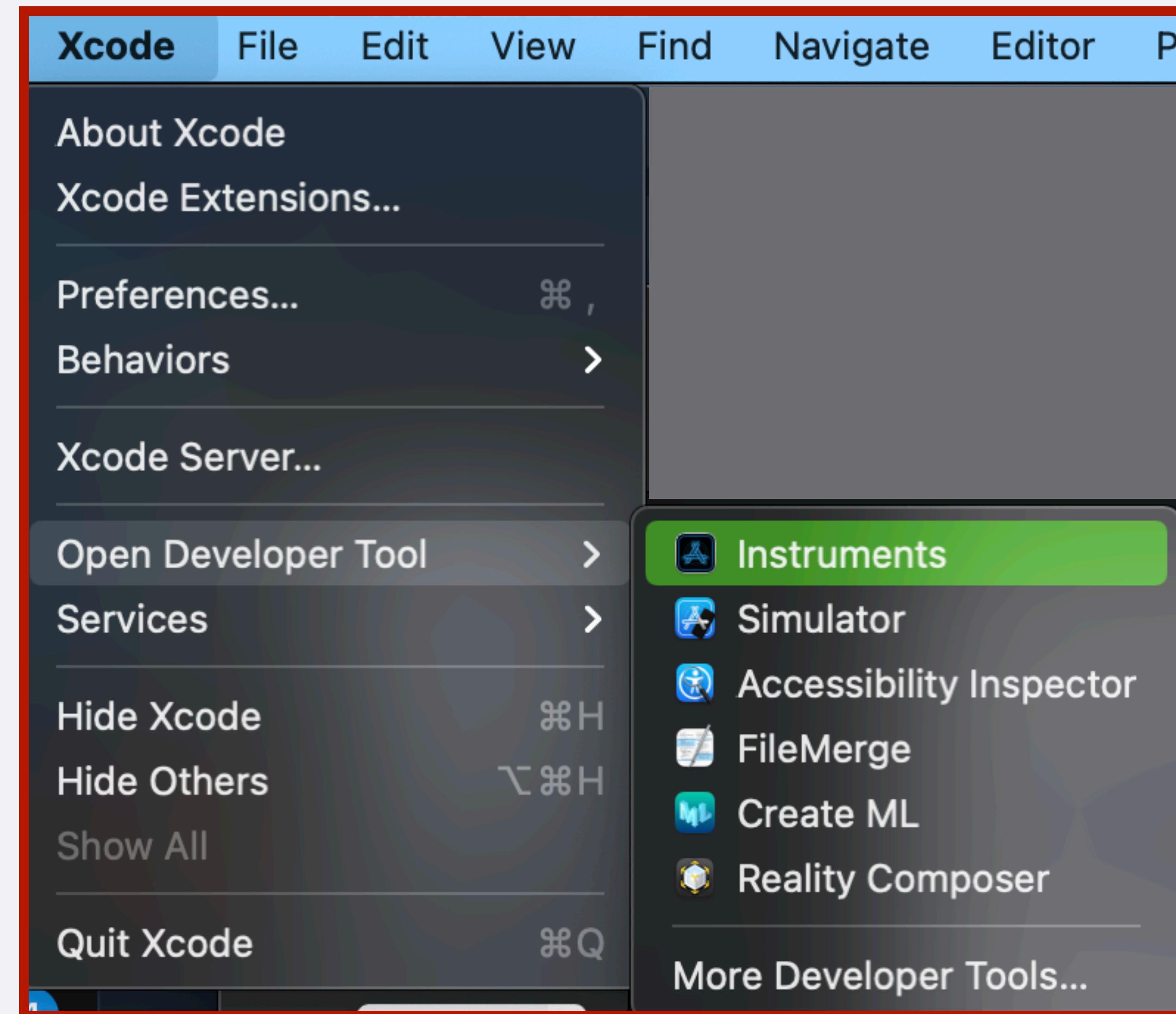


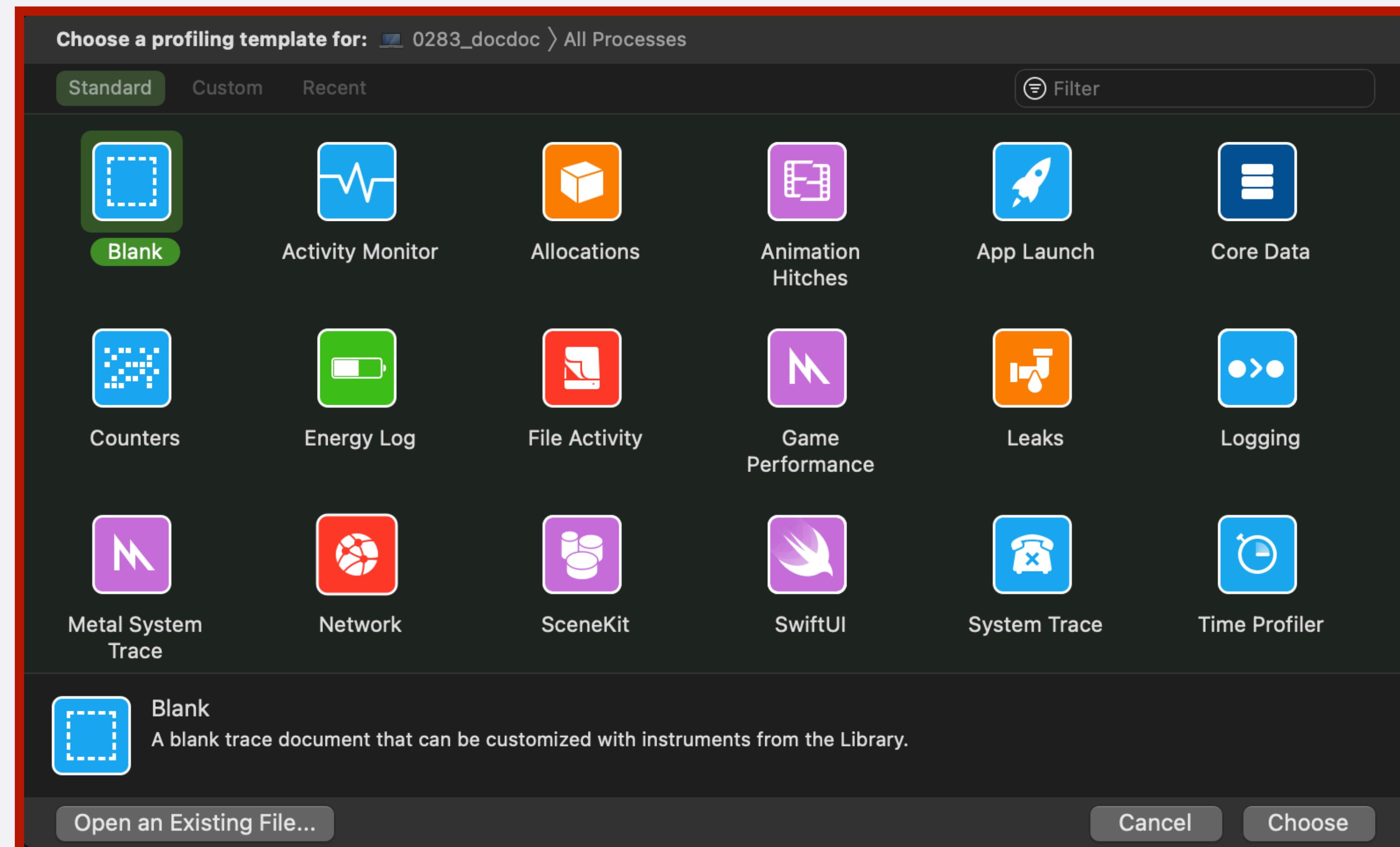
Профилирование

Профилирование – сбор характеристик работы программы, таких как время выполнения отдельных фрагментов (обычно подпрограмм), число верно предсказанных условных переходов, число кэш-промахов и т. д.

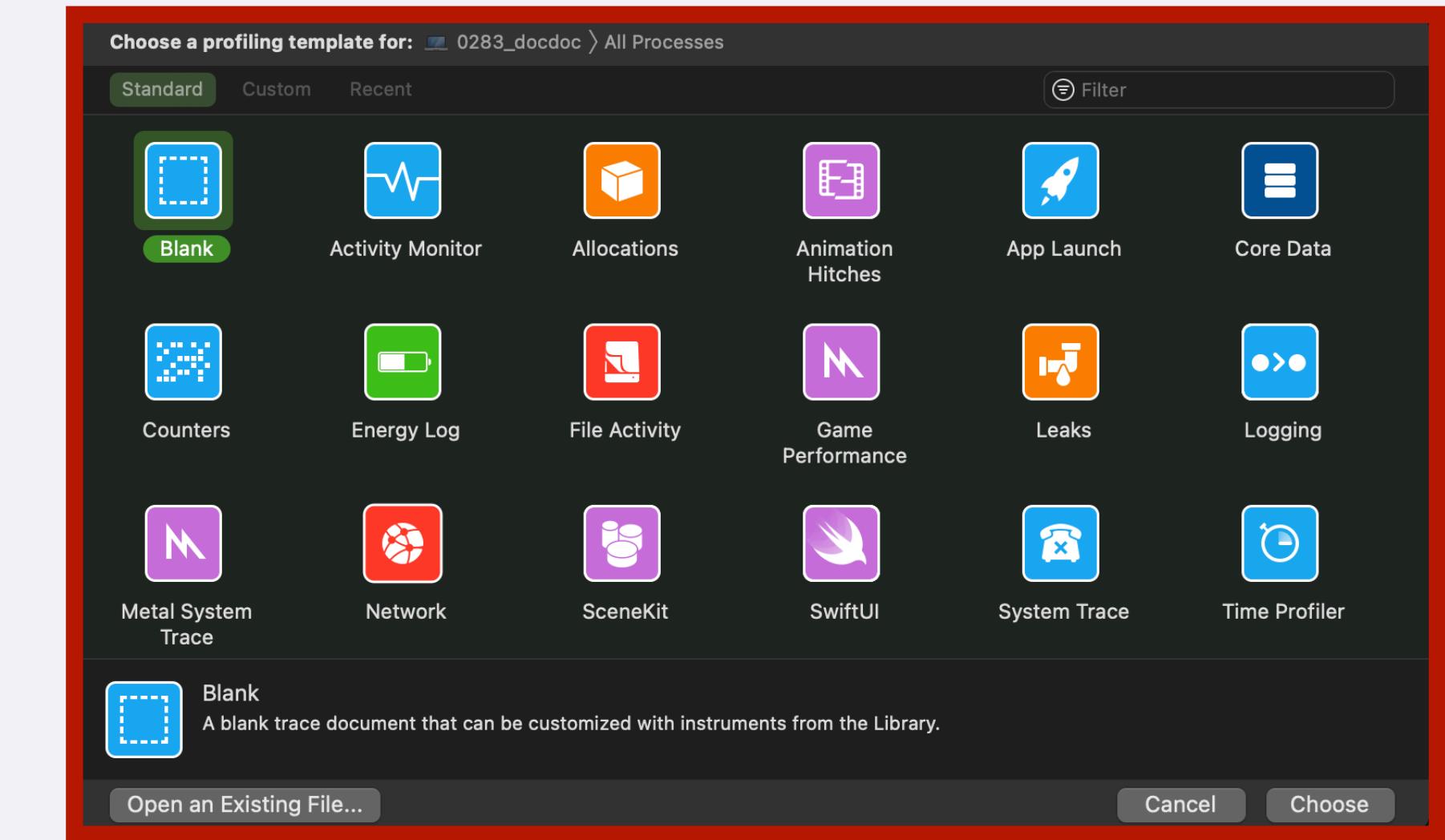
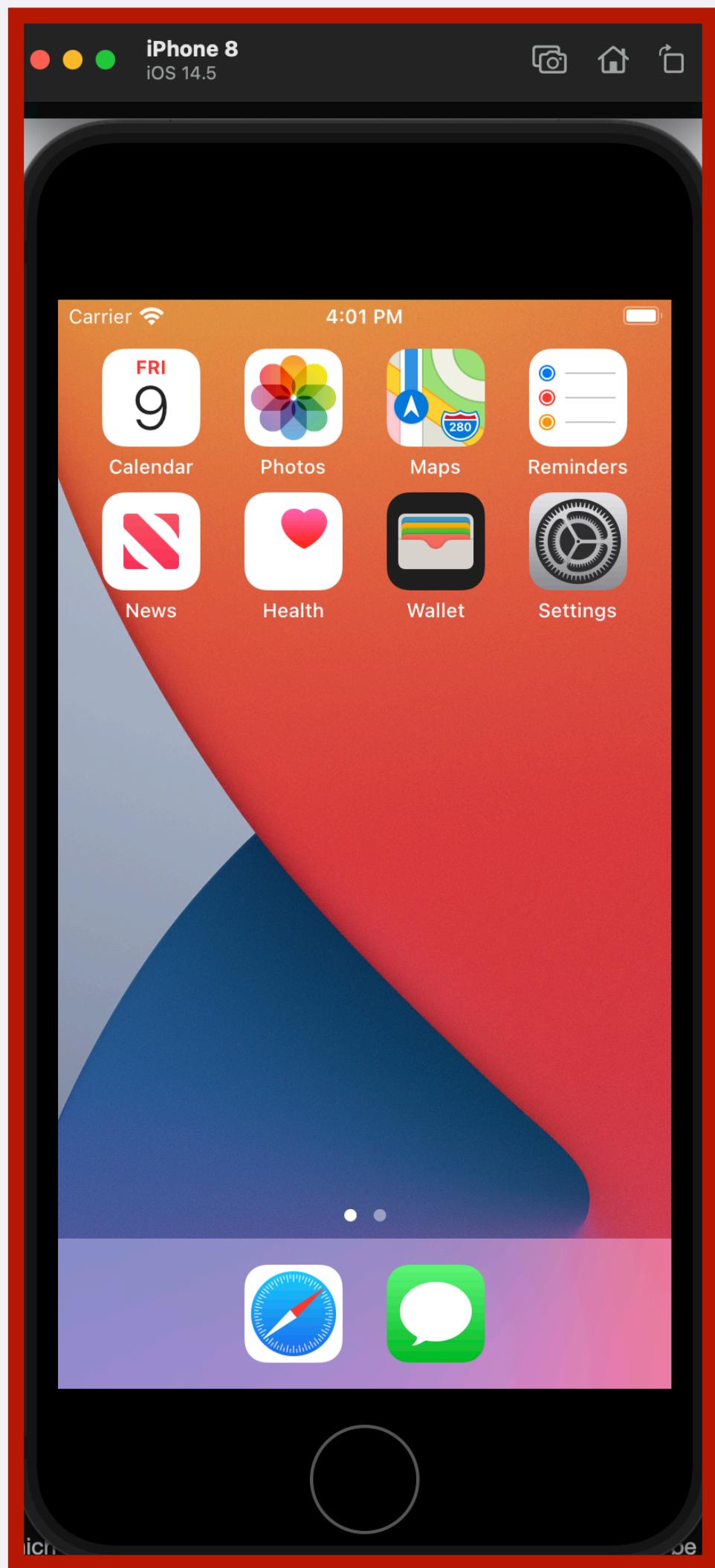
Instruments



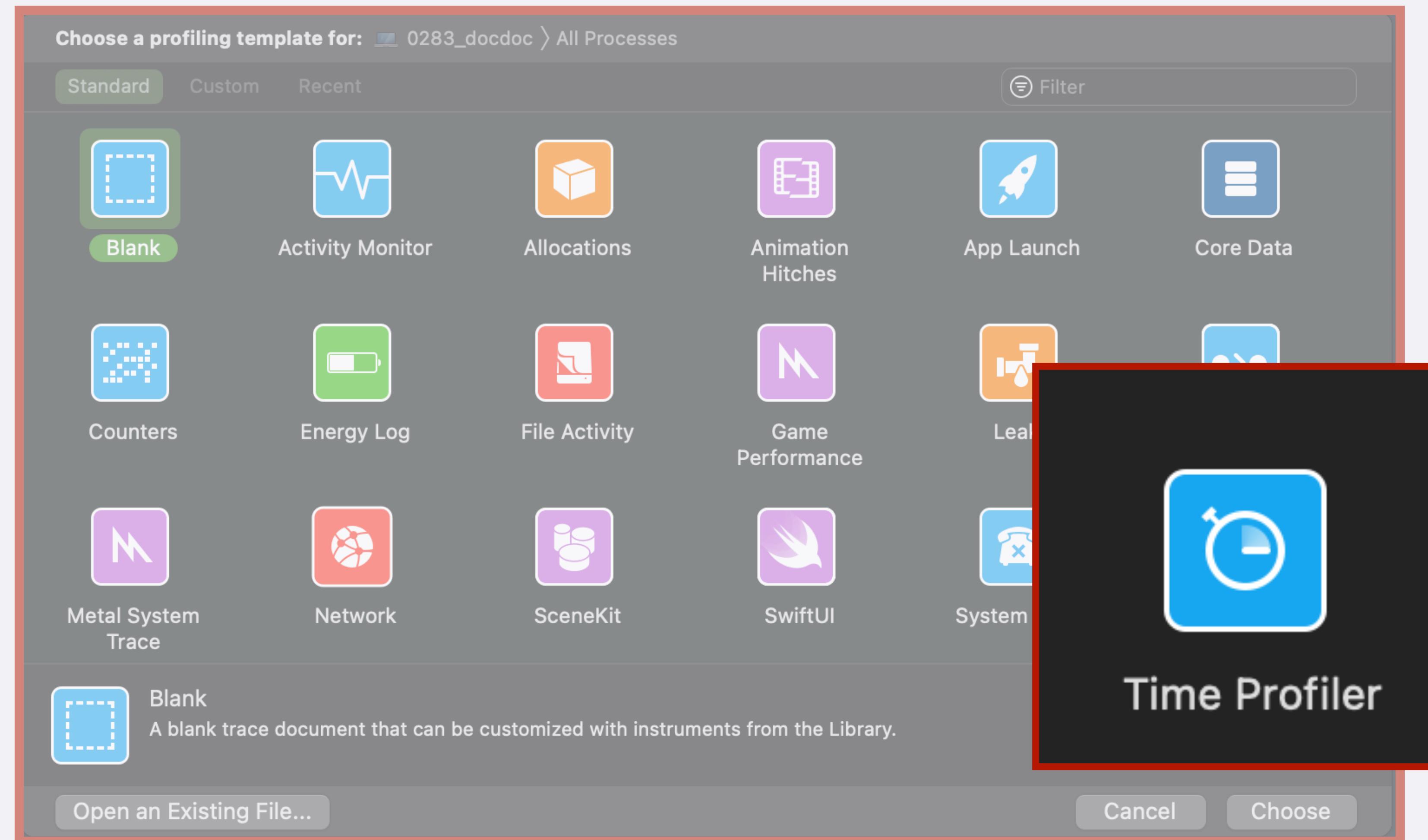
Instruments



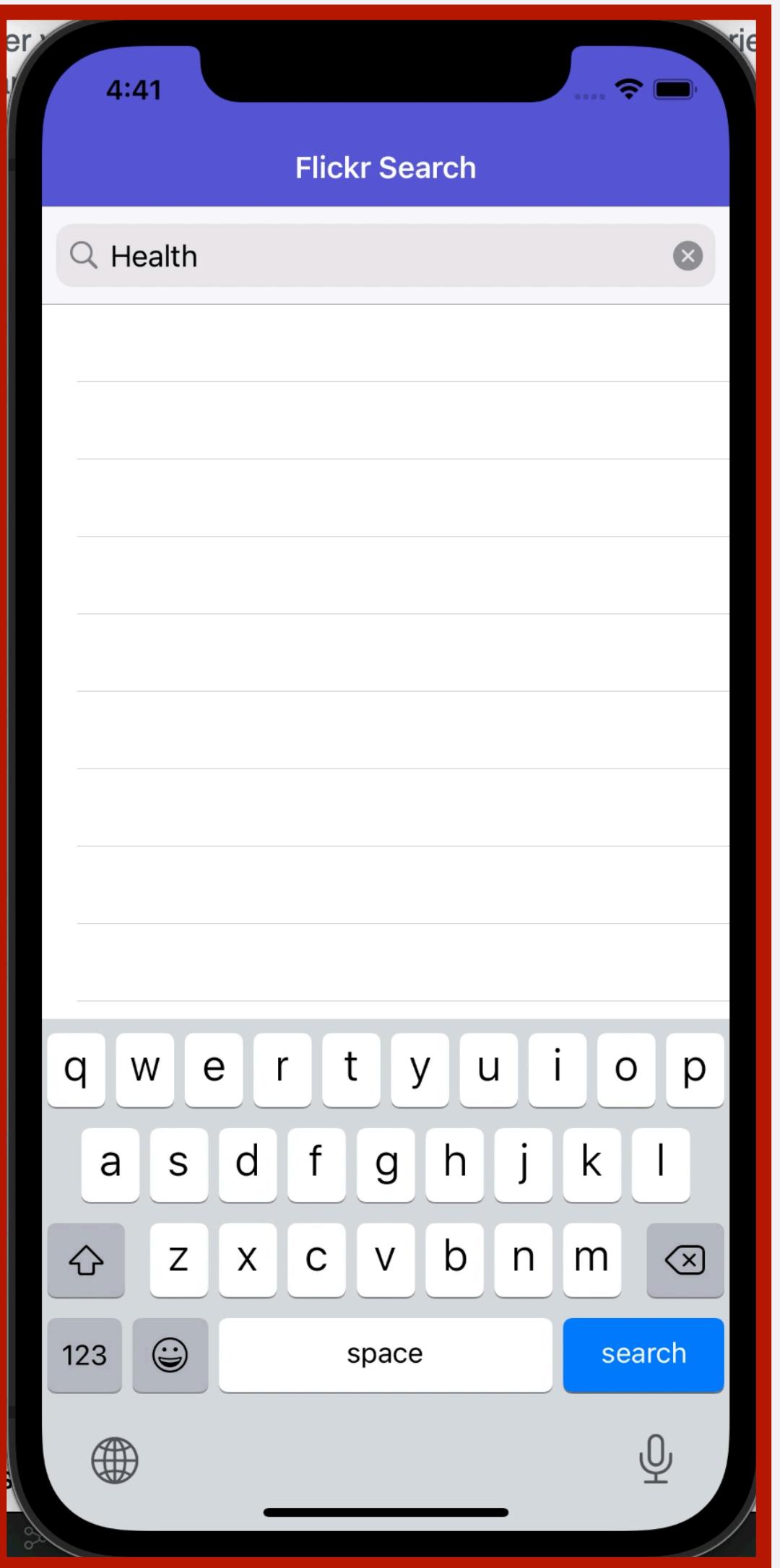
Instruments



Instruments



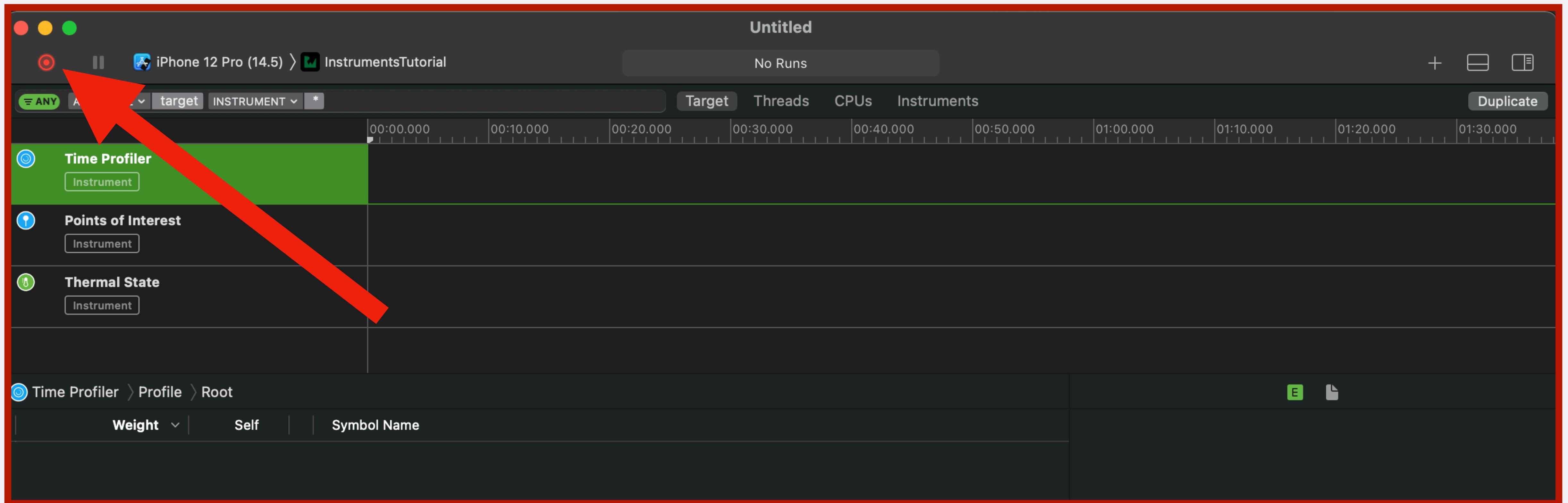
Instruments. Time profiler



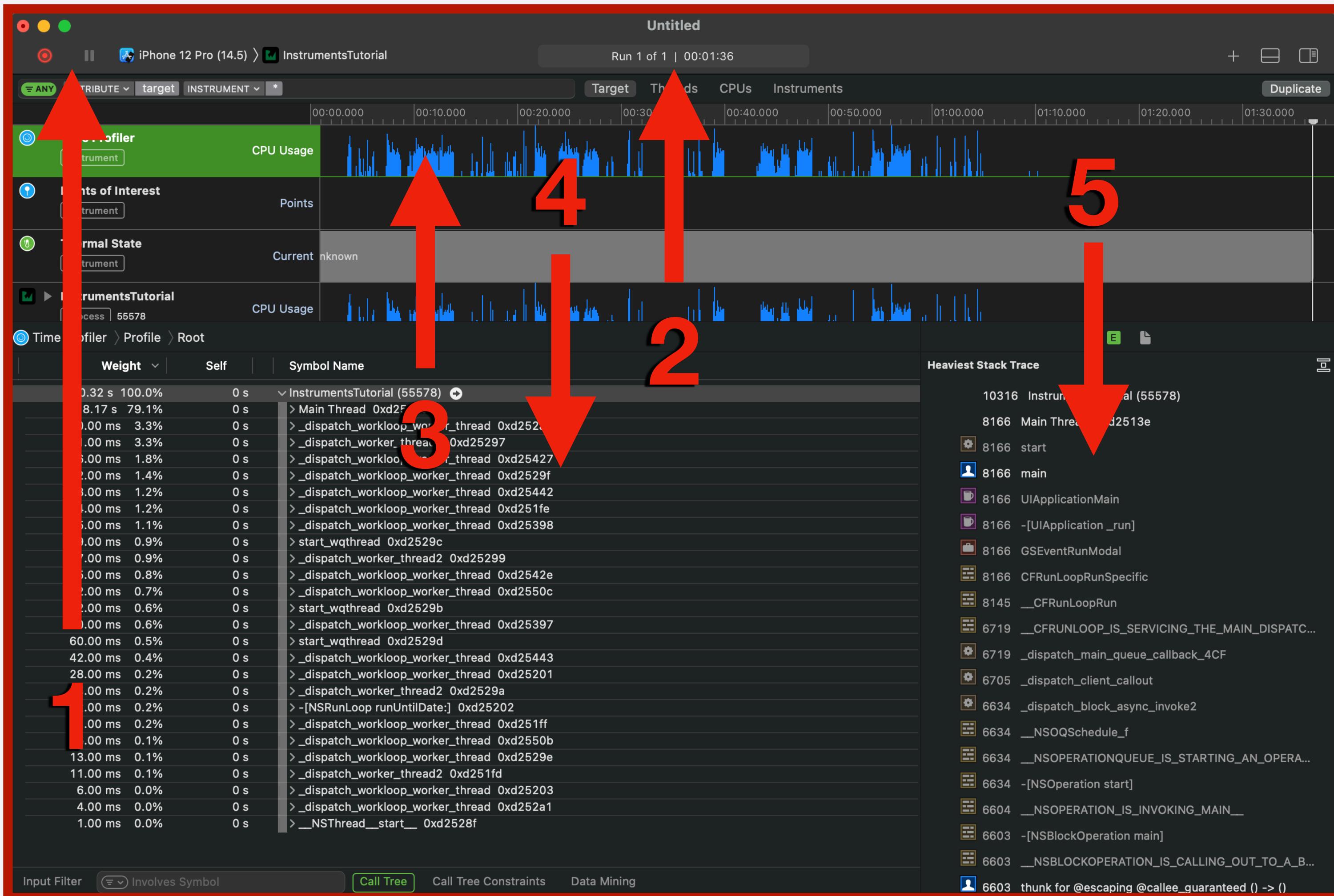
Instruments. Time profiler

Time Profiler > Profile > Root			
Weight	Self Weight	Symbol Name	
890.00 ms	100.0%	0 s	▼ InstrumentsTutorial (35104)
605.00 ms	67.9%	0 s	▼ Main Thread 0xd09fae
468.00 ms	52.5%	0 s	main InstrumentsTutorial
15.00 ms	1.6%	0 s	► <Unknown Address>
15.00 ms	1.6%	0 s	► 0x109046a31
12.00 ms	1.3%	0 s	► 0x10902c1f9
11.00 ms	1.2%	0 s	► specialized SearchResultsCollectionViewCell.init(coder:) InstrumentsTutorial
10.00 ms	1.1%	0 s	► ViewController.searchBarSearchButtonClicked(_) InstrumentsTutorial
7.00 ms	0.7%	0 s	► closure #1 in ViewController.searchBarSearchButtonClicked(_) InstrumentsTutorial
7.00 ms	0.7%	0 s	► specialized ViewController.init(coder:) InstrumentsTutorial
6.00 ms	0.6%	0 s	► ViewController.tableView(_:cellForRowAt:) InstrumentsTutorial
5.00 ms	0.5%	0 s	► static FlickrAPI.loadImage(for:withSize:_) InstrumentsTutorial
4.00 ms	0.4%	0 s	► 0x10903b3a2
3.00 ms	0.3%	0 s	► SearchResultsViewController.collectionView(_:cellForItemAt:) InstrumentsTutorial
3.00 ms	0.3%	0 s	► static FlickrAPI.searchFlickrForTerm(_:completion:) InstrumentsTutorial
3.00 ms	0.3%	0 s	► closure #1 in SearchResultsViewController.collectionView(_:cellForItemAt:) InstrumentsTutorial
2.00 ms	0.2%	2.00 ms	► 0x10904ef86
2.00 ms	0.2%	0 s	► FlickrPhoto.isFavourite.getter InstrumentsTutorial
2.00 ms	0.2%	2.00 ms	► 0x10904ee9c
2.00 ms	0.2%	0 s	► 0x10903c94a
1.00 ms	0.1%	0 s	► 0x10903c6dc
1.00 ms	0.1%	1.00 ms	► 0x109040740
1.00 ms	0.1%	1.00 ms	► 0x109045d1b
1.00 ms	0.1%	1.00 ms	► 0x10904efcd
1.00 ms	0.1%	1.00 ms	► 0x10903434a
1.00 ms	0.1%	1.00 ms	► 0x10902f154
1.00 ms	0.1%	1.00 ms	► 0x10902bcef
1.00 ms	0.1%	1.00 ms	► type metadata accessor for AppDelegate InstrumentsTutorial
1.00 ms	0.1%	1.00 ms	► 0x10904115d
1.00 ms	0.1%	1.00 ms	► 0x1090400000

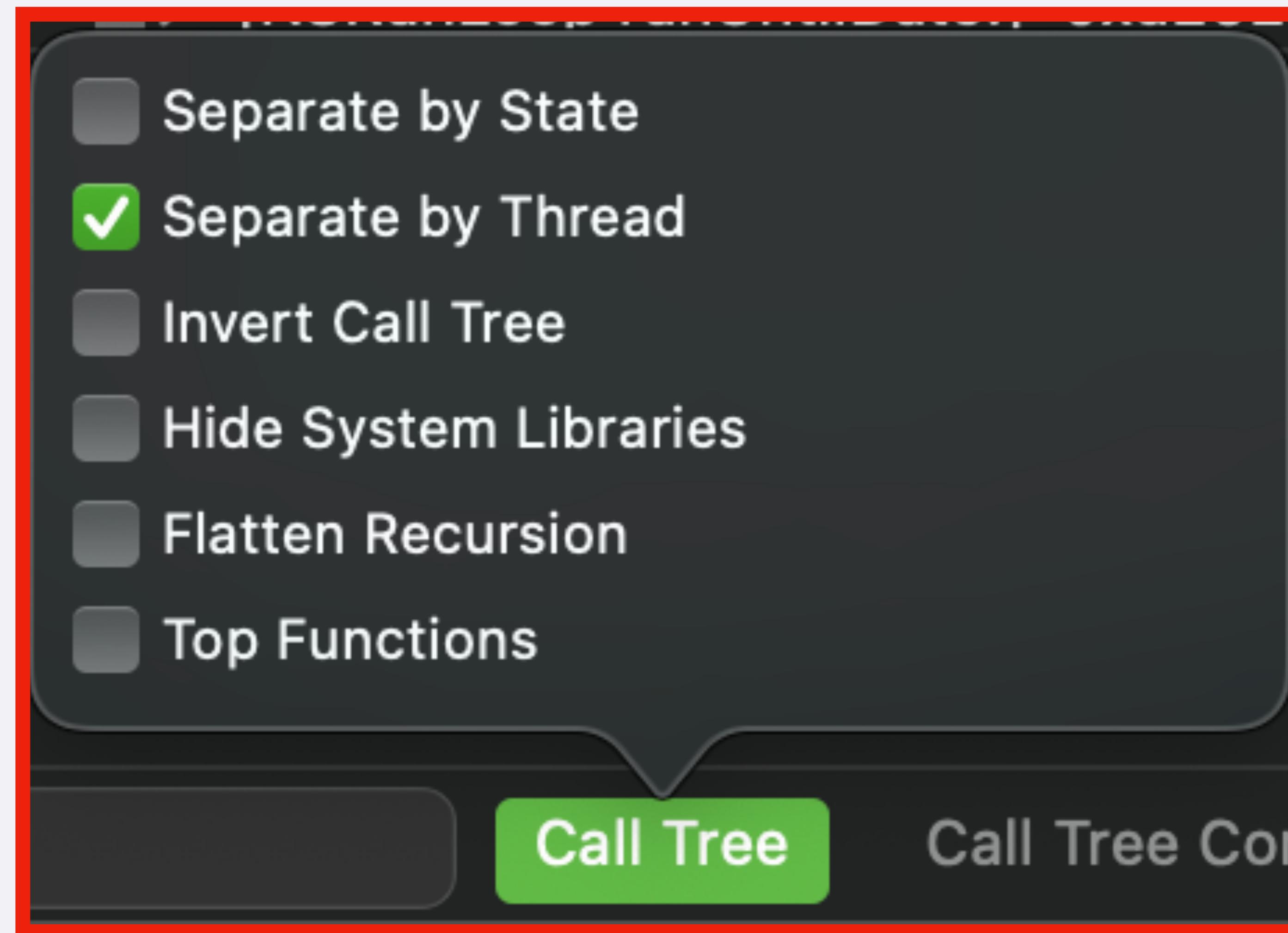
Instruments. Time profiler



Instruments . Time profiler



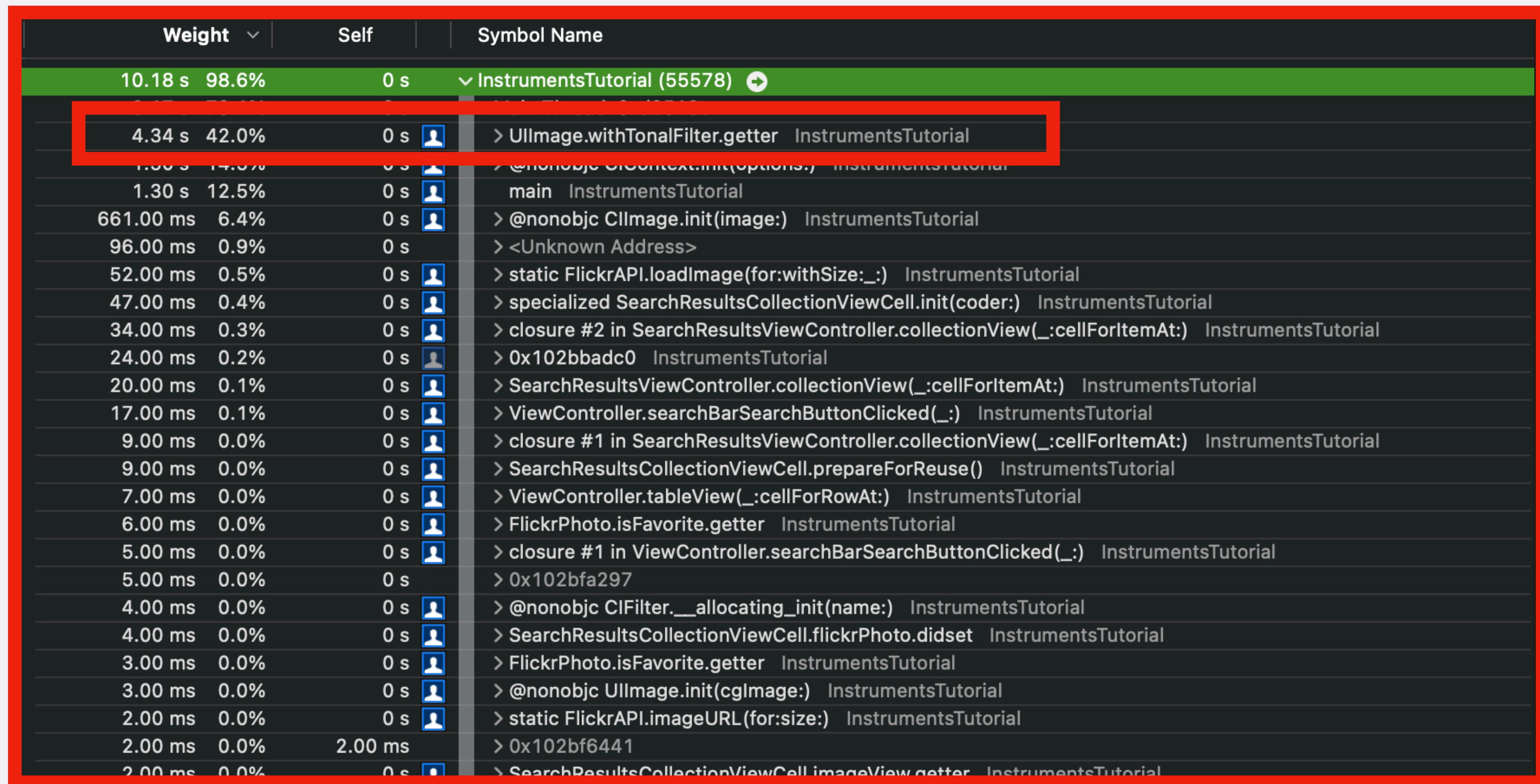
Instruments . Time profiler



Instruments . Time profiler

Weight	Self	Symbol Name
10.18 s 98.6%	0 s	▀ InstrumentsTutorial (55578) ➔
8.17 s 79.1%	0 s	> Main Thread 0xd2513e
24.00 ms 0.2%	0 s	> _dispatch_worker_thread2 0xd25297
23.00 ms 0.2%	0 s	> _dispatch_workloop_worker_thread 0xd252a0
13.00 ms 0.1%	0 s	> _dispatch_workloop_worker_thread 0xd25427
10.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd2529f
10.00 ms 0.0%	0 s	> start_wqthread 0xd2529b
8.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd251fe
8.00 ms 0.0%	0 s	> _dispatch_worker_thread2 0xd25299
6.00 ms 0.0%	0 s	> start_wqthread 0xd2529d
6.00 ms 0.0%	0 s	> start_wqthread 0xd2529c
6.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd251ff
5.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd25442
4.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd25398
3.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd2550b
3.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd25443
2.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd25397
2.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd25201
2.00 ms 0.0%	0 s	> _dispatch_workloop_worker_thread 0xd2542e
1.00 ms 0.0%	0 s	> _dispatch_worker_thread2 0xd2529a

Instruments. Time profiler



Instruments. Time profiler

```
extension UIImage {
    var withTonalFilter: UIImage? {
        let context = CIContext(options: nil)
        let filter = CIFilter(name: "CIPhotoEffectTonal")
        let input = CoreImage.CIImage(image: self)
        filter?.setValue(input, forKey: kCIInputImageKey)

        guard
            let outputImage = filter?.outputImage,
            let outImage = context.createCGImage(outputImage, from: outputImage.extent)
        else { return nil }

        let returnImage = UIImage(cgImage: outImage)
        return returnImage
    }
}
```

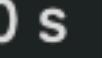
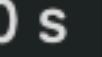
Instruments. Time profiler

```
extension UIImage {
    var withTonalFilter: UIImage? {
        let context = CIContext(options: nil)
        let filter = CIFilter(name: "CIPhotoEffectTonal")
        let input = CoreImage.CIImage(image: self)
        filter?.setValue(input, forKey: kCIInputImageKey) ! 0.02%
    }

    guard
        let outputImage = filter?.outputImage, ! 0.42%
        let outImage = context.createCGImage(outputImage, from: outputImage.extent)
    else { return nil } ! 97.99% ×

        let returnImage = UIImage(cgImage: outImage)
        return returnImage
    }
}
```

Instruments. Time profiler

Weight	Self	Symbol Name
8.30 s 100.0%	0 s	▀ InstrumentsTutorial (55578)
8.17 s 98.3%	0 s	▀ Main Thread 0xd2513e
4.34 s 52.2%	0 s	▀  <code>UllImage.withTonalFilter.getter</code> InstrumentsTutorial
4.34 s 52.2%	0 s	▀  <code>> closure #2 in SearchResultsController.collectionView(_:cellForItemAt:)</code> InstrumentsTutorial 
1.50 s 18.0%	0 s	▀  <code>> @nonobjc CIContext.init(options:)</code> InstrumentsTutorial
1.30 s 15.6%	0 s	▀  <code>main</code> InstrumentsTutorial
661.00 ms 7.9%	0 s	▀  <code>> @nonobjc CILmage.init(image:)</code> InstrumentsTutorial
96.00 ms 1.1%	0 s	▀ <code>> <Unknown Address></code>

Instruments. Time profiler

```
resultsCell.heartToggleHandler = { _ in
    self.collectionView.reloadItems(at: [indexPath])
}

ImageCache.shared.loadThumbnail(for: flickrPhoto) { result in
    switch result {
        case .success(let image):
            if resultsCell.flickrPhoto == flickrPhoto {
                if flickrPhoto.isFavorite {
                    resultsCell.imageView.image = image
                } else if let filteredImage = image.withTonalFilter {
                    resultsCell.imageView.image = filteredImage
                }
            }
        case .failure(let error):
            print("Error: \(error)")
    }
}
return resultsCell
```

! 100.00%

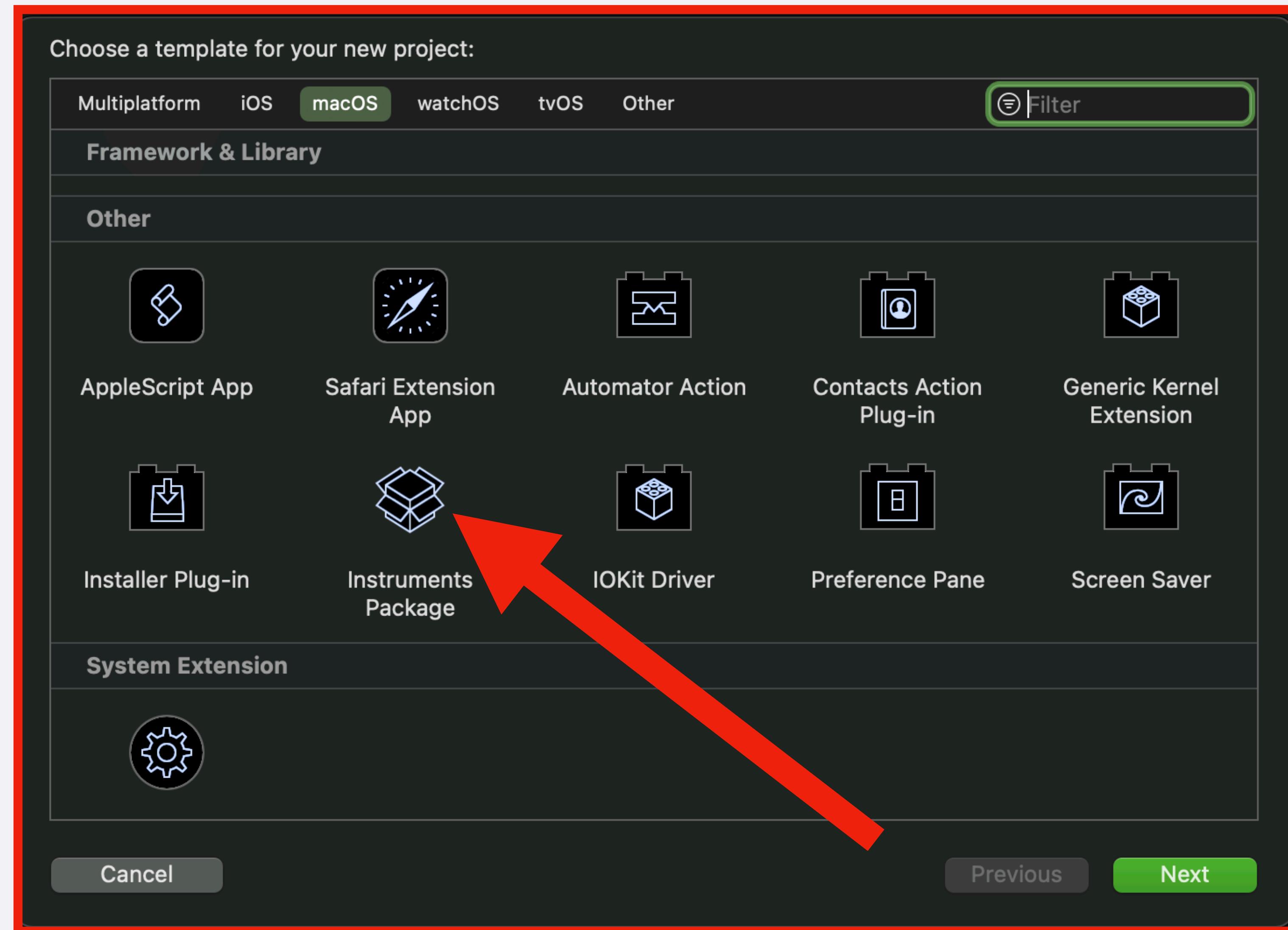
Demo

Instruments. Задание

- Allocations
- Zombies
- Network
- Energy use

300,000

Instruments . Custom



Instruments . Custom . Структура

Что	Атрибуты	Описание
Схемы данных	interval-schema, point-schema и т.д.	Описывает структуру данных в виде таблицы подобно sql-схемам. Схемы используются в других элементах разметки, чтобы определить тип данных на входе и выходе модели, например, при описании отображения (UI).
Импорт схем данных	import-schema	Импорт готовых схем. Он позволяет использовать структуры данных, которые определены Apple.
Модель инструмента	modeler	Связывает инструмент с файлом .clp, в котором определена логика инструмента, и объявляет ожидаемую схему данных на входе и выходе модели.
Описание инструмента	instrument	Описывает модель данных и определяет, как события будут отображаться в UI. Модель данных описывается с помощью атрибутов create-table, create-parameter и тд. Графики инструмента определяются атрибутами graph, а таблица деталей — list, narrative и т.д.

```
11 <point-schema>
12 <id>detected-mocks-narrative</id>
13 <title>Результаты</title>
14 <timestamp-column>
15   <mnemonic>time</mnemonic>
16   <title>Время</title>
17   <type>event-time</type>
18 </timestamp-column>
19 <column>
20   <mnemonic>description</mnemonic>
21   <title>Описание результата</title>
22   <type>narrative</type>
23 </column>
24 </point-schema>
```

Instruments . Custom . CLIPS

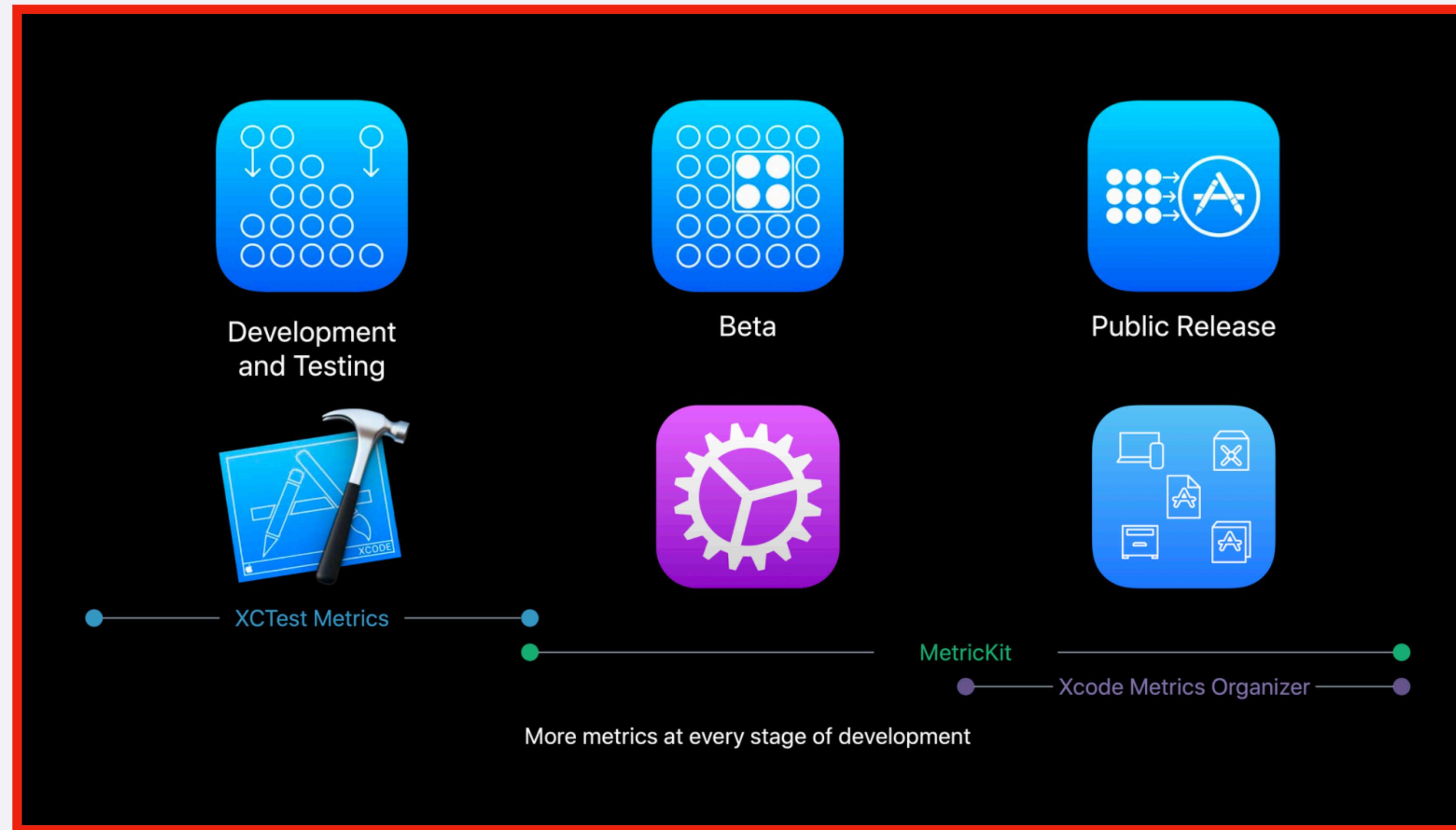


- «Fact» — событие
- «Rule» — if-блок

```
21 (defrule MODELER::record-test-started
22   (os-signpost (subsystem "com.sberbank.sbol") (category "OSTUnitTestLog") (name "Test") (event-type "Begin")
23     (message$ "Name:" ?testName) (time ?time) (identifier ?identifier)
24   )
25   =>
26   (assert (unitTest (time ?time) (signpost-id ?identifier) (testName ?testName)))
27 )
28
29 (defrule MODELER::record-mock-started
30   (os-signpost (subsystem "ccom.sberbank.sbol") (category "OCMockLog") (name "Mock") (event-type "Begin")
31     (message$ "Class:" ?class) (time ?time) (identifier ?identifier)
32   )
33   (unitTest (time ?t1) (testName ?testName))
34   =>
35   (assert (mock (time ?time) (signpost-id ?identifier) (mockedClass ?class) (createdTestName ?testName)))
36 )
```

MetricKit

MetricKit



Почитать

1. <https://developer.apple.com/videos/play/wwdc2019/411/>
2. <https://www.raywenderlich.com/16126261-instruments-tutorial-with-swift-getting-started>
3. <https://www.avanderlee.com/debugging/xcode-instruments-time-profiler/>
4. <https://www.raywenderlich.com/20952676-monitoring-for-ios-with-metrickit-getting-started>
5. <https://developer.apple.com/videos/play/wwdc2020/10081/>
6. <https://habr.com/ru/company/sberbank/blog/447594/>