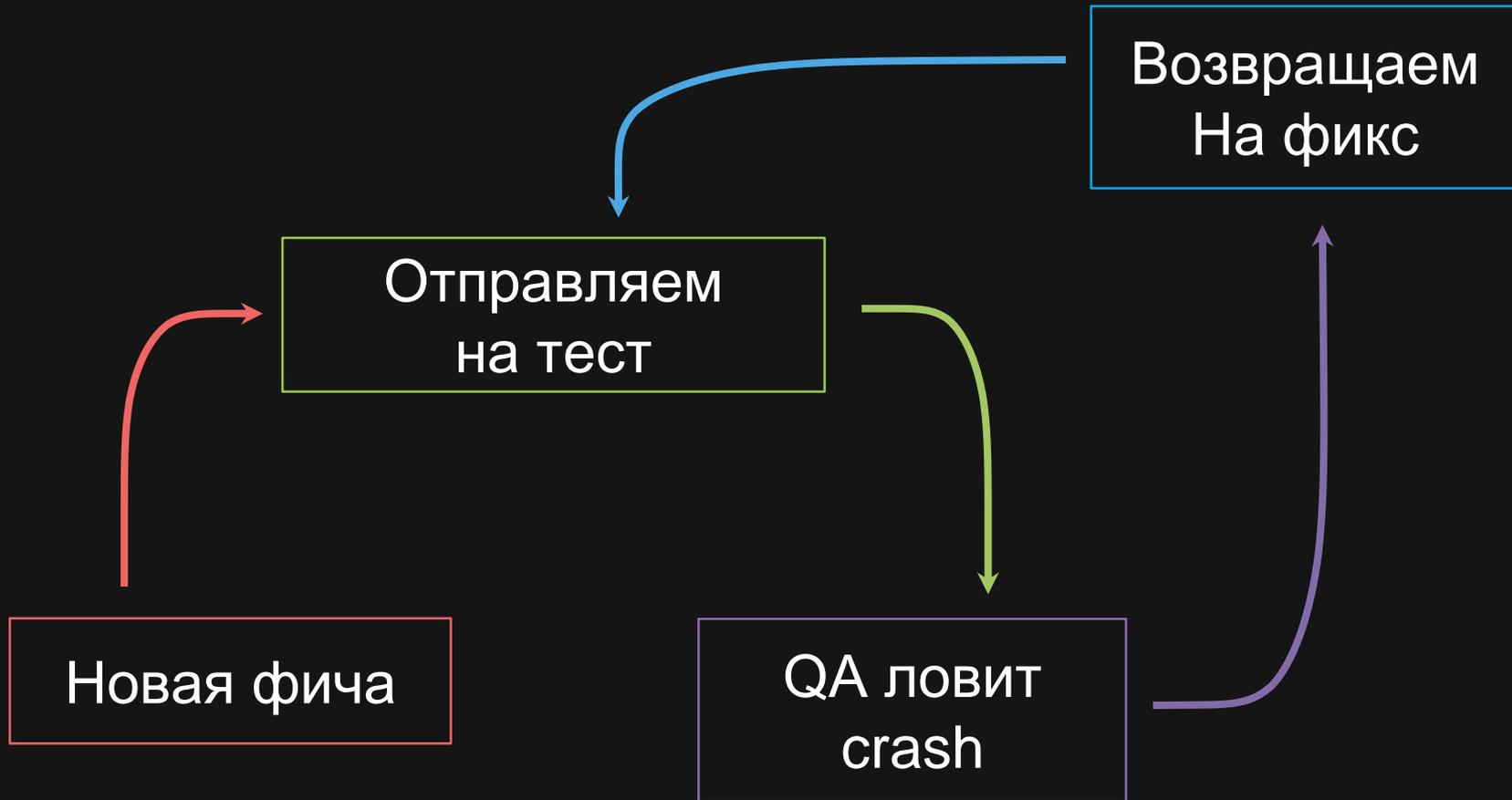


Новая фича

Новая фича

Отправляем
на тест





Unit-тесты не нужны - Проверяем JSON

```
func test_bookingResponse_unboxes_whenResponseIsSuccess() {  
    let resource = "BookingResponse"  
    typealias Type = BookingResponse  
  
    guard let response = unbox(from: resource) as Type? else {  
        XCTFail("Fail to unbox response")  
    }  
}
```

Unit-тесты не нужны - Проверяем DI

```
allAssemblies=$(search  
    "${avitoFolder}"  
    "*.Assembly(Impl)?\.swift$"  
    «+DITestableAssembly" "" "check")  
allAssembliesCount=$(printf "%s" "${allAssemblies}" | wc -l)
```

Unit-тесты не нужны

Проверяем DI контейнер

```
allTestableAssemblies=$(search  
    "${avitoFolder}"  
    ". *Assembly(Impl)?\+DITestableAssembly\.swift$" ""  
    "+DITestableAssembly" "")  
allTestableAssembliesCount=  
    $(printf "%s" "${allTestableAssemblies}" | wc -l)
```

Unit-тесты не нужны

Находим «уникальные» assembly

```
uniqAssemblies=$(printf "%s\n%s" "${allAssemblies}"  
"${allTestableAssemblies}" | sort | uniq -u )
```

Кодогенерация



Кодогенерация



Custom templates

Choose a template for your new file:

iOS macOS watchOS tvOS Filter

Autoteka

Avito

				
Class	ClickStreamEvent	Protocol+Impl	Dynamic Publish Slot Module	Module
				
Service	Module+Collection	Module+FormCollection	Module+FormCollection+Deeplinks	AvitoTheme Custom Styles
				
NetworkRequest	CurrentFlowProvider	DeepLinkDrivenModuleRegistrar	DeepLinkFactory	DITestableAssembly

Cancel Previous **Next**

Choose options for your new file:

New Module Name:

Interface Name: __VARIABLE_moduleName__Module

View Controller Name: __VARIABLE_moduleName__ViewController

View Name: __VARIABLE_moduleName__View

Interactor Name: __VARIABLE_moduleName__Interactor

Presenter Name: __VARIABLE_moduleName__Presenter

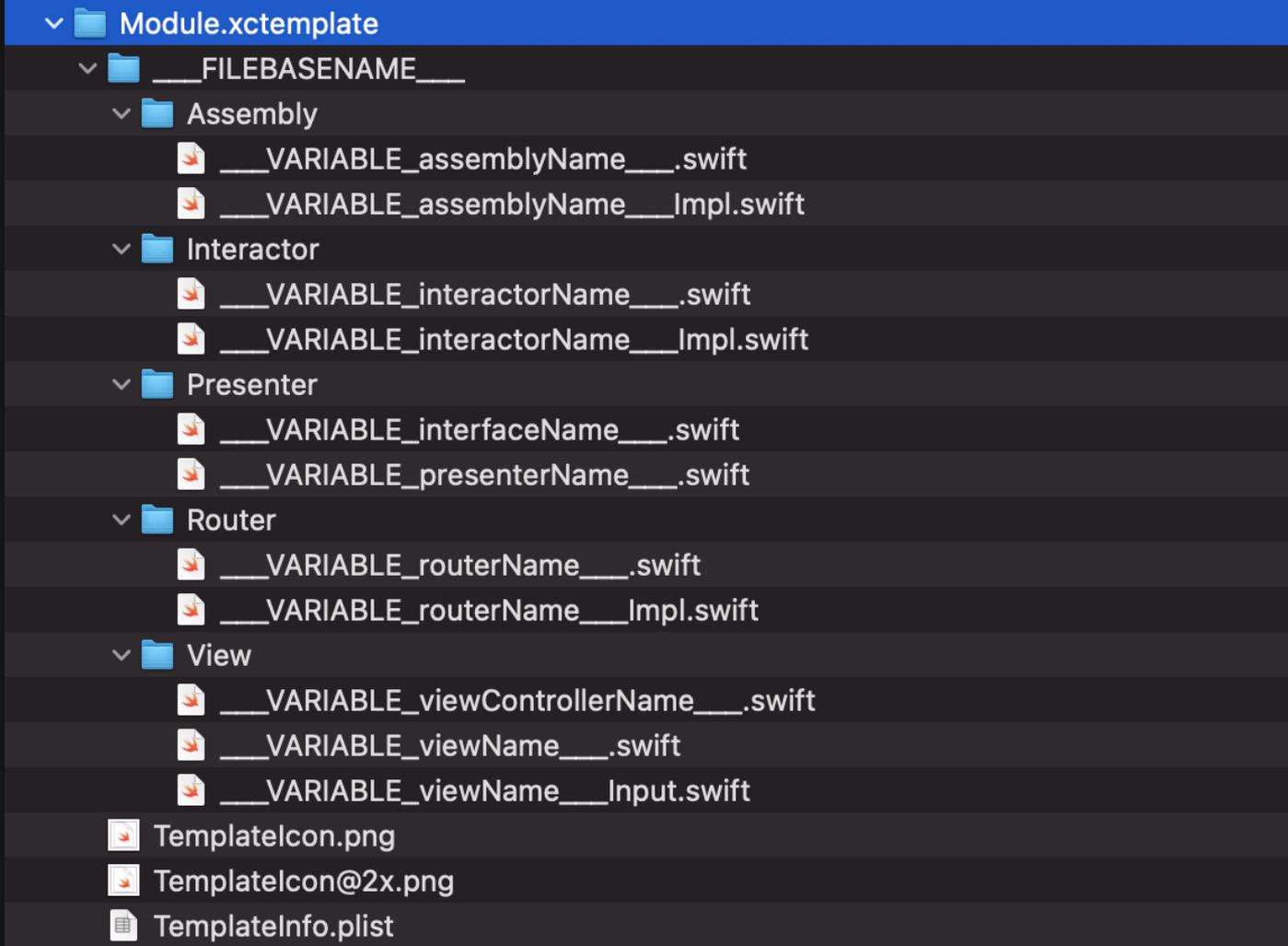
Router Name: __VARIABLE_moduleName__Router

Assembly Name: __VARIABLE_moduleName__Assembly

Cancel

Previous

Next



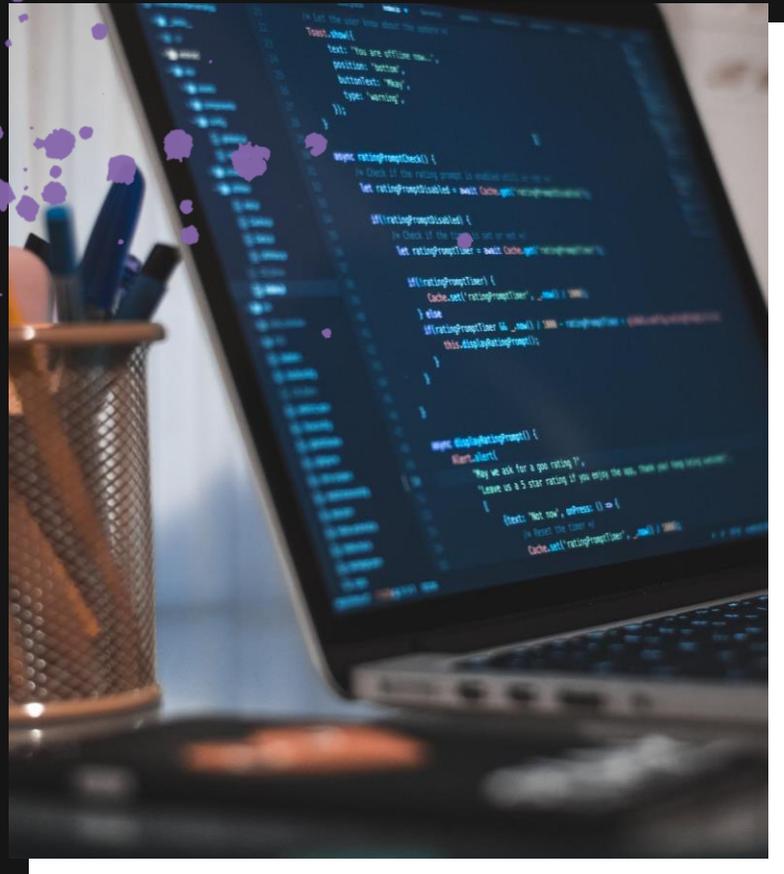
TemplateInfo.plist

TemplateInfo.plist > No Selection

Key	Type	Value
Information Property List	Dictionary	(5 items)
Kind	String	Xcode.IDEKit.TextSubstitutionFileTemplateKind
Options	Array	(9 items)
Item 0	Dictionary	(6 items)
Description	String	The name of the module to create
Identifier	String	moduleName
Name	String	New Module Name:
NotPersisted	Boolean	1
Required	Boolean	1
Type	String	text
Item 1	Dictionary	(3 items)
Default	String	__VARIABLE_moduleName__
Identifier	String	productName
Type	String	static
Item 2	Dictionary	(6 items)
Default	String	__VARIABLE_moduleName__Module
Description	String	The interface name
Identifier	String	interfaceName
Name	String	Interface Name:
Required	Boolean	1
Type	String	static
Item 3	Dictionary	(6 items)
Default	String	__VARIABLE_moduleName__ViewController
Description	String	The view controller name
Identifier	String	viewControllerName
Name	String	View Controller Name:
Required	Boolean	1
Type	String	static
Item 4	Dictionary	(6 items)
Item 5	Dictionary	(6 items)
Item 6	Dictionary	(6 items)
Item 7	Dictionary	(6 items)
Item 8	Dictionary	(6 items)
Description	String	Viper module description
Summary	String	Viper module summary
SortOrder	String	11

```
___VARIABLE_presenterName___.swift
___VARIABLE_presenterName___.swift } No Selection
1 import AvitoPresentation
2 import Dialogs
3
4 final class ___VARIABLE_presenterName___:
5     ___VARIABLE_interfaceName___
6 {
7     // MARK: - Private properties
8     private let interactor: ___VARIABLE_interactorName___
9     private let router: ___VARIABLE_routerName___
10    private let messageDisplayable: MessageDisplayable
11    private let activityDisplayable: ActivityDisplayable
12    private let screenPerformanceMeasurer: ScreenPerformanceMeasurer
13
14    // MARK: - Init
15    init(
16        interactor: ___VARIABLE_interactorName___,
17        router: ___VARIABLE_routerName___,
18        messageDisplayable: MessageDisplayable,
19        activityDisplayable: ActivityDisplayable,
20        screenPerformanceMeasurer: ScreenPerformanceMeasurer)
21    {
22        self.interactor = interactor
23        self.router = router
24        self.messageDisplayable = messageDisplayable
25        self.activityDisplayable = activityDisplayable
26        self.screenPerformanceMeasurer = screenPerformanceMeasurer
27    }
28
29    // MARK: - Weak properties
30    weak var view: ___VARIABLE_viewName___Input? {
31        didSet {
32            setUpView()
33        }
34    }
35
36    // MARK: - Private
37    private func setUpView() {
38        /* set up your view callbacks here
39        view?.onViewDidLoad = { [weak self] in
40
41        }
42        */
43    }
44
```

Проверка утечек памяти



Проверка утечек памяти

Создаем performance test для той функциональности, на которой хотим тестировать утечку



Ktrace files



Memory graphs

Проверка утечек памяти

```
$ xcodebuild test  
-project AvitoApp.xcodeproj  
-scheme PerformanceTests  
-destination platform=iOS, name="iPhone"  
-enablePerformanceTestsDiagnostics YES
```

Проверка утечек памяти



Pre memgraph



Post memgraph

Проверка утечек памяти

Tests	Duration	Memory
✓ SkyengDictionaryTests SkyengDictionaryTests 1 passed (50 %), 1 failed (50 %) in 0,06s		
✓ testExample()	0,01s	
✗ testPerformanceExample()	0,05s	
✗ Performance Regression at SkyengDictionaryTests.swift:28: failed: Memory Physical measurement #3 is 150% worse than average (max allowed: 40%).		
✓ SkyengDictionaryUITests SkyengDictionaryUITests 2 passed (100 %) in 40s		
> ✓ testExample()	2s	
✓ testLaunchPerformance()	38s	
Start Test at 2021-10-03 22:11:05.810 (Start)		
Some screenshots were deleted because testing is configured to remove automatic screenshots on success. (0.00s)		
Set Up (0.03s)		
> Open com.neifmetus.SkyengDictionary (0.04s)		
> Tap "Введите слово" TextField (5.07s)		
> Type 'table' into "Введите слово" TextField (6.07s)		
> Open com.neifmetus.SkyengDictionary (6.45s)		
> Tap "Введите слово" TextField (11.51s)		
> Type 'table' into "Введите слово" TextField (12.42s)		
> Open com.neifmetus.SkyengDictionary (12.78s)		
> Tap "Введите слово" TextField (17.89s)		
> Type 'table' into "Введите слово" TextField (18.81s)		
> Open com.neifmetus.SkyengDictionary (19.18s)		
> Tap "Введите слово" TextField (24.14s)		
> Type 'table' into "Введите слово" TextField (25.14s)		
> Open com.neifmetus.SkyengDictionary (25.55s)		
> Tap "Введите слово" TextField (30.56s)		
> Type 'table' into "Введите слово" TextField (31.42s)		
> Open com.neifmetus.SkyengDictionary (31.72s)		
> Tap "Введите слово" TextField (36.68s)		
> Type 'table' into "Введите слово" TextField (37.54s)		
Tear Down (37.90s)		

```

memgraphs appledev$ leaks post_Meal\ Planner_2021-05-03-083219.GpxoEC.memgraph
Hardware Model:  iPhone10,5
Process:         Meal Planner [8860]
Path:           /Applications/Meal Planner.app/Meal Planner
Load Address:   0x100000000
Identifier:     Meal Planner
Version:       ???
Code Type:     ARM64
Platform:      iOS
Parent Process: debugserver [8859]

Date/Time:      2021-05-03 08:32:15.125 -0700
Launch Time:    2021-05-03 08:32:05.191 -0700
OS Version:     iPhone OS 15.0 (19A221b)
Report Version: 104

```

```

Physical footprint:      125.0M
Physical footprint (peak): 125.0M
-----

```

```

leaks Report Version: 4.0, multi-line stacks
Process 8860: 29980 nodes malloced for 16221 KB
Process 8860: 4 leaks for 240 total leaked bytes.

```

```

STACK OF 1 INSTANCE OF 'ROOT CYCLE: <MenuItem>':

```

```

12 libsystem_pthread.dylib      0x1e196eb28 start_wqthread + 8
11 libsystem_pthread.dylib      0x1e196b778 _pthread_wqthread + 224
10 libdispatch.dylib            0x18bc93ea0 _dispatch_worker_thread2 + 116
9  libdispatch.dylib            0x18bc93808 _dispatch_root_queue_drain + 644
8  libdispatch.dylib            0x18bc96f10 _dispatch_client_callout + 16
7  libdispatch.dylib            0x18bcae628 _dispatch_call_block_and_release + 24
6  Meal Planner                  0x100012480 thunk for @escaping @callee_guaranteed () -> () + 48
5  Meal Planner                  0x10003fde4 closure #1 in MKTCustomMealPlannerCollectionViewCell.saveMeal(:) + 804
4  Meal Planner                  0x1000408dc MKTCustomMealPlannerCollectionViewCell.populateMealData() + 1124

```

Parent Process: debugserver [8859]

Date/Time: 2021-05-03 08:32:15.125 -0700

Launch Time: 2021-05-03 08:32:05.191 -0700

OS Version: iPhone OS 15.0 (19A221b)

Report Version: 104

Physical footprint: 125.0M

Physical footprint (peak): 125.0M

leaks Report Version: 4.0, multi-line stacks

Process 8860: 29980 nodes malloced for 16221 KB

Process 8860: 4 leaks for 240 total leaked bytes.

STACK OF 1 INSTANCE OF 'ROOT CYCLE: <MenuItem>':

```

12 libsystem_pthread.dylib      0x1e196eb28 start_wqthread + 8
11 libsystem_pthread.dylib      0x1e196b778 _pthread_wqthread + 224
10 libdispatch.dylib            0x18bc93ea0 _dispatch_worker_thread2 + 116
9  libdispatch.dylib            0x18bc93808 _dispatch_root_queue_drain + 644
8  libdispatch.dylib            0x18bc93808 _dispatch_root_queue_drain + 644
7  libdispatch.dylib            0x18bc93808 _dispatch_root_queue_drain + 644
7  libdispatch.dylib            0x18bcaef610 _dispatch_client_callout + 16
7  libdispatch.dylib            0x18bcaef610 _dispatch_client_callout + 16
7  libdispatch.dylib            0x18bcaef610 _dispatch_client_callout + 16
6  Meal Planner                  0x100012480 think for @escaping @callee_guaranteed () -> () + 48
5  Meal Planner                  0x10003fde4 closure #1 in MKTCustomMealPlannerCollectionViewCell.saveMeal(:) + 804
4  Meal Planner                  0x1000408dc MKTCustomMealPlannerCollectionViewCell.populateMealData() + 1124
3  Meal Planner                  0x100005860 MenuItem.__allocating_init(imageAsset:title:subtitle:) + 64
2  libswiftCore.dylib            0x19023eec4 swift_allocObject + 52
1  libswiftCore.dylib            0x19023ed0c swift_slowAlloc + 60
0  libsystem_malloc.dylib        0x19bf36da8 _malloc_zone_malloc + 152

```

====

4 (240 bytes) ROOT CYCLE: <MenuItem 0x11031b160> [80]

3 (160 bytes) plans --> ROOT CYCLE: <Swift._ContiguousArrayStorage<Swift.Optional<Meal_Planner.MealPlan>> 0x11030ff70> [48]

2 (112 bytes) countAndCapacity + 16 --> ROOT CYCLE: <MealPlan 0x11030ffc0> [64]

1 (48 bytes) meals --> ROOT CYCLE: <Swift._ContiguousArrayStorage<Meal_Planner.MenuItem> 0x11030f420> [48]

countAndCapacity + 16 --> CYCLE BACK TO <MenuItem 0x11031b160> [80]

Parent Process: debugserver [8859]

Date/Time: 2021-05-03 08:32:15.125 -0700

Launch Time: 2021-05-03 08:32:05.191 -0700

OS Version: iPhone OS 15.0 (19A221b)

Report Version: 104

Physical footprint: 125.0M

Physical footprint (peak): 125.0M

Leaks Report Version: 4.0, multi-line stacks

Process 8860: 29980 nodes malloced for 16221 KB

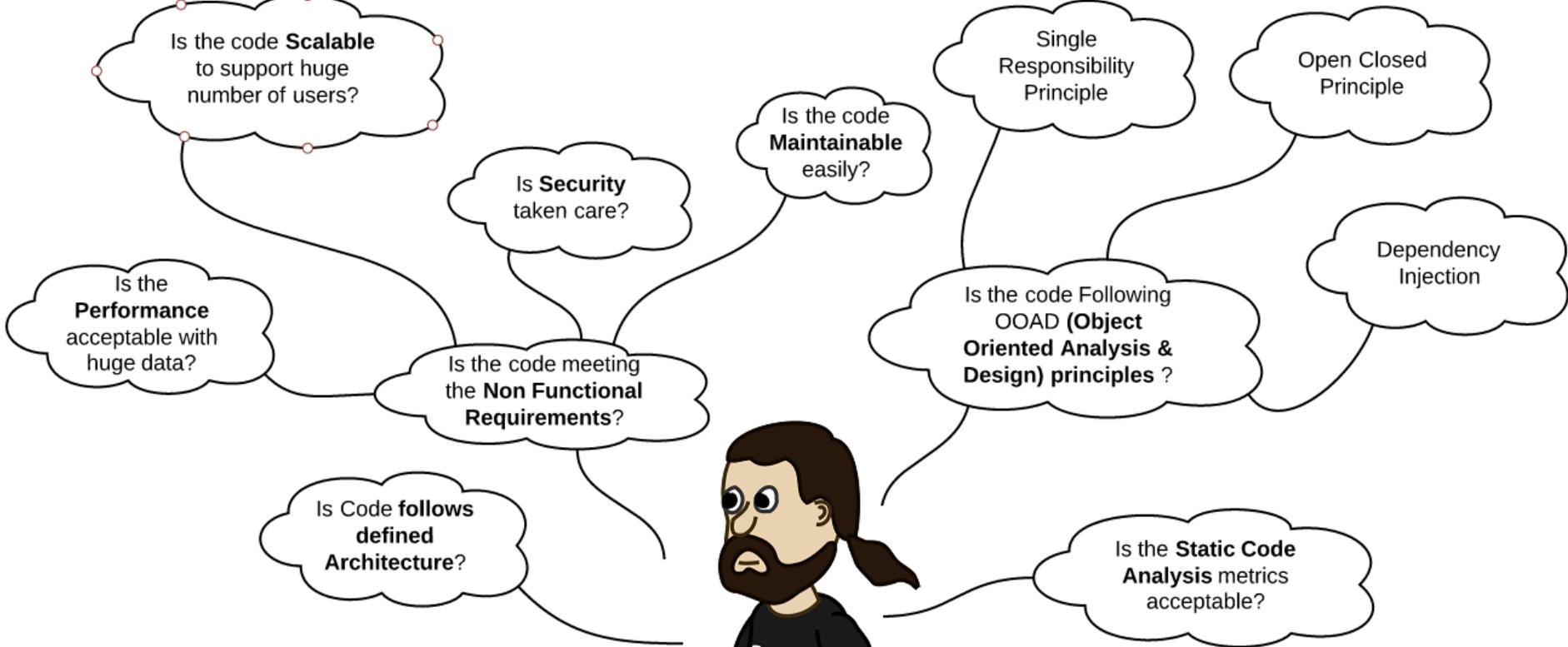
Process 8860: 4 leaks for 240 total leaked bytes.

STACK OF 1 INSTANCE OF 'ROOT CYCLE: <MenuItem>':

```
12 libsystem_pthread.dylib      0x1e196eb28 start_wqthread + 8
11 libsystem_pthread.dylib      0x1e196b778 _pthread_wqthread + 224
10 libdispatch.dylib            0x18bc93ea0 _dispatch_worker_thread2 + 116
9 libdispatch.dylib            0x18bc93808 _dispatch_root_queue_drain + 644
8 libdispatch.dylib            0x18bc9610 _dispatch_client_callout + 16
7 libdispatch.dylib            0x18bcae628 _dispatch_call_block_and_release + 24
6 Meal Planner                  0x100012480 thunk for @escaping @callee_guaranteed () -> () + 48
5 Meal Planner                  0x10003fde4 closure #1 in MKTCustomMealPlannerCollectionViewCell.saveMeal(:) + 804
4 Meal Planner                  0x1000408dc MKTCustomMealPlannerCollectionViewCell.populateMealData() + 1124
3 Meal Planner                  0x100005860 MenuItem.__allocating_init(imageAsset:title:subtitle:) + 64
2 libswiftCore.dylib            0x19023eec4 swift_allocObject + 52
1 libswiftCore.dylib            0x19023ed0c swift_slowAlloc + 60
0 libsystem_malloc.dylib        0x19bf36da8 _malloc_zone_malloc + 152
```

Проверяем с чек-листом





Expert Code Reviewer

Проверяем с чек-листом

- ▶ Корректность выполнения задания
- ▶ Есть ли дублирование?
- ▶ Соответствует нашим стандартам?
- ▶ Есть ли изменение сторонней функциональности?



Что облегчает ручное ревью?

MSG-14375-redesign-channel-cell-optimization → develop **MERGED**

MSG-14375: Optimize redesign channel list cells rendering.

[Overview](#) [Diff](#) [Commits](#) [Builds](#)

Aleksandr Rogachev created a pull request 5 days ago

Куча мелких изменений в RedesignChannelListItemView и связанных классах для оптимизации отрисовки:

- Поправил лэйаут в VerticalButton. Предыдущий код был тяжелый и пересчитывал лэйаут много раз за проход
- Порефачил вызовы на конфигурацию и обновление кнопок на слайдах ячеек - вызывались слишком много раз на каждый чих, а с учетом того, что background там выставлялся через генерацию картинок, это было довольно тяжело
- Пооптимизировал AvatarListItemView: поправил blending картинок, заменил backgroundView на CALayer
- Пооптимизировал AvatarListView: уменьшил частоту пересозданий саввьюх при переиспользовании
- В ячейках чатов пооптимизировал загрузку картинок: убрал блендинг, добавил AF-фильтры для пре-рендеринга скругленных краёв
- Откатил замену message на Label и вернул UILabel. Label похоже работает чуть медленнее, даже с кэшированием...
- Добавил кэширование размеров channelNameLabel. может, стоит тоже откатить на UILabel в итоге...

По итогу со скроллингом стало получше, но не идеально. Надо будет ещё посмотреть в сторону дальнейших оптимизаций подгрузки картинок и отрисовки лейблов.

PS-957-newprofilesettings-common-settings-module → develop **OPEN**

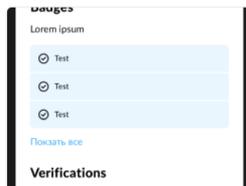
PS-957 newprofilesettings common settings module

[Overview](#) [Diff](#) [Commits](#) [Builds](#)

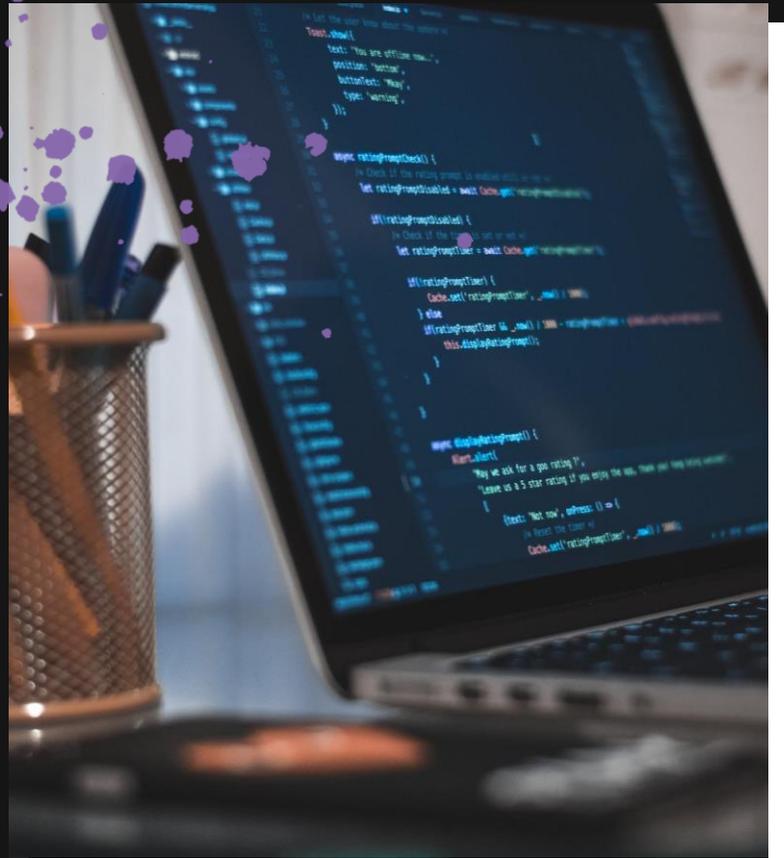
Mikhail Kuzmin created a pull request 4 days ago

- **PS-957:** Generate ProfileCommonSettings
- **PS-957:** Implement widgets celldata building
- **PS-957:** WIP | Implement ProfileBageBar adapter for form
- **PS-957:** Add badgeBarFormItem and new widget parser in ProfileCommonSettings
- **PS-957:** Implement ProfileCommonSettings module

Макеты



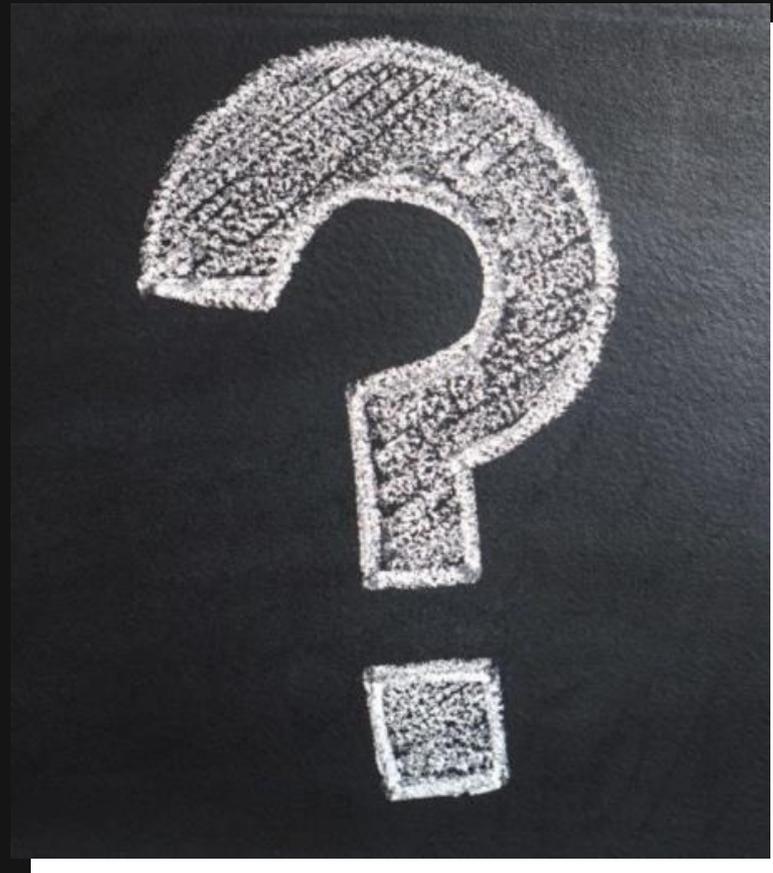
Выводы



Выводы

01. PR отнимают много времени - оптимизируем
02. Время потраченное на Unit-тесты, окупается на PR
03. Даем больше информации review'ерам

Ссылки и Вопросы



avito.tech

Москва — 2021

Екатерина Батеева

iOS Разработчик

 eabateeva@avito.ru

 [Neifmetus](#)

 [Neifmetus](#)

Ссылки:

[Исследование Smartbear](#)

[Swiftlint](#)

[Swift-format](#)