

# Автоматизируем инфраструктуру с ПОМОЩЬЮ БОТОВ

Пендрак Олег



Спикер 

# Пендрак Олег

Руководитель по автоматизации тестирования  
VK Реклама

---

Пишу автотесты для iOS и Android

---

Занимаюсь инфраструктурой проекта

---

Кидаю мемы в командные чаты

---



САЙТ: [THREADQA.RU](https://threadqa.ru)

Эксперт 

# Рубинов Никита

Старший разработчик в команде генерации контента  
VK Реклама

---

Автор идеи и разработчик проекта автоматизации

---

Занимаюсь генеративными ИИ в рекламе

---

Преподаю на курсе "Обеспечение качества в  
разработке ПО" Образовательного центра VK в МГТУ

---



TELEGRAM: @NIKITARUB

Цель тестирования и  
разработки

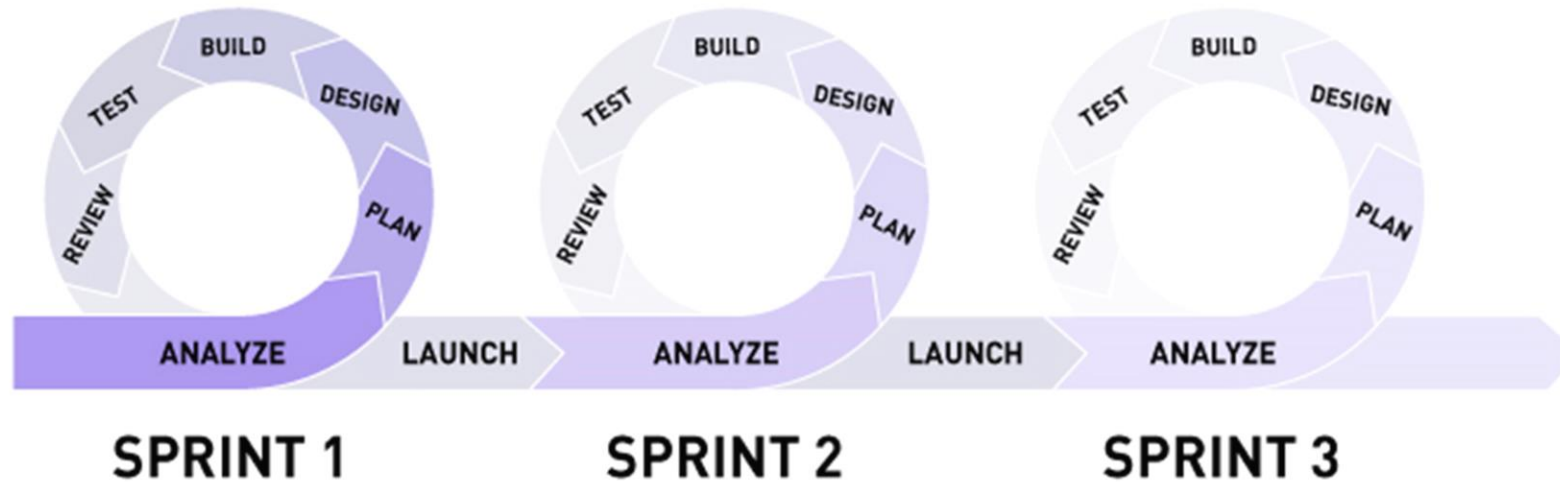
Как можно быстрее  
выпустить качественный  
продукт



# Итеративно повторяющийся процесс разработки

## ПРИМЕР СХЕМЫ РАБОТЫ ПО AGILE

**AGILE**

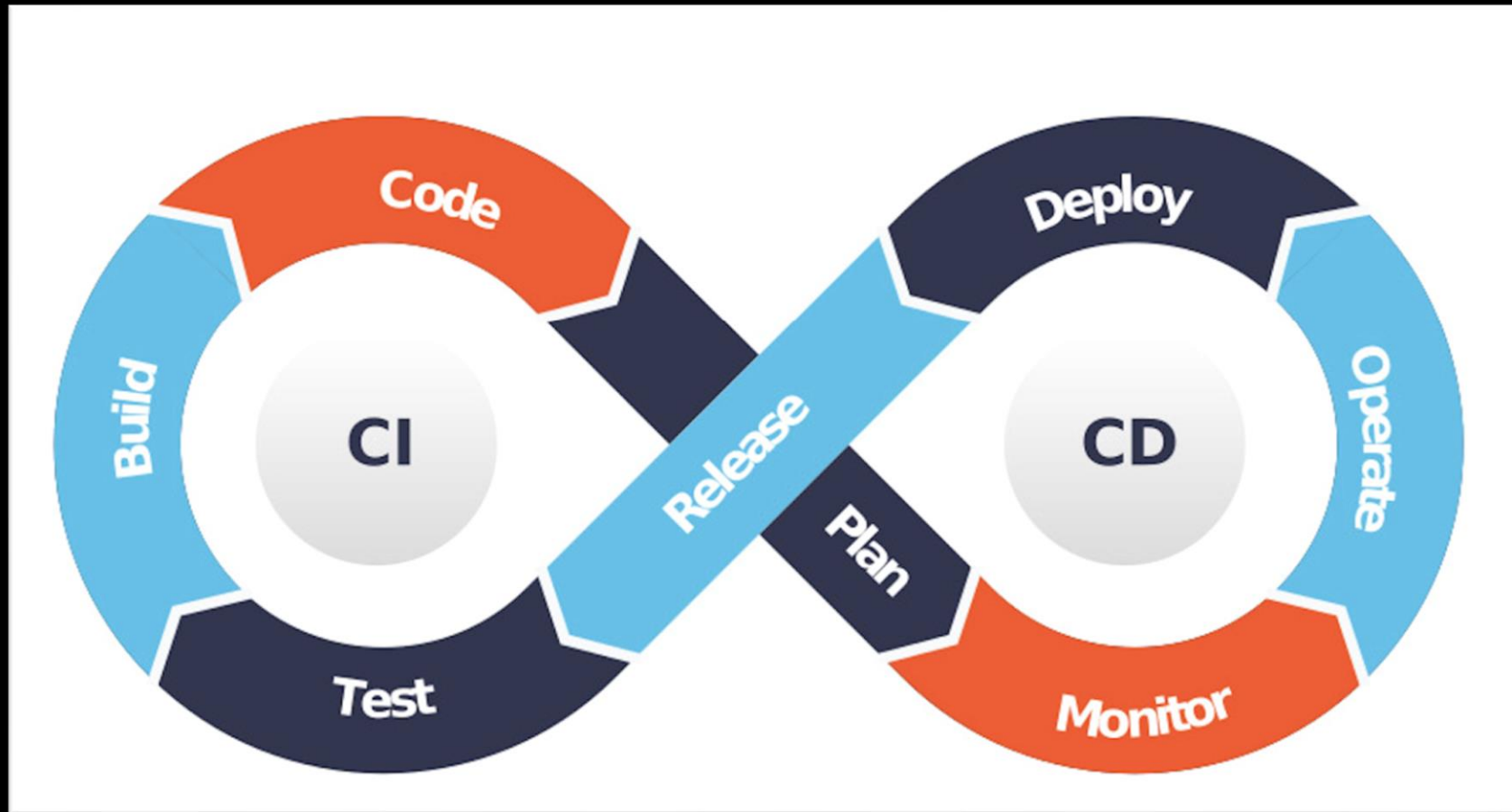


# Доска в Jira

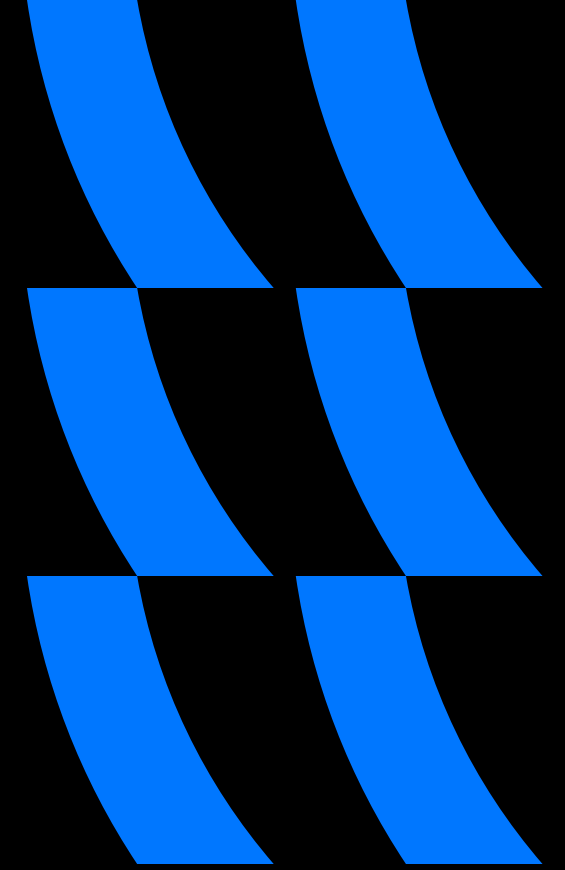
The screenshot shows the Jira Board interface for the 'Teams in Space' project. The board is organized into four columns: TO DO (5 items), IN PROGRESS (5 items), CODE REVIEW (2 items), and DONE (8 items). Each item card displays a title, a team label, a progress indicator, a priority, and a due date.

Column	Item Title	Team	Progress	Priority	Due Date
TO DO (5)	Engage Jupiter Express for outer solar system travel	SPACE TRAVEL PARTNERS	50%	High	TIS-25
	Create 90 day plans for all departments in the Mars Office	LOCAL MARS OFFICE	0%	Medium	TIS-12
	Engage Saturn's Rings Resort as a preferred provider	SPACE TRAVEL PARTNERS	0%	High	TIS-17
IN PROGRESS (5)	Requesting available flights is now taking > 5 seconds	SEESPACEEZ PLUS	25%	High	TIS-8
	Engage Saturn Shuttle Lines for group tours	SPACE TRAVEL PARTNERS	50%	High	TIS-15
	Establish a catering vendor to provide meal service	LOCAL MARS OFFICE	25%	High	TIS-15
CODE REVIEW (2)	Register with the Mars Ministry of Revenue	LOCAL MARS OFFICE	100%	High	TIS-11
	Draft network plan for Mars Office	LOCAL MARS OFFICE	100%	High	TIS-15
DONE (8)	Homepage footer uses an inline style - should use a class	LARGE TEAM SUPPORT	100%	High	TIS-68
	Engage JetShuttle SpaceWays for travel	SPACE TRAVEL PARTNERS	100%	High	TIS-23
	Engage Saturn Shuttle Lines for group tours	SPACE TRAVEL PARTNERS	100%	High	TIS-15

# Инфраструктура



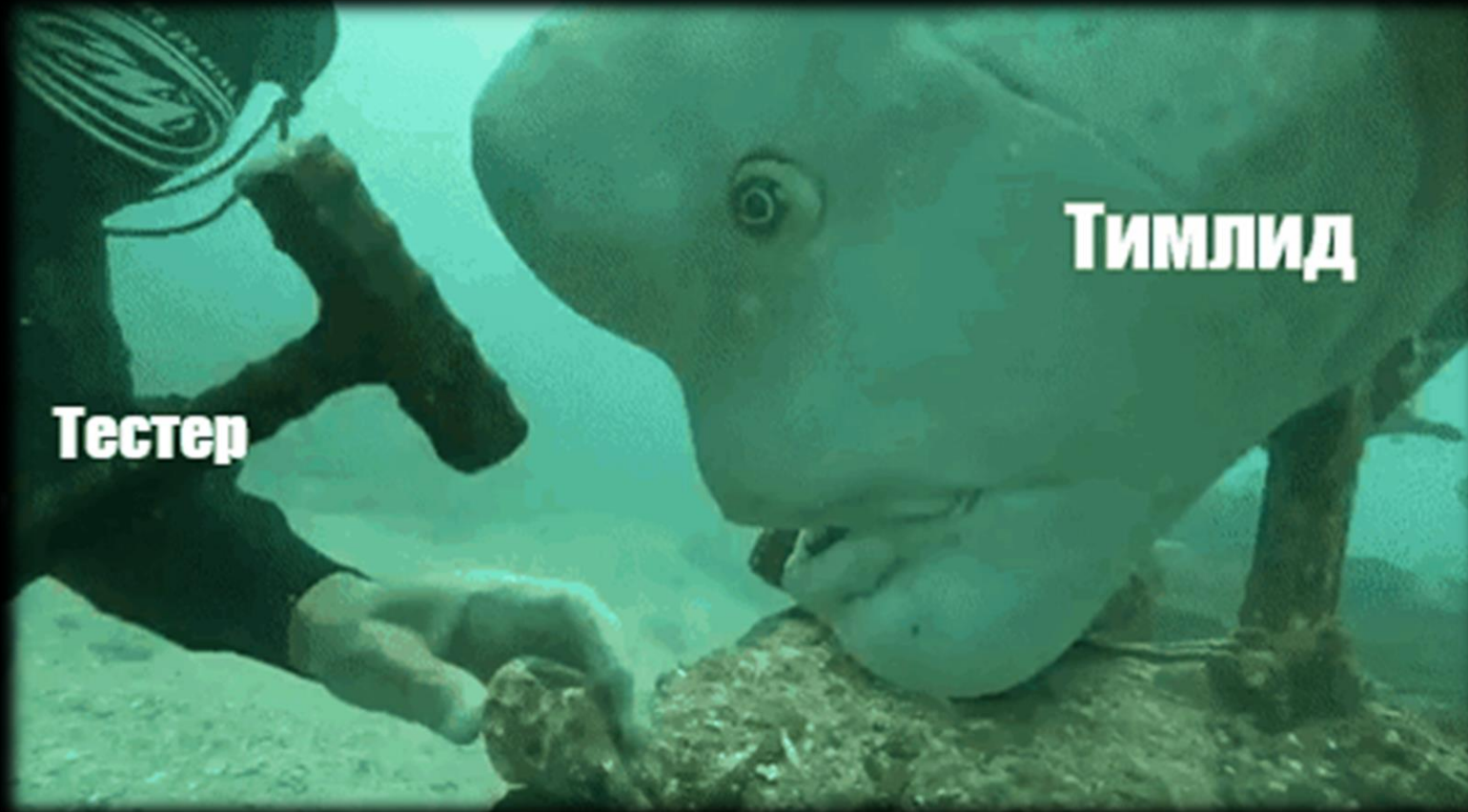
На реальных проектах  
бывает не все так  
идеально



# Узнаем статусы по задаче у коллег

# Забыли взять задачу в работу



# Путаница в параметрах на Jenkins

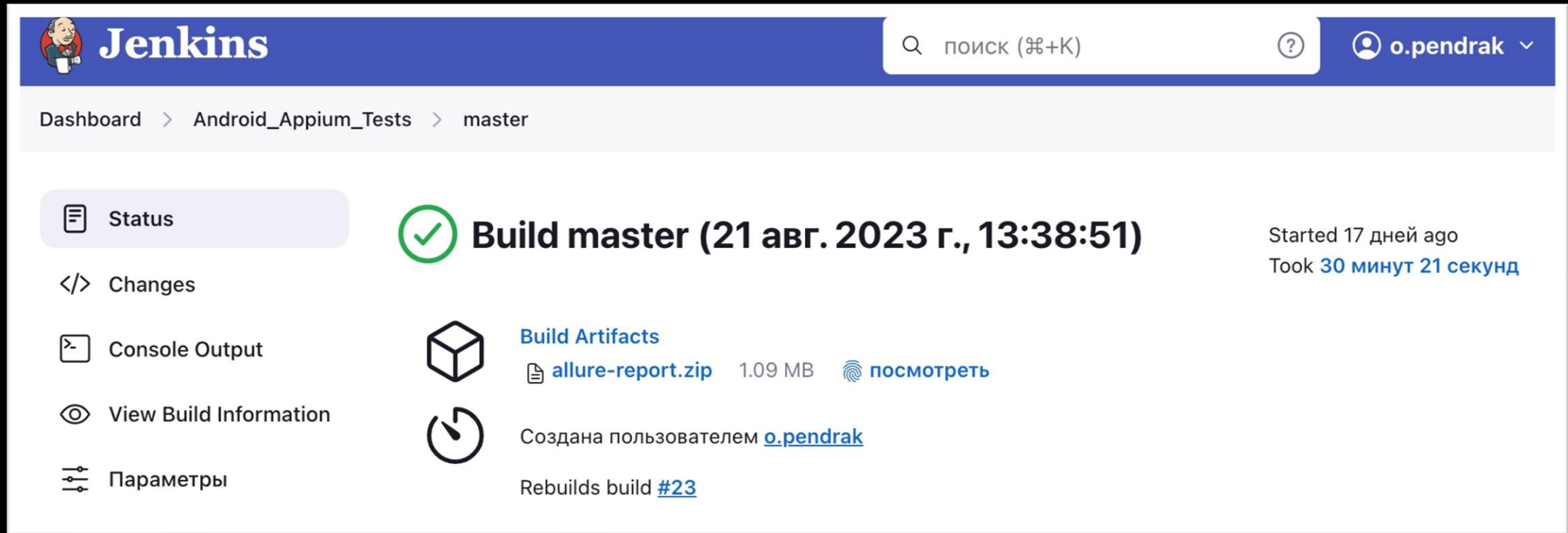
The screenshot shows the Jenkins interface for configuring a pipeline. The top navigation bar includes the Jenkins logo, a search bar with the text 'поиск (⌘+K)', and a user profile 'o.pendrak'. The breadcrumb trail is 'Dashboard > Android\_Appium\_Tests >'. The left sidebar contains several menu items: 'Status', 'Changes', 'Собрать с параметрами', 'Full Stage View', 'Open Blue Ocean', 'Allure Report', 'Schedule Build', and 'История сборок' (Build History) with a 'Тренд' (Trend) dropdown.

The main content area is titled 'Pipeline Android\_Appium\_Tests'. Below the title, it states: 'Для этой сборки необходимы следующие параметры:' (The following parameters are required for this build:). The parameters are listed as follows:

- PIPELINE\_BRANCH**: Input field containing 'master'.
- TASK\_BRANCH**: Input field containing 'origin/master'. This field is highlighted with a red border and has the text 'Обязательное поле!!!' (Mandatory field!!!) next to it.
- DEV\_BRANCH**: Input field containing 'master'. This field is highlighted with a blue border and has the text 'Тоже обязательное поле' (Also mandatory field) next to it.
- DOCKER\_IMAGE**: Input field containing 'maven/selenium:4.10.0'.



# Запустили тесты не на той ветке



The screenshot shows the Jenkins web interface. At the top, there is a blue header with the Jenkins logo and name on the left, a search bar with the text "поиск (⌘+K)" in the center, and a user profile icon for "o.pendrak" on the right. Below the header, a breadcrumb trail reads "Dashboard > Android\_Appium\_Tests > master".

On the left side, there is a vertical menu with the following items:

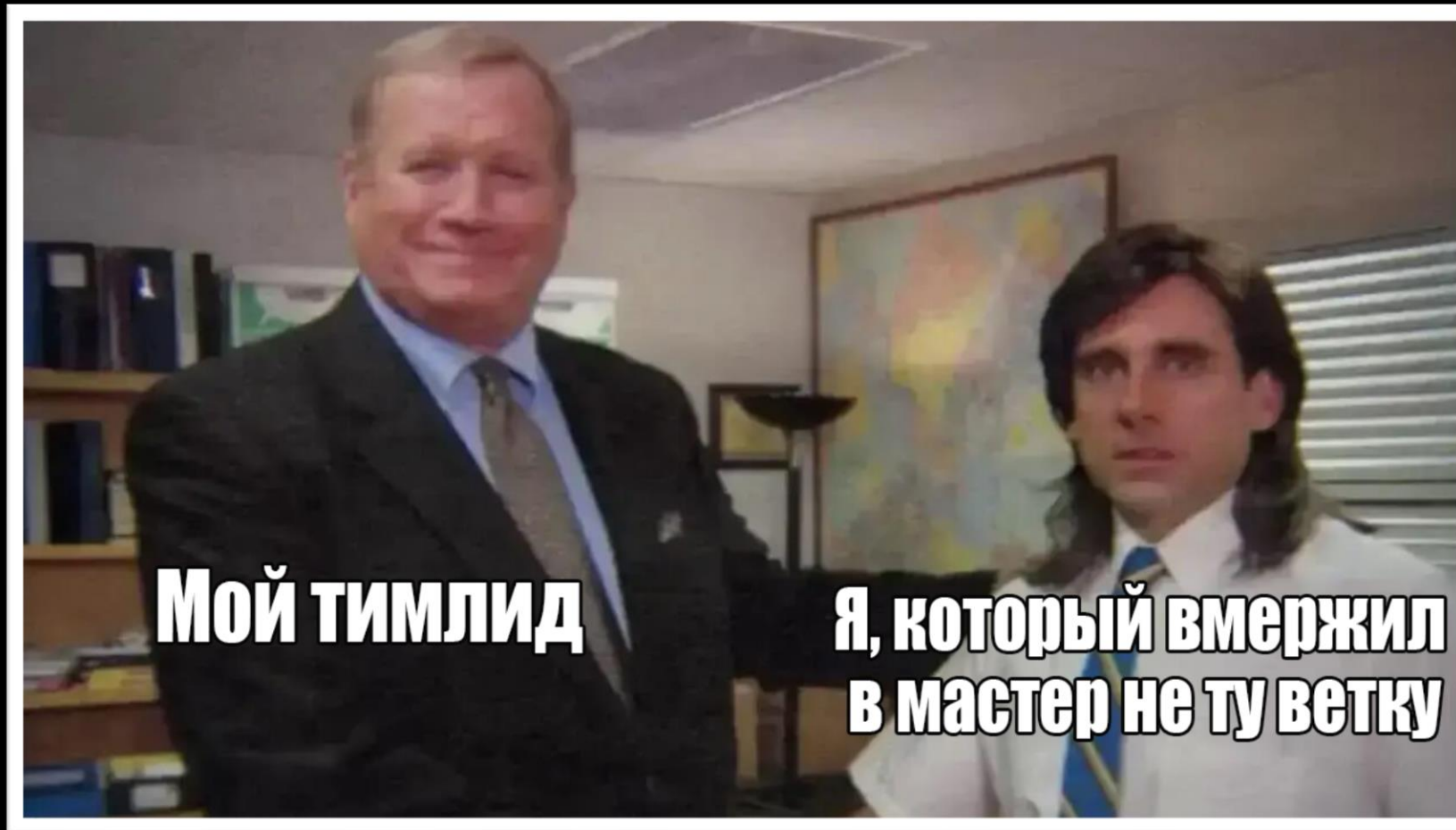
- Status (highlighted)
- Changes
- Console Output
- View Build Information
- Параметры

The main content area displays the details of a build:

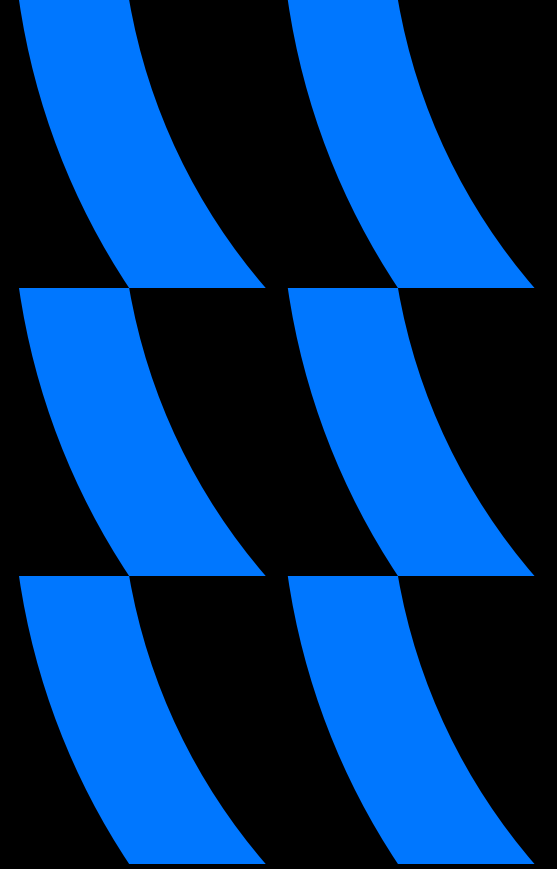
- A green checkmark icon followed by the text "Build master (21 авг. 2023 г., 13:38:51)".
- Metadata: "Started 17 дней ago" and "Took 30 минут 21 секунд".
- A "Build Artifacts" section with a cube icon, listing "allure-report.zip" (1.09 MB) with a "посмотреть" link.
- A clock icon indicating the build was "Создана пользователем o.pendrak".
- A note that the build "Rebuilds build #23".



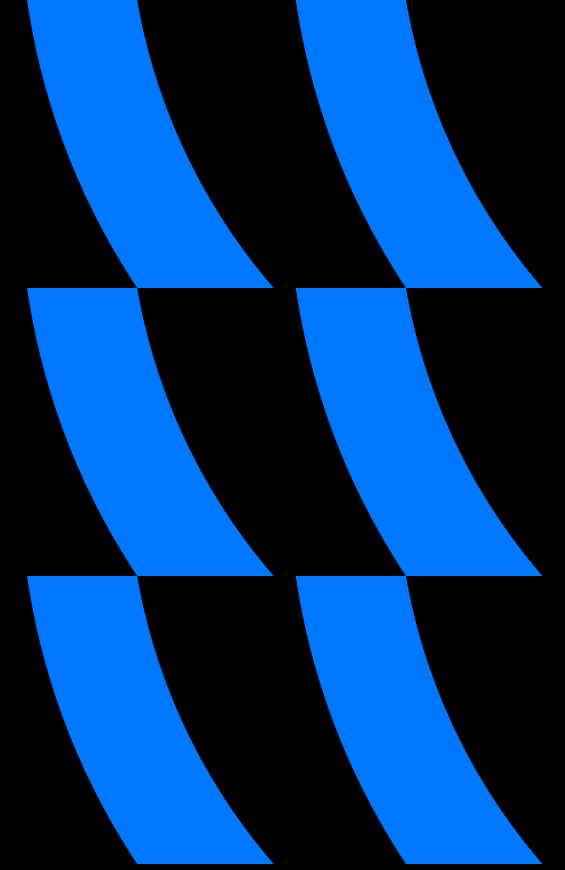
# Вмержили в мастер не ту ветку



# Человеческий фактор невнимательности



А как же устранить  
проблему?



# План доклада

1

Наш проект и его инфраструктура

4

Примеры использования ботов

2

Проблематика статусов задач в Jira

5

Заключение и итоги

3

Проектирование бота для автоматизации задач в бизнес логике

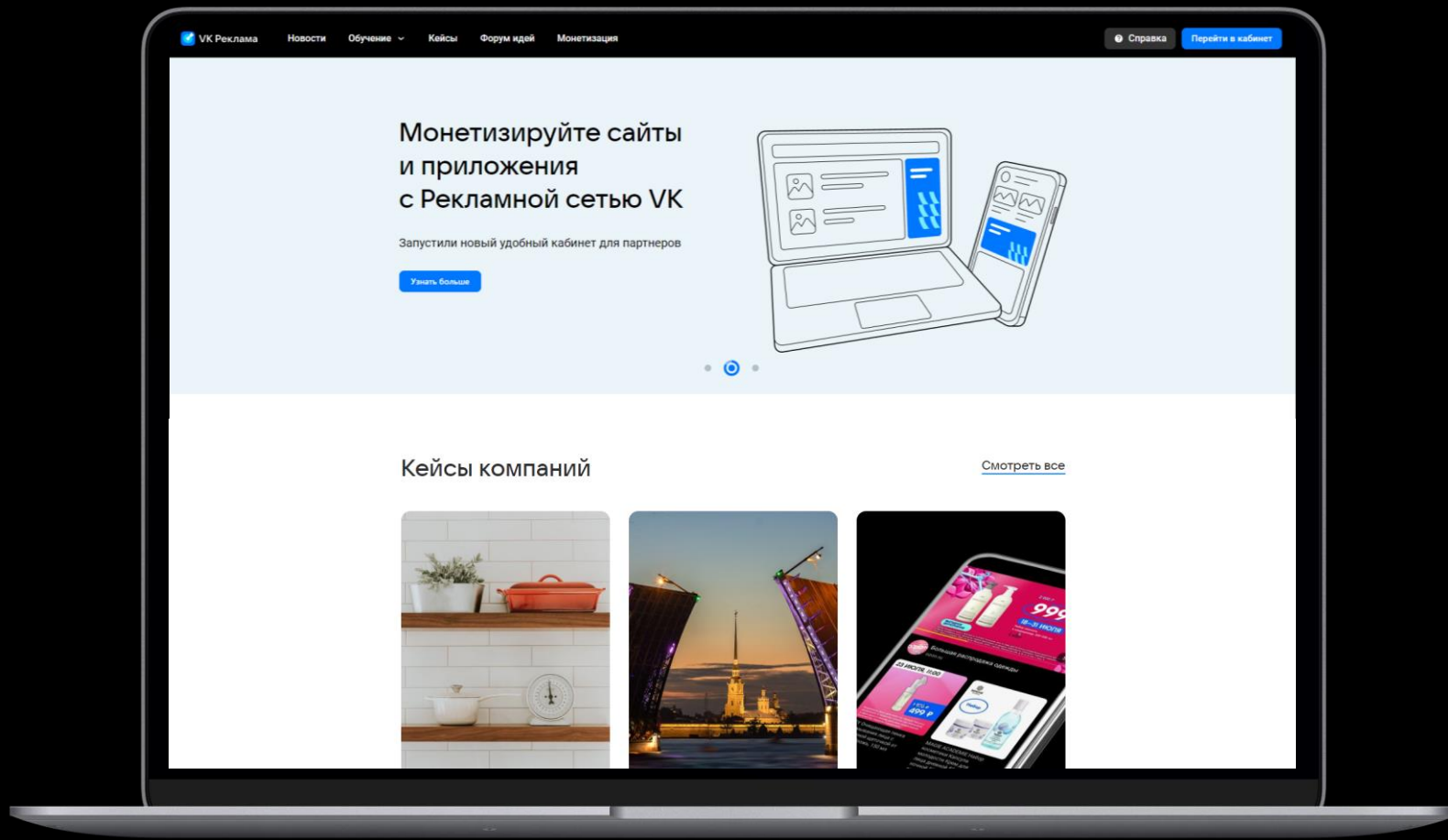
6

Вопросы / ответы



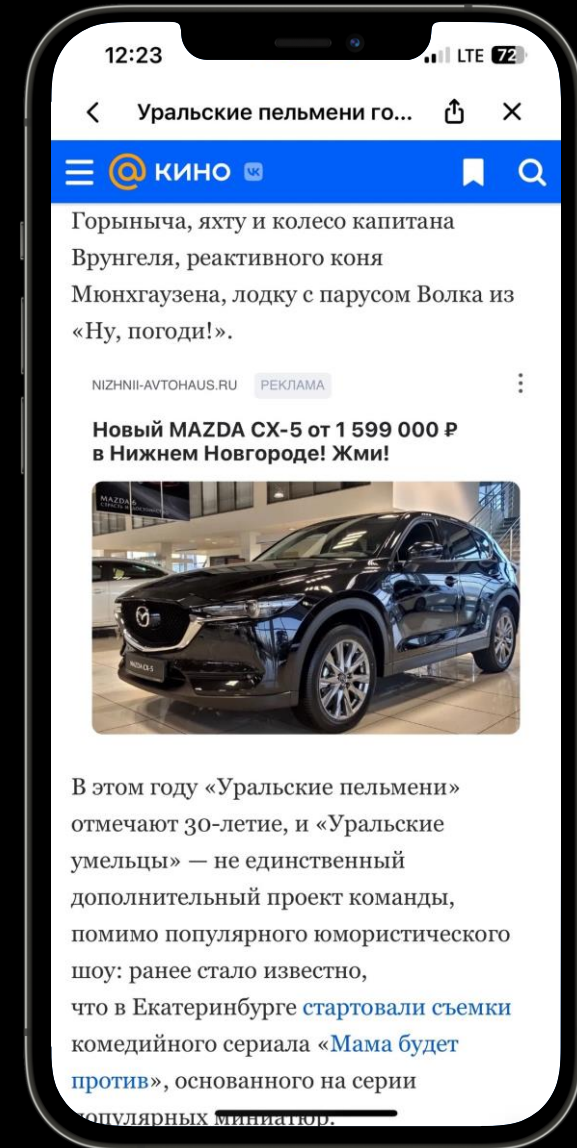
# Проект VK Реклама

- Позволяет взаимодействовать с пользователями соцсетей, классифайдов, медиапроектов и рекламной сети, превращая их в ваших клиентов
- Широкий выбор форматов рекламы, такие как баннеры, видеореклама, аудиореклама, shoppable ads и другие.



# myTarget SDK

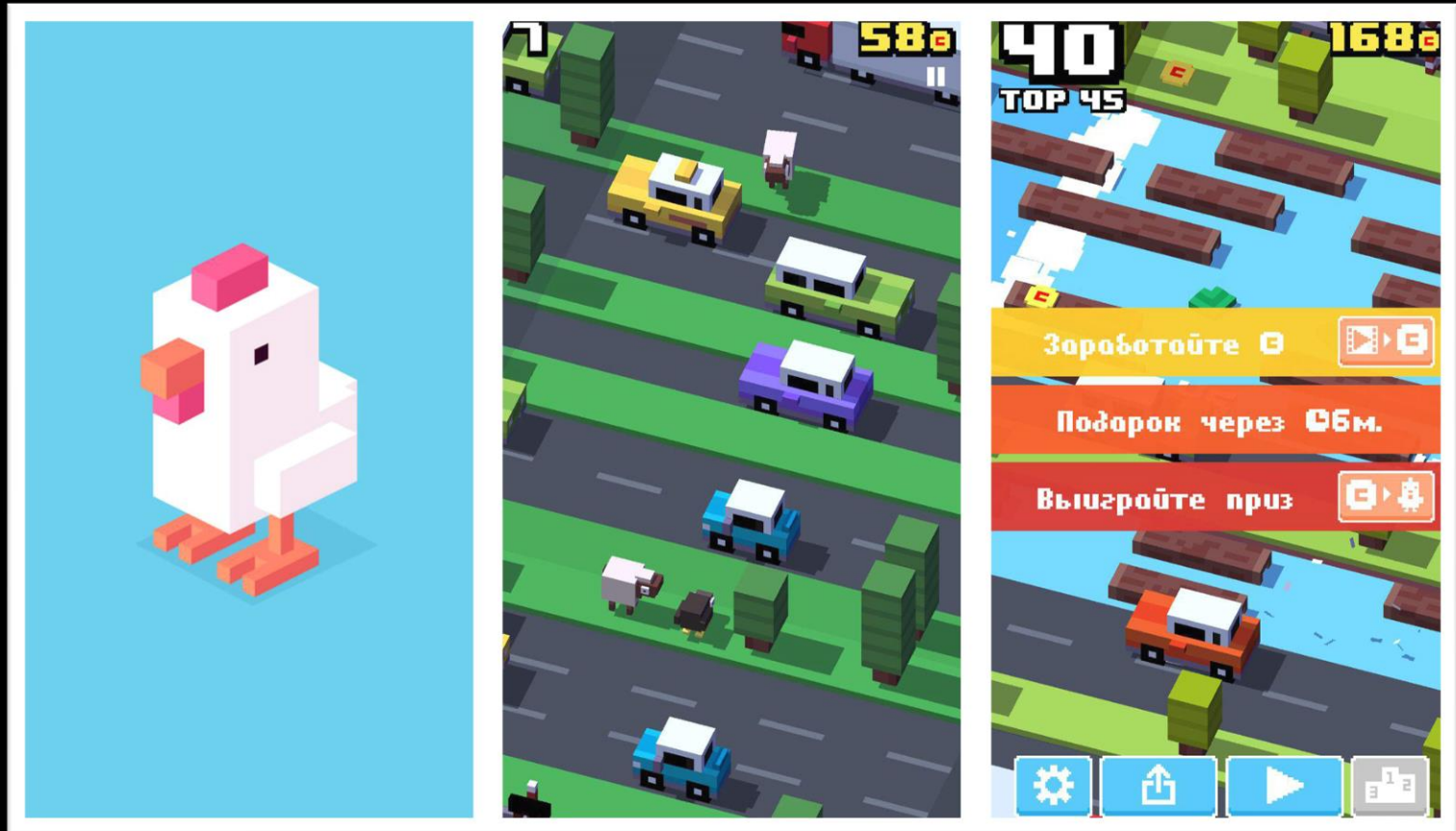
- Набор инструментов для разработчиков мобильных приложений, чтобы можно было в свое приложение интегрировать рекламу
- Чаще всего рекламу можно встретить между абзацами в статьях



# myTarget SDK

Игры, где можно посмотреть рекламу и получить бесплатные ресурсы

- Дополнительные жизни
- Дополнительные монетки
- Открыть нового персонажа
- Сократить время



# Из чего состоит наш проект?



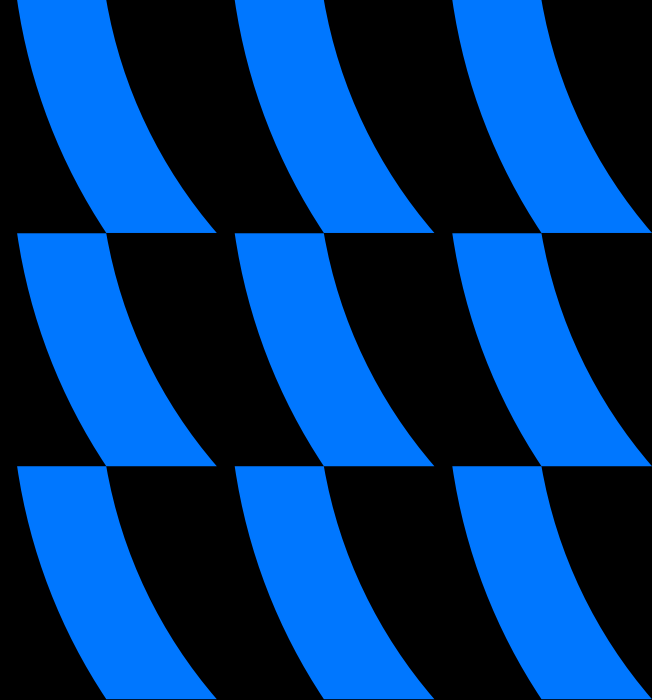
myTracker – сбор статистики

Mediation – интеграция рекламы с другими площадками

myTarget – показ рекламы



# Общий SDK



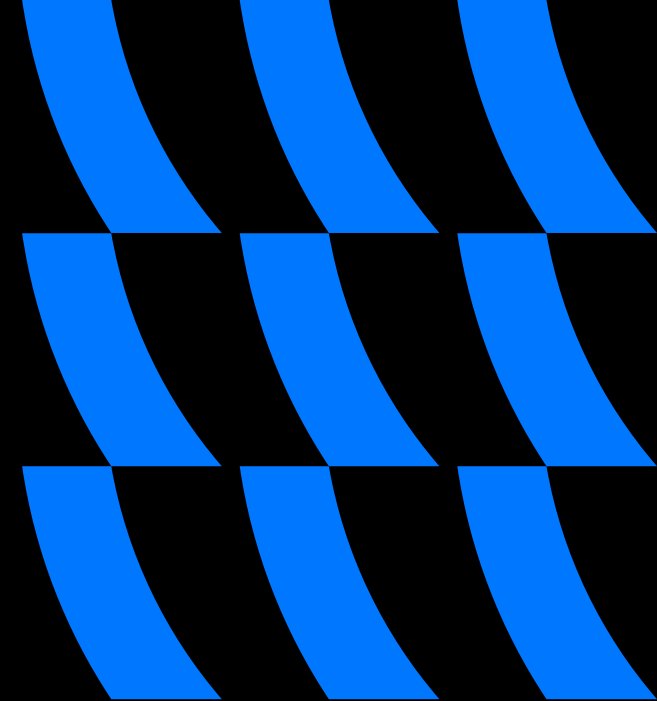
# Количество Jenkins Jobs

myTarget: iOS + Android

Интеграционные тесты: 3

Юнит тесты: 2

Сборка приложений: 2



# Количество Jenkins Jobs

## myTarget: iOS + Android

Интеграционные тесты: 3

Юнит тесты: 2

Сборка приложений: 2



## myTracker: iOS + Android + Unity (x2)

Интеграционные тесты: 4

Юнит тесты: 3

Сборка приложений: 3



# Количество Jenkins Jobs

Адаптеры: Yandex, IronSource, UnityAds, Facebook, Applovin

Интеграционные тесты: 10

Юнит тесты: 2

Сборка приложений: 2



# Как все это поместить в инфраструктуру?

**31**

Количество джоб в Jenkins

**14 Jobs**

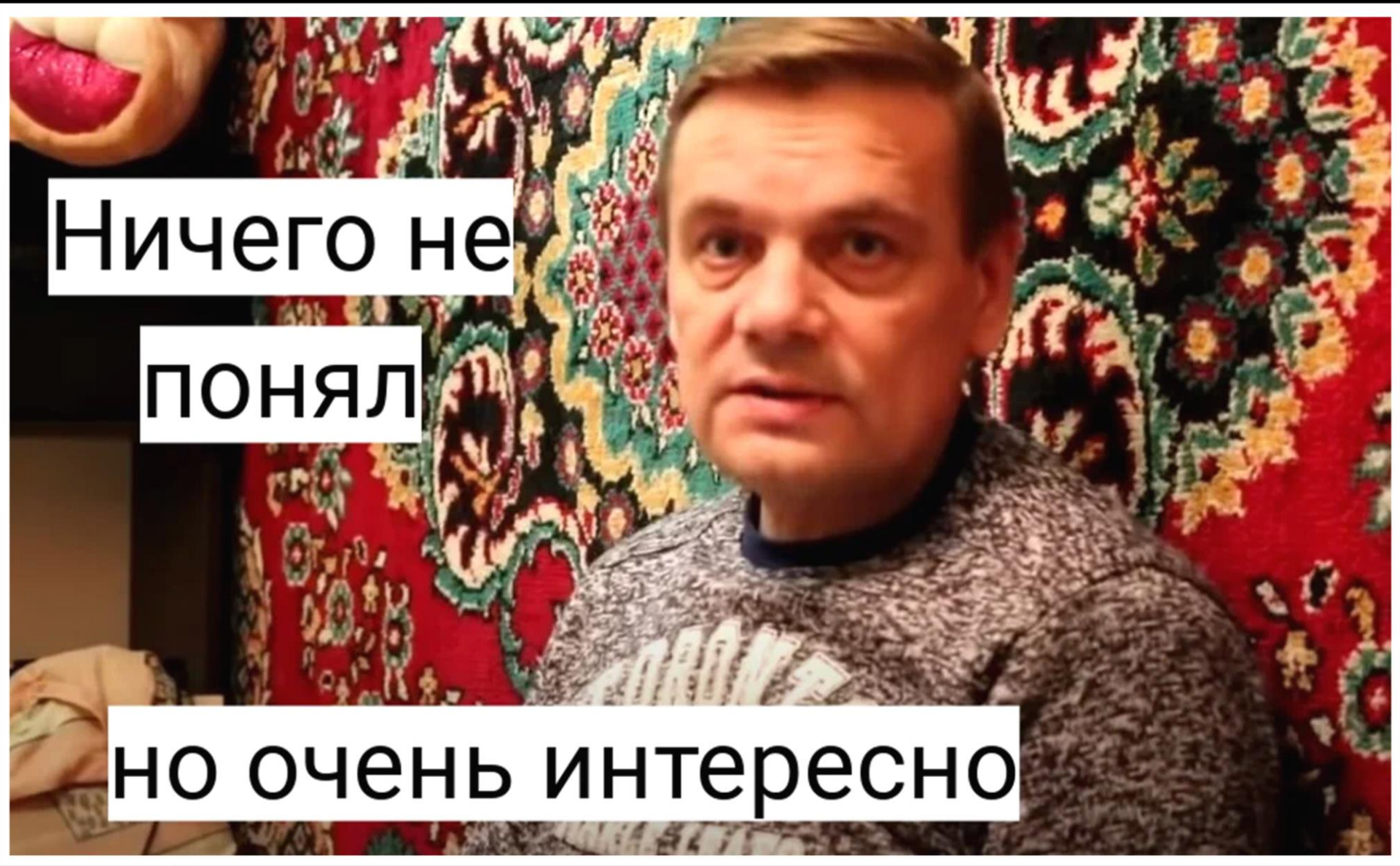
Для iOS платформ

**17 Jobs**

Для Android платформ

На скрине влезла  
ТОЛЬКО  
половина джоб

S	W	Name	Последний успех
🟢	🌟	Mediation_MITM_Building	24 дней #5
🟡	🌟	Mediation_MITM_Building_App_Builder	Н/Д
🟢	🌟	Mediation_MITM_Building	29 дней #3
🟢	🌟	Mediation_MITM_Building_Mediation_Building_Cli	21 дней #2
🟢	🌟	Mediation_MITM_Building_MITM_Building_Cli	21 дней #2
🟢	🌟	Mediation_MITM_Building_Building_Cli	21 дней #2
🟢	🌟	Mediation_MITM_Building	29 дней #2
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	3 часа 53 минути #10
🟢	🌟	Mediation_MITM_Building	6 дней 6 часов #10
🟢	🌟	Mediation_MITM_Building	4 часа 10 минути #10
🟢	🌟	Mediation_MITM_Building	12 дней #31
🟢	🌟	Mediation_MITM_Building	4 часа 32 минути #10
🟢	🌟	Mediation_MITM_Building	6 дней 17 часов #10
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	29 дней #3
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	20 дней #17
🟢	🌟	Mediation_MITM_Building	2 часа 25 минути #10
🟢	🌟	Mediation_MITM_Building	2 минуты 14 секунд #10
🟢	🌟	Mediation_MITM_Building	16 дней #4
🟡	🌟	Mediation_MITM_Building	7 дней 15 часов #10
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	6 дней 18 часов #10
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	10 дней #10
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟡	🌟	Mediation_MITM_Building	Н/Д
🟢	🌟	Mediation_MITM_Building	1 день 13 часов #10
🟡	🌟	Mediation_MITM_Building	Н/Д



Ничего не

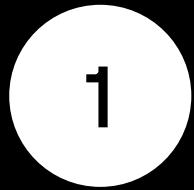
понял

но очень интересно

# Testing Flow



# Testing Flow



Запуск юнит  
тестов

# Testing Flow

1

Запуск юнит  
тестов

2

Сборка  
приложений для  
iOS и Android, в  
зависимости от  
задачи

# Testing Flow

1

Запуск юнит  
тестов

2

Сборка  
приложений для  
iOS и Android, в  
зависимости от  
задачи

3

Запуск  
интеграционных  
тестов

# Testing Flow

1

Запуск юнит  
тестов

2

Сборка  
приложений для  
iOS и Android, в  
зависимости от  
задачи

3

Запуск  
интеграционных  
тестов

4

Merge to Master

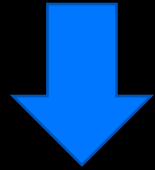
# Есть один нюанс

Код приложения и код автотестов  
находятся в разных репозиториях



# Есть один нюанс

Код приложения и код автотестов  
находятся в разных репозиториях

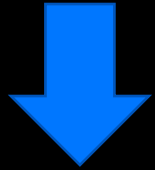


В задаче Jira может возникнуть путаница

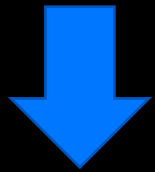


# Есть один нюанс

Код приложения и код автотестов  
находятся в разных репозиториях



В задаче Jira может возникнуть путаница

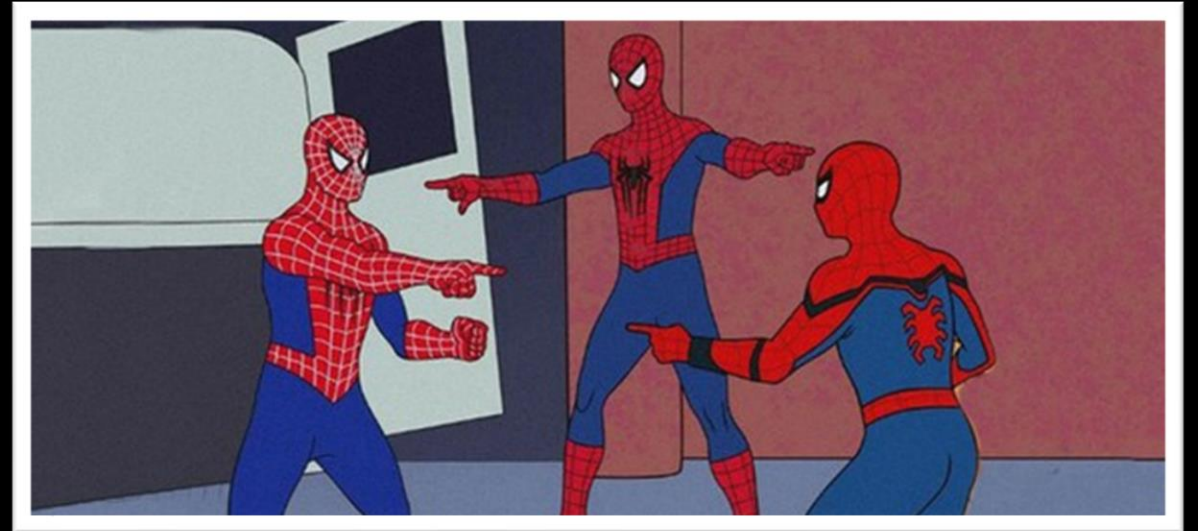


Актуальность данных может различаться



# Есть другой нюанс

Бывает ситуация, когда задача в статусе тестирования, то как понять запускались автотесты или нет?





















# Есть другой нюанс

Как экономить время и не перезапускать по 10 раз тесты?



# Дженкинс джоба

Запуск одного и того же по многу раз  
приводит к лишним временным  
затратам

Dashboard > Target_Appium_Android_Tests >	
 <u>master</u>	
13 сент. 2023 г., 3:18	
 <u>TASK-0001-add-new-test</u>	
13 сент. 2023 г., 1:54	
 <u>TASK-0001-add-new-test</u>	
13 сент. 2023 г., 0:29	
 <u>TASK-0001-add-new-test</u>	
12 сент. 2023 г., 23:26	
 <u>TASK-0001-add-new-test</u>	
12 сент. 2023 г., 23:17	
 <u>TASK-0001-add-new-test</u>	
12 сент. 2023 г., 23:15	
 <u>TASK-0001-add-new-test</u>	
12 сент. 2023 г., 22:20	
 <u>TASK-0001-add-new-test</u>	
12 сент. 2023 г., 21:31	

# 4 основные проблемы из-за человеческого фактора



# 4 основные проблемы из-за человеческого фактора

1

Исполнитель задачи  
в Jira забывает  
менять статус и  
комментировать  
прогресс



# 4 основные проблемы из-за человеческого фактора

1

Исполнитель задачи в Jira забывает менять статус и комментировать прогресс

2

Путаница в параметрах при ручном запуске джобы на CI/CD



# 4 основные проблемы из-за человеческого фактора

1

Исполнитель задачи в Jira забывает менять статус и комментировать прогресс

2

Путаница в параметрах при ручном запуске джобы на CI/CD

3

Не понятно какие действия запускались на CI/CD для задачи

# 4 основные проблемы из-за человеческого фактора

1

Исполнитель задачи в Jira забывает менять статус и комментировать прогресс

2

Путаница в параметрах при ручном запуске джобы на CI/CD

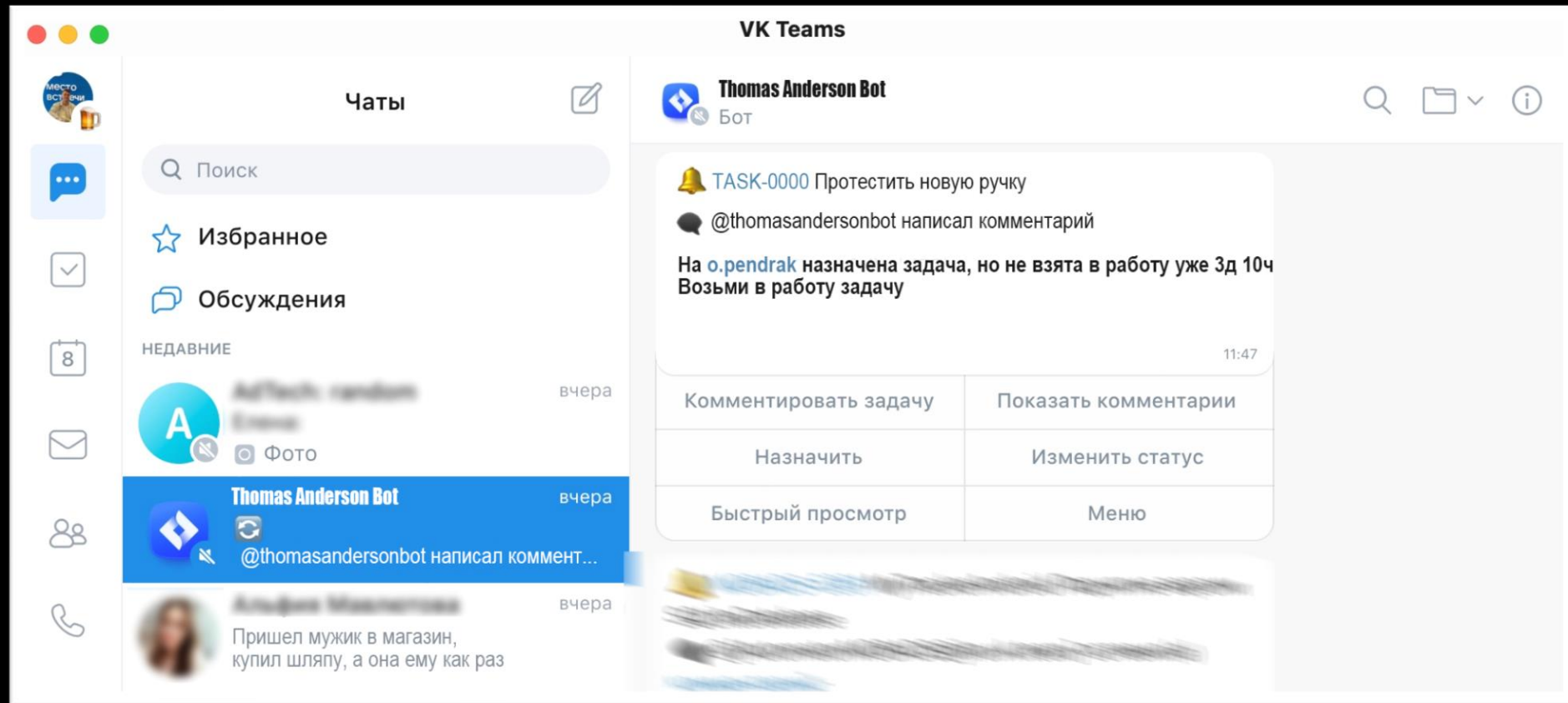
3

Не понятно какие действия запускались на CI/CD для задачи

4

В релизных процессах слишком много кода

# 1. Уведомлять исполнителя в мессенджер





## 2. Автоматизировать статусы задач в Jira

Самый крутой проект / TASK-0000 Протестить новую ручку

### Протестить новую ручку

Редактировать | Добавить комментарий | Назначить | Еще... | Мержить в мастер | Баг | Тестирование | Бизнес-процесс

Детали задачи

Тип:	Подзадача	Статус:	<b>В АВТОТЕСТЫ</b>
Приоритет:	Стандартный		(Посмотреть бизнес-процесс)
Затронуты версии:	Не заполнено	Решение:	Нет решения
Компоненты:	Не заполнено	Исправить в версиях:	Не заполнено
Метки:	API		

## 3. Вести логи активности в задаче

### ▼ Активность

Все Комментарии Журнал работ История изменений Активность Письма Переходы

Estimate (async mode)


---

▼  [Томас Андерсон](#) добавил(а) комментарий - 2023-01-23 17:29

 [iOS Unit Tests](#) прошли успешно, время выполнения составило 7 минут 22 секунд

Reply · Редактировать · Удалить · 

---

▼  [Томас Андерсон](#) добавил(а) комментарий - 2023-01-23 17:36

 [Appium iOS Tests](#) упали, время выполнения составило 27 минут 13 секунд

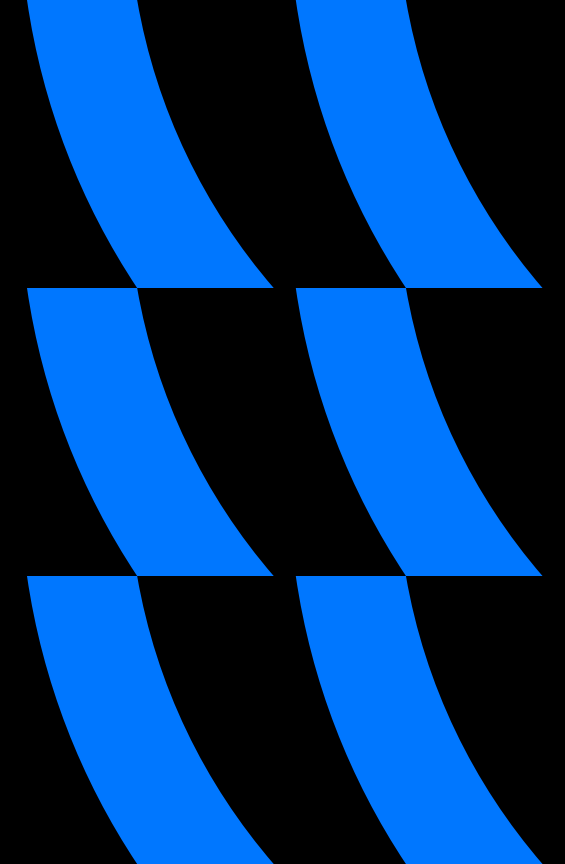
Reply · Редактировать · Удалить · 

## 4. Упростить процесс релиза

- Обновление Changelog
- Загрузка публичных демо приложений с использованием SDK
- Создание тегов и веток

# Построение автоматизации

- Кто будет пользоваться автоматизацией?

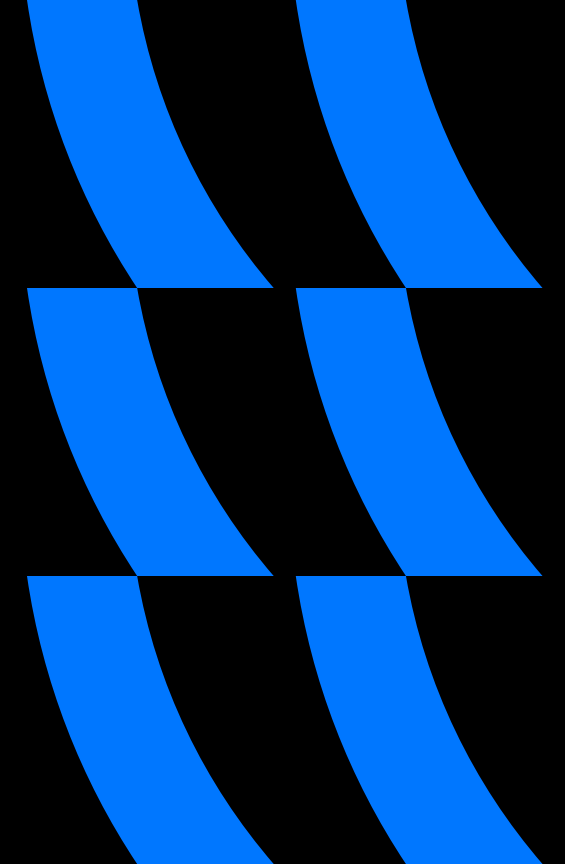


# Построение автоматизации

- Кто будет пользоваться автоматизацией?

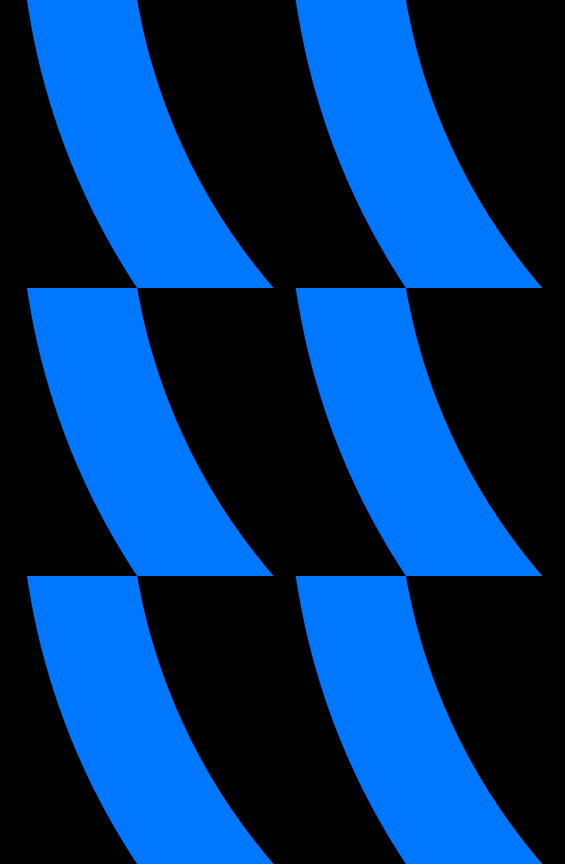
Основные заказчики:

- Разработчики
- Тестировщики
- Релиз инженеры



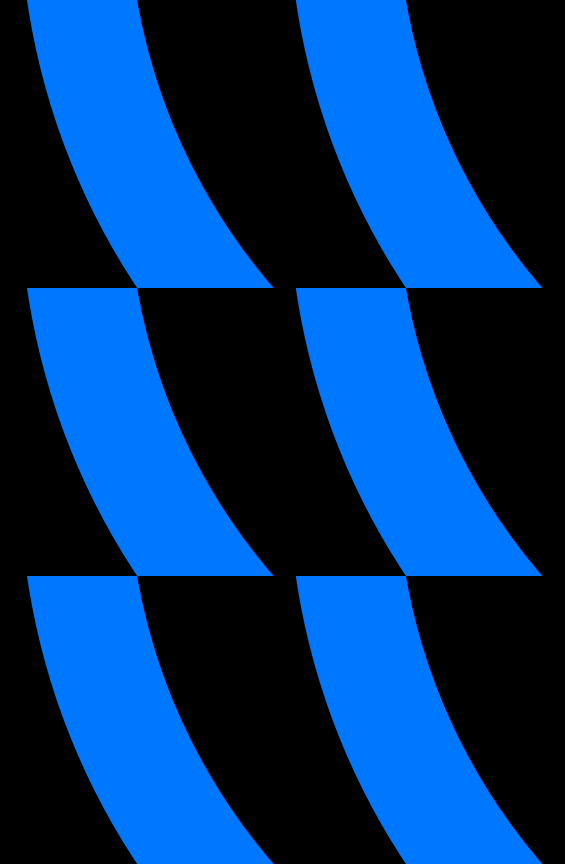
# Построение автоматизации

- Кто будет пользоваться автоматизацией?
- Как обеспечить качество автоматизации?



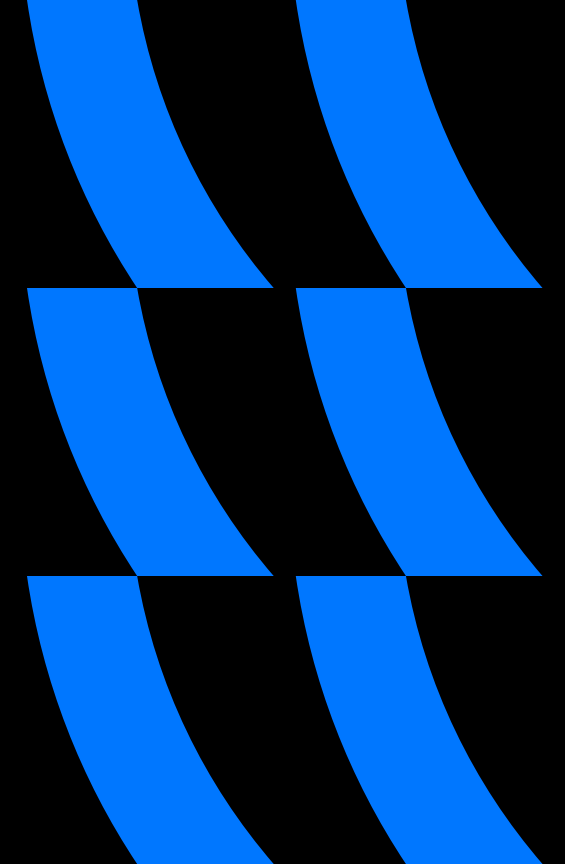
# Построение автоматизации

- Кто будет пользоваться автоматизацией?
- Как обеспечить качество автоматизации?
  - На каких моментах возникает чаще всего человеческий фактор невнимательности



# Построение автоматизации

- Кто будет пользоваться автоматизацией?
- Как обеспечить качество автоматизации?
- Кто будет ее поддерживать и развивать?





# На ком будет лежать автоматизация в Jira?



Знает продукт, но не умеет писать код

# На ком будет лежать автоматизация в Jira?



Знает продукт, но не умеет писать код



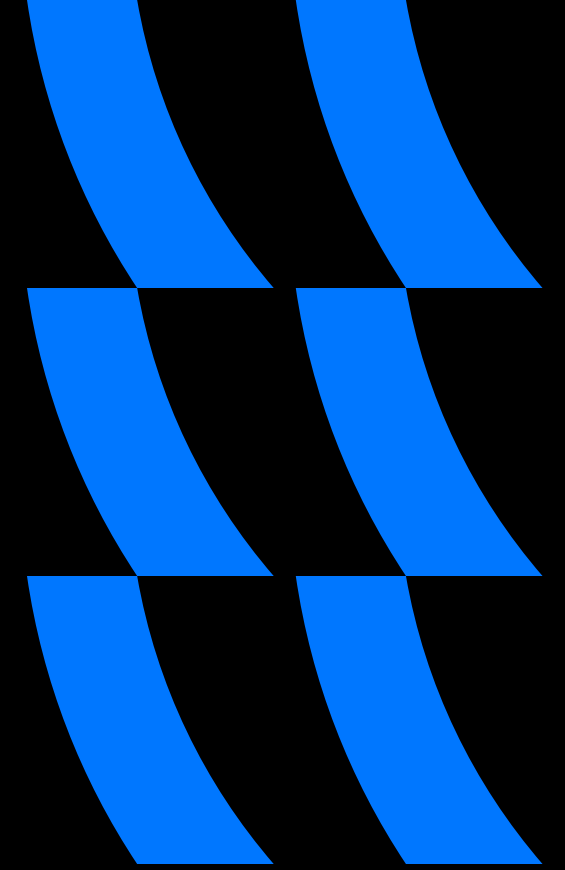
Знает продукт, умеет писать код

# На ком будет лежать автоматизация в Jira?



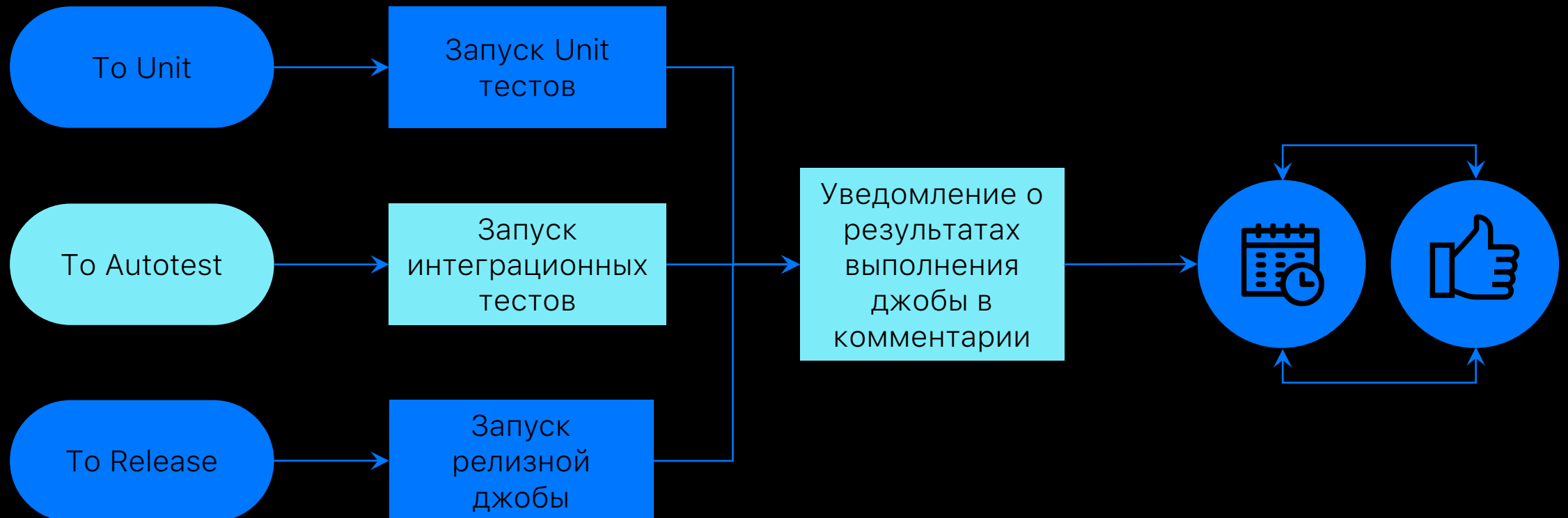
Знает продукт, пишет код и  
разбирается в инфраструктуре

# Как спроектировать логику?



# Автоматизация статусов в Jira задачах

В Jira есть своя API, которая позволяет получать информацию о задаче, в том числе о кастомных атрибутах



# API для получения информации о задаче

<https://jira.atlassian.com/rest/api/latest/issue/JRA-9>



## API для получения информации о задаче

<https://jira.atlassian.com/rest/api/latest/issue/JRA-9>

<https://jira.atlassian.com/rest/api/2/issue/126823>



```
{
  "expand": "renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations",
  "id": "126823",
  "self": "https://jira.atlassian.com/rest/api/2/issue/126823",
  "key": "JSWSERVER-2855",
  "fields": {
    "fixVersions": [
      {
        "self": "https://jira.atlassian.com/rest/api/2/version/19896",
        "id": "19896",
        "description": "",
        "name": "5.7.2",
        "archived": false,
        "released": true,
        "releaseDate": "2011-09-12"
      }
    ],
    "resolution": {
      "self": "https://jira.atlassian.com/rest/api/2/resolution/5",
      "id": "5",
      "description": "All attempts at reproducing this issue failed, or not enough information was available to reproduce the issue.",
      "name": "Cannot Reproduce"
    },
    "customfield_14830": "<div class=\"aui-message\">\r\n<p>\r\nThanks for taking the time to raise a Suggestion for Jira, Atlassian product roadmap. If you would like to know more about how the Jira Product Management team uses your input in our planning process into-what-s/ba-p/1029901\">our post on Atlassian Community</a>.<br>\r\n<br>\r\nBefore creating a new Suggestion, please search t your suggestion.<br>\r\n<br>\r\nKind Regards,<br>\r\nJira Product Management Team<br>\r\n</p>\r\n<span class=\"aui-icon icon-inf",
    "customfield_10510": "kyabut(kyabut)",
    "customfield_11436": "28935",
    "components": [
      {
        "self": "https://jira.atlassian.com/rest/api/2/component/13011",
        "id": "13011",
        "name": "Gadget"
      }
    ],
    "customfield_11435": "28922",
    "customfield_11437": "29852",
    "customfield_17130": null.
  }
}
```





# Логика наблюдения за статусами в Jira

# Логика наблюдения за статусами в Jira

1

Каждые 10 минут  
запускается  
джоба на Jenkins,  
которая получает  
все задачи на  
проекте

# Логика наблюдения за статусами в Jira

1

Каждые 10 минут  
запускается  
джоба на Jenkins,  
которая получает  
все задачи на  
проекте

2

Фильтрация задач  
по нужному  
пользователю и  
получение  
нужных  
параметров

# Логика наблюдения за статусами в Jira

1

Каждые 10 минут  
запускается  
джоба на Jenkins,  
которая получает  
все задачи на  
проекте

2

Фильтрация задач  
по нужному  
пользователю и  
получение  
нужных  
параметров

3

Запуск бота с  
**нужными**  
**аргументами,**  
**который**  
**триггернет**  
**определенную**  
Jenkins джобу

# Логика наблюдения за статусами в Jira

1

Каждые 10 минут запускается джоба на Jenkins, которая получает все задачи на проекте

2

Фильтрация задач по нужному пользователю и получение нужных параметров

3

Запуск бота с нужными аргументами, который триггернет определенную Jenkins джобу

4

**В случае успеха или неудачи, бот напишет в комментарии к задаче результаты выполнения джобы**

# Логика наблюдения за статусами в Jira

1

Каждые 10 минут запускается джоба на Jenkins, которая получает все задачи на проекте

2

Фильтрация задач по нужному пользователю и получение нужных параметров

3

Запуск бота с нужными аргументами, который триггернет определенную Jenkins джобу

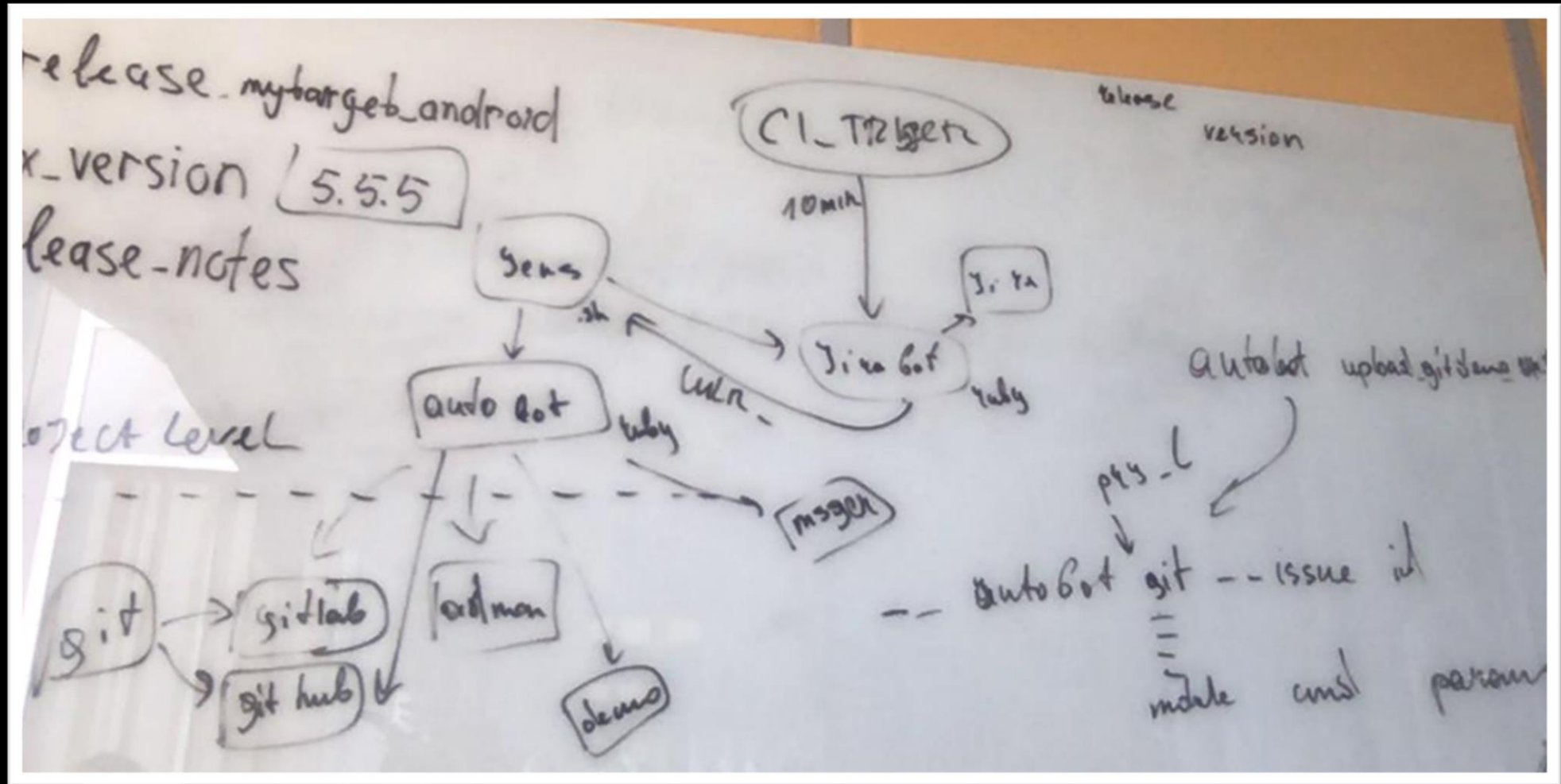
4

В случае успеха или неудачи, бот напишет в комментарии к задаче результаты выполнения джобы

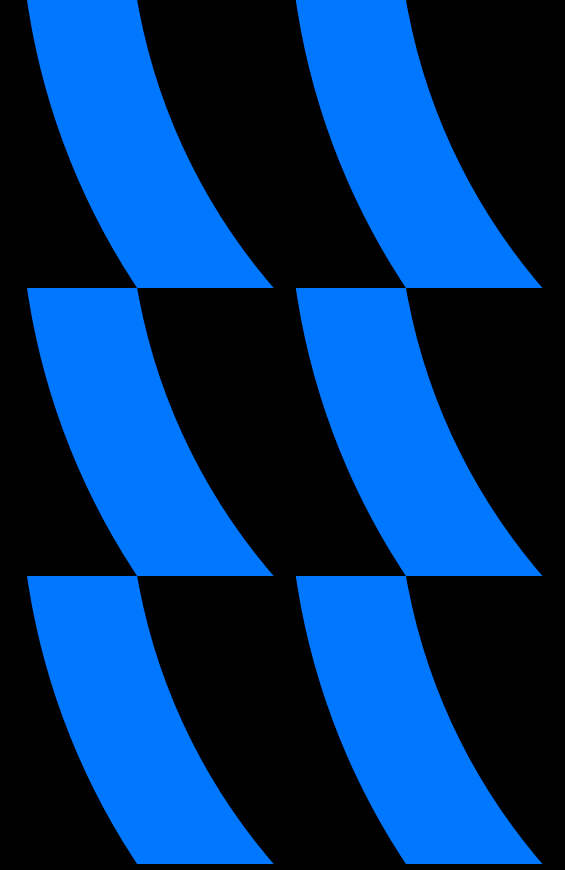
5

**Автоматически поменяется исполнитель, если джоба пройдена успешно**

# Проектирование возможностей в боте



# Как правильно спроектировать интерфейс?

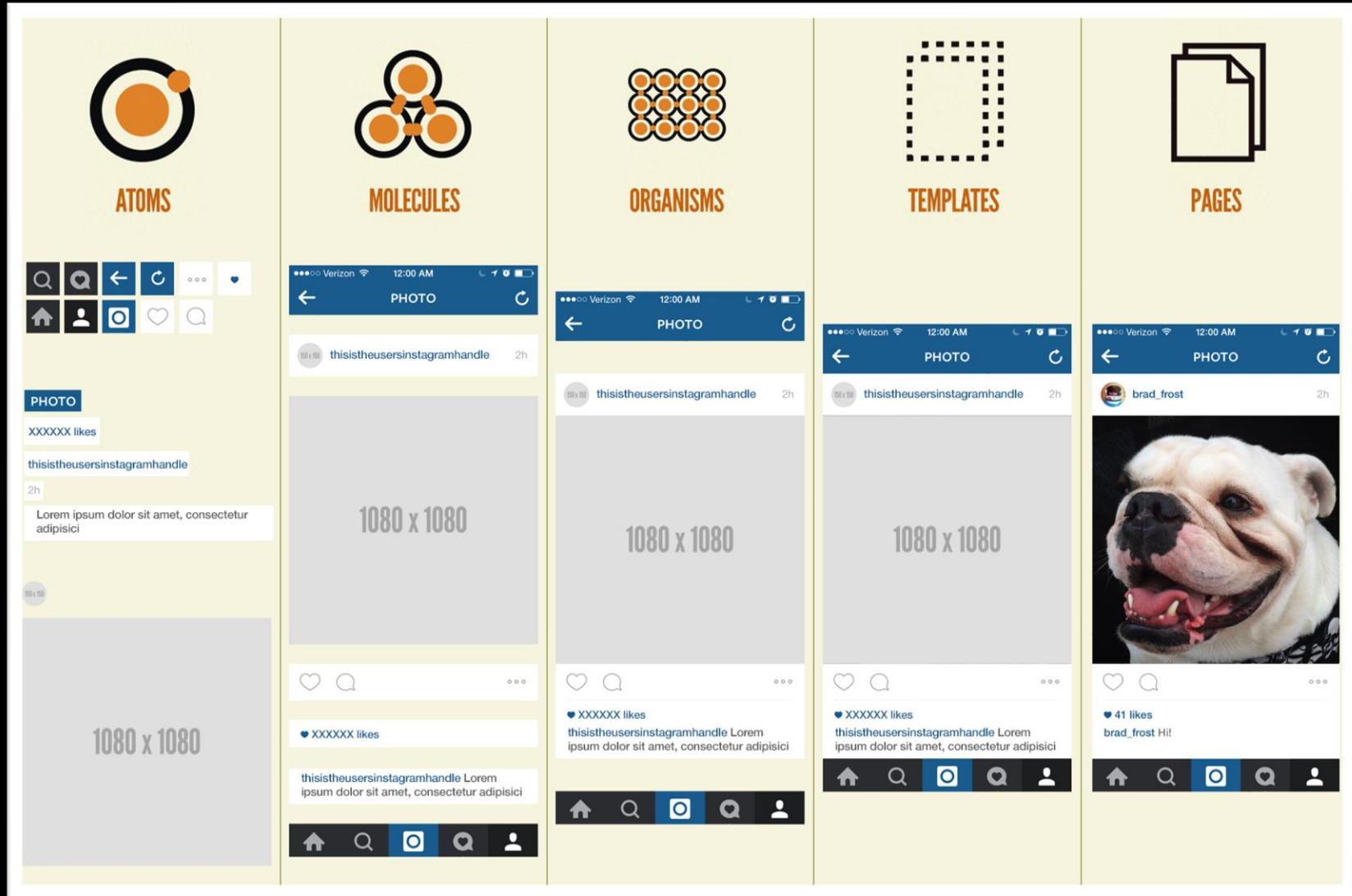




# Atomic Design Methodology



Подробнее про интерфейс



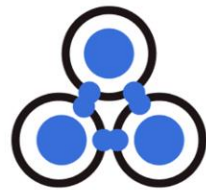
# Atomic в автоматизации

Конфиги



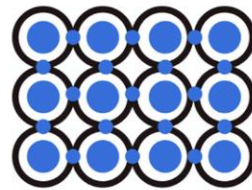
ATOMS

Общие функции



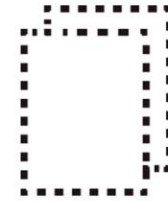
MOLECULES

Модули



ORGANISMS

Сценарии



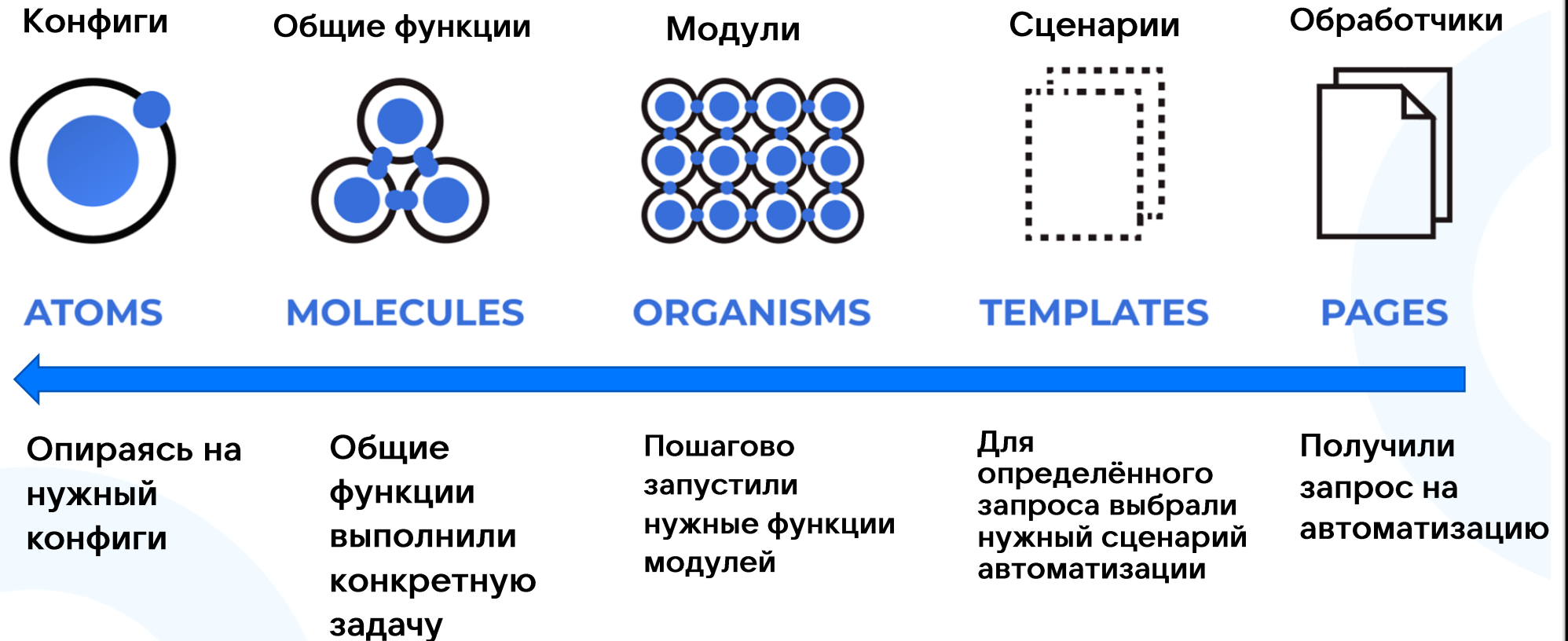
TEMPLATES

Обработчики



PAGES

# Обратный путь



# На примере

Конфиги



**ATOMS**

Нужные ссылки, пути к файлам, ключи и т.д.

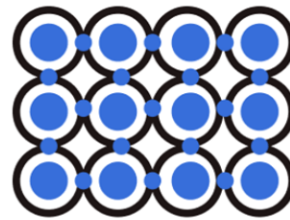
Общие функции



**MOLECULES**

Git, bash команды, работа с файловой системой

Модули



**ORGANISMS**

Используем модуль Github, Test, Deploy для клонирования, тестирования и развертки

Сценарии



**TEMPLATES**

По атрибутам из Jira выбираем сценарий запуска

Обработчики

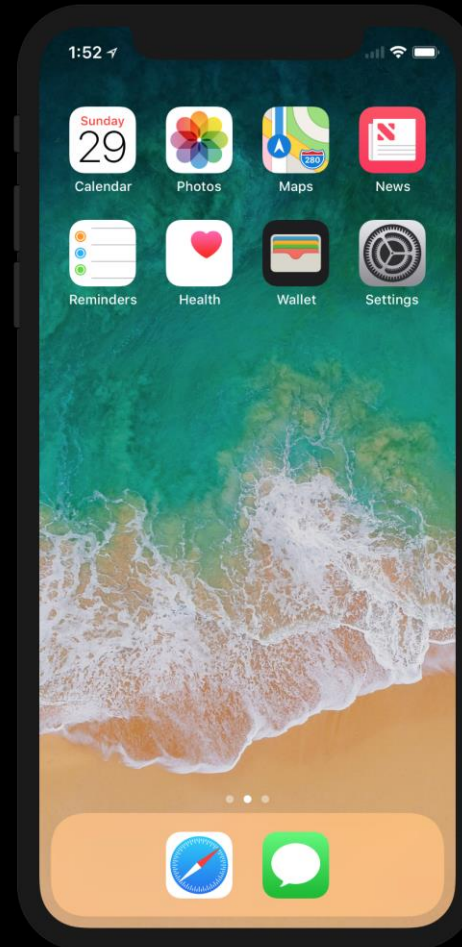


**PAGES**

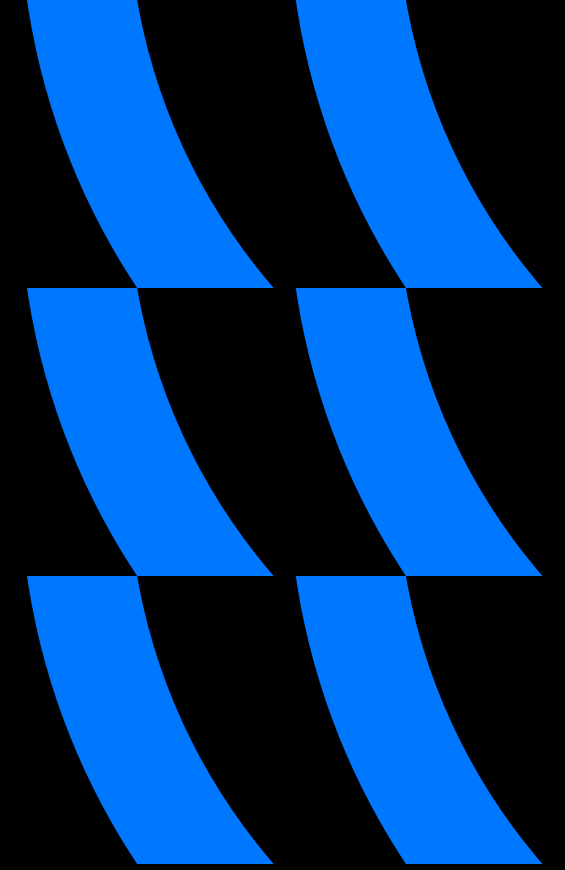
Получили запрос на запуск автотестов



# Окружение на Linux и на Mac



Какой язык  
программирования  
выбрать?



# Языки программирования

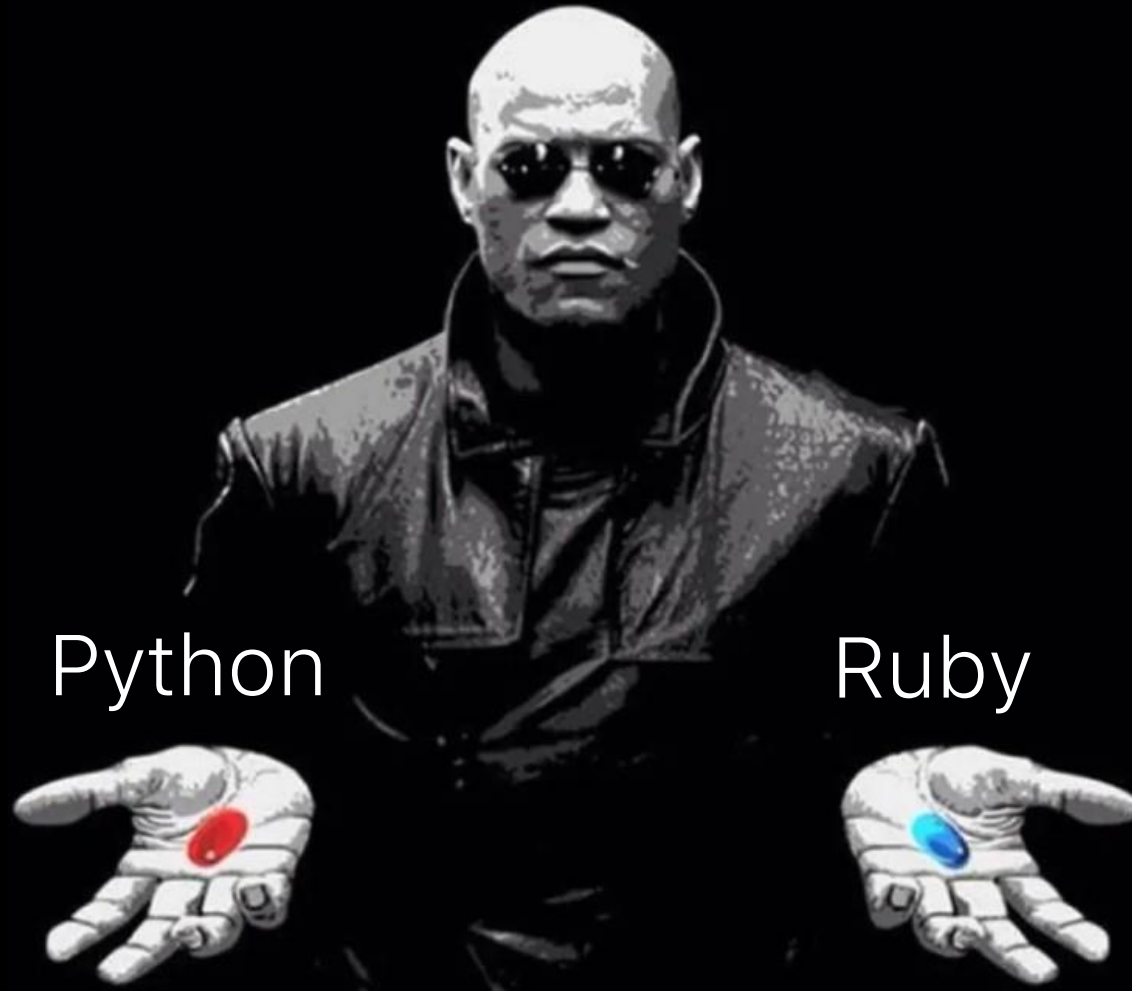
- Java
- C#
- Kotlin

# Языки программирования

- Java
- C#
- Kotlin
- Python
- Ruby
- Node.js



# Языки программирования



# А почему именно Ruby?



The Ruby logo consists of a large, dark gray circle. Inside this circle is a smaller, black circle. The word "Ruby" is written in a light blue, sans-serif font in the center of the black circle. Surrounding the black circle is a thin, light blue ring with four small, light blue circles positioned at the top, bottom, left, and right, resembling a ring of four gemstones.

Ruby

Просто  
использовать и  
писать код

Ruby

Просто  
использовать и  
писать код

Ruby

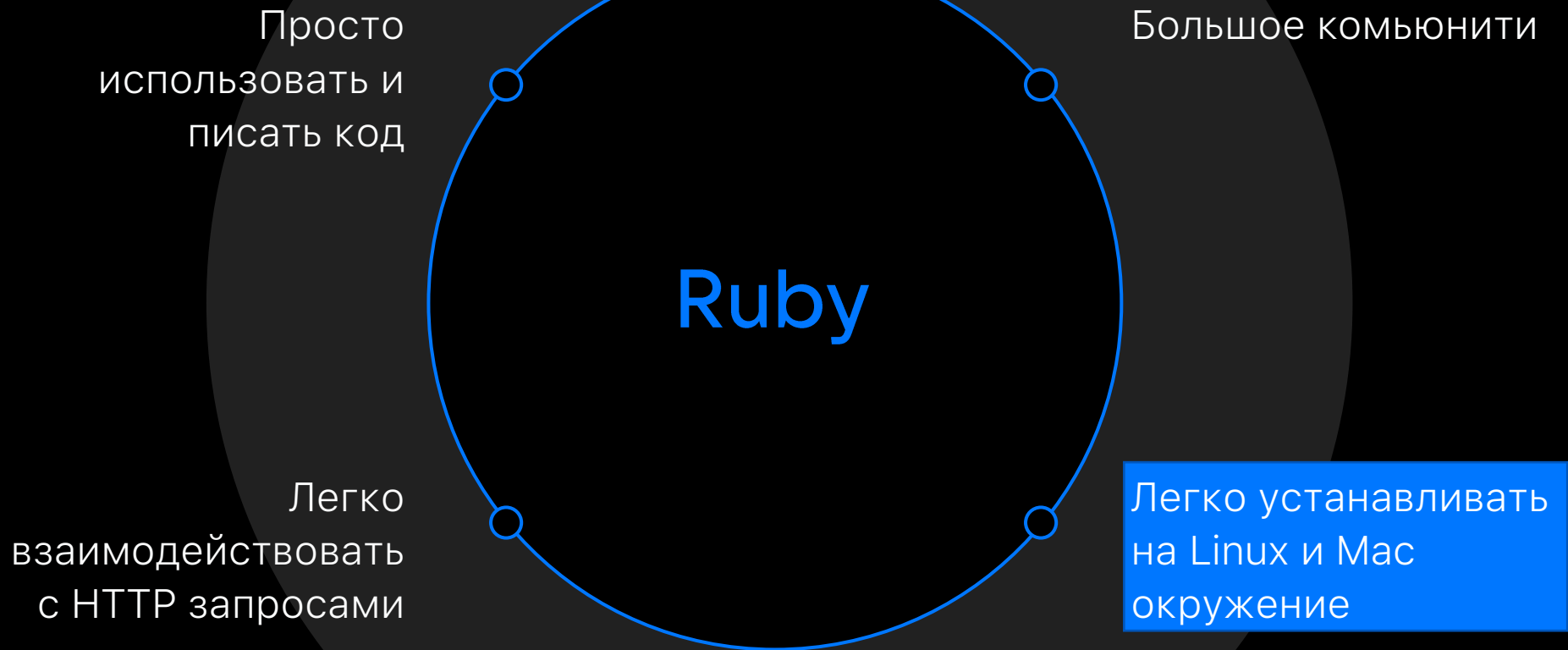
Большое комьюнити

Просто  
использовать и  
писать код

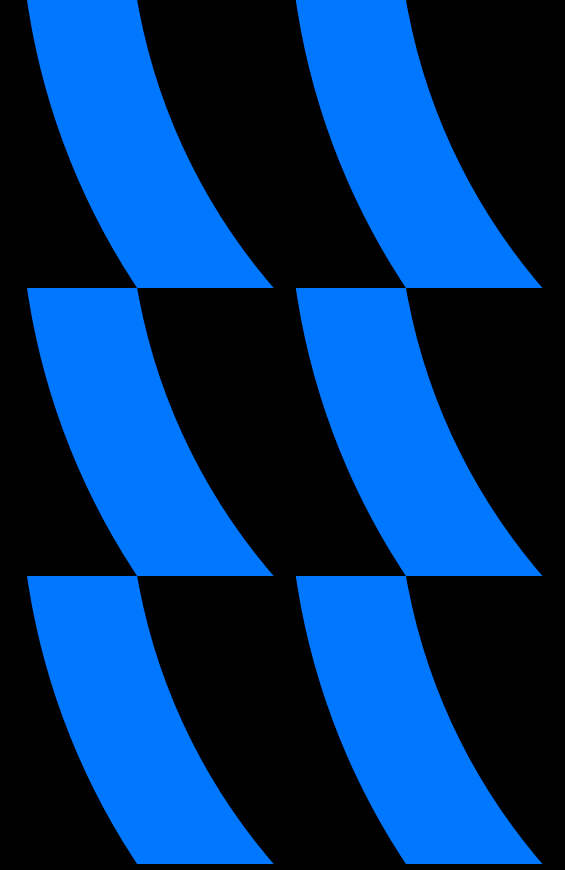
Большое комьюнити

Ruby

Легко  
взаимодействовать  
с HTTP запросами



Перейдем к  
технической  
части





# Логика наблюдения за статусами в Jira

1

Каждые 10 минут  
запускается  
джоба на Jenkins,  
которая получает  
все задачи на  
проекте

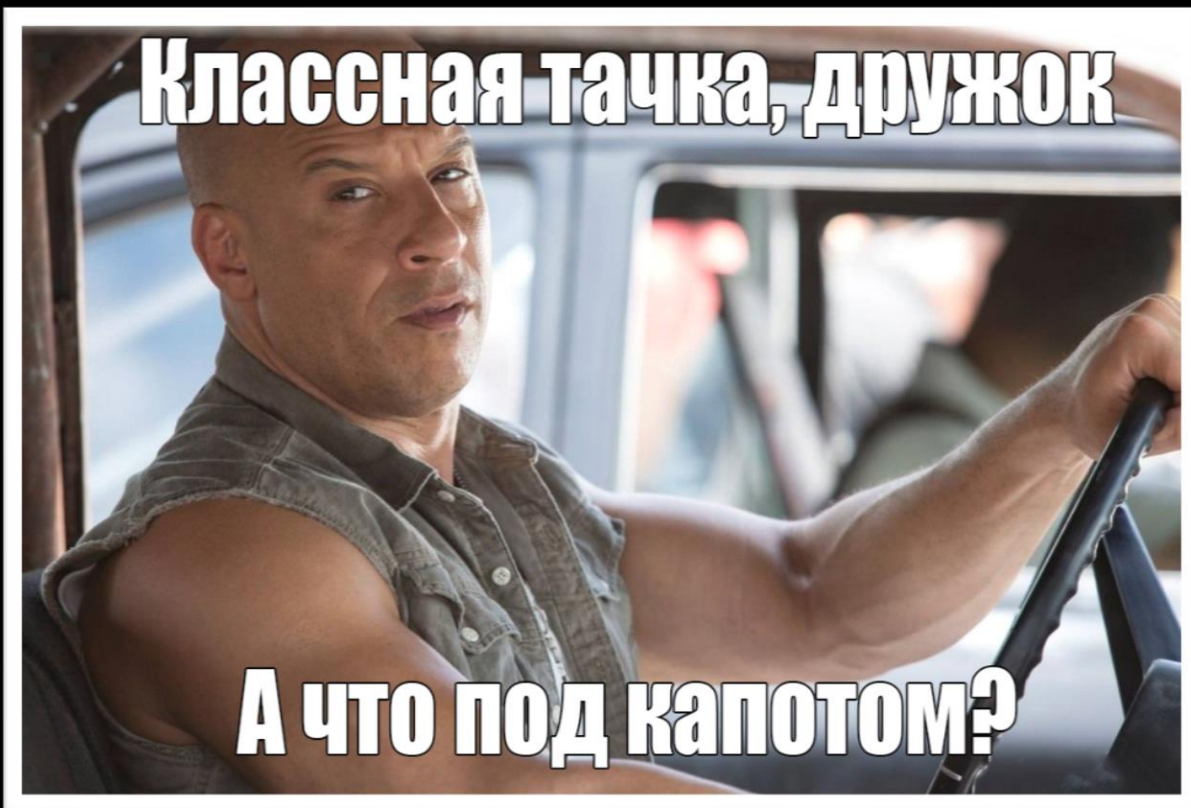
2

Фильтрация задач  
по нужному  
пользователю и  
получение  
нужных  
параметров

# Создать пользователя бота в Jira



# Ruby код



```
require 'rubygems'
require 'jira-ruby'

def get_jira_api_client
  options = {
    :site => 'https://jiraUrl.com',
    :username => 'some email',
    :password => 'API TOKEN',
    :auth_type => :basic
  }
  client = JIRA::Client.new(options)
  client
end
```

# Ruby код

- Получаем задачи, назначенные на бота

```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end
end
```

# Ruby код

- Получаем задачи, назначенные на бота
- Получаем компонентный класс с доступными методами, которые можно будет вызывать на выбранном проекте

```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end

  def start_jira_component
    tasks = JSON.parse(get_bot_assigned_tasks)
    (0..tasks.length).each do |number|
      task = tasks[number]
      task_component = task['fields']['comp']
      @project = find_class(task_component)
      run_by_status(task)
    end
  end
end
```


# Ruby код

- Получаем задачи, назначенные на бота
- Получаем компонентный класс с доступными методами, которые можно будет вызывать на выбранном проекте

```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end

  def start_jira_component
    tasks = JSON.parse(get_bot_assigned_tasks)
    (0..tasks.length).each do |number|
      task = tasks[number]
      task_component = task['fields']['comp']
      @project = find_class(task_component)
      run_by_status(task)
    end
  end
end
```



# Ruby код

```
class JiraListener
```

```
def get_bot_assigned_tasks
```

```
  pattern = 'assignee=currentuser()'
```

```
  tasks = client.Issue.jql(pattern)
```



MyTarget / TASK-0001

## [automation][myTarget][android] Обновить версию SDK в автотестах

✎ Редактировать

🔍 Добавить комментарий

Назначить

Еще... ▾

Return to New

Work

Close

Approve

### ▼ Детали задачи

Тип:  Задача

Приоритет: Стандартный

Затронуты версии: Не заполнено

Компоненты: autotest, mytarget\_android

Метки: ui

Automation required: No

links to branches: [TASK-0001-update-sdk](#)

Статус: **В РАБОТЕ** (Посмотреть бизнес-процесс)

Решение: Нет решения

Исправить в версиях: Не заполнено

```
end
```

```
end
```


# Ruby код

- Получаем задачи, назначенные на бота
- Получаем компонентный класс с доступными методами, которые можно будет вызывать на выбранном проекте

```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end

  def start_jira_component
    tasks = JSON.parse(get_bot_assigned_tasks)
    (0..tasks.length).each do |number|
      task = tasks[number]
      task_component = task['fields']['comp']
      @project = find_class(task_component)
      run_by_status(task)
    end
  end
end
```





# Ruby код

- Определяем компонентный класс по полю из Jira задачи через регулярные выражения

```
def find_class(component)
  compClass = nil
  if component =~ /myTracker-ios|tracker-ios/
    compClass = MyTrackerIOS.new
  elsif component =~ /autotest/
    && component =~ /(A|a)ndroid/
    && component =~ /(T|t)arget/
    compClass = MyTargetAndroidAutotest.new
  end
  return compClass
end
```

# Ruby код

- Определяем компонентный класс по пути из link задачи через

```
def find_class(component)
  compClass = nil
  if component =~ /myTracker-ios|tracker-ios/
    compClass = MyTrackerIOS.new
  elsif component =~ /autotest/
```



MyTarget / TASK-0001

## [automation][myTarget][android] Обновить версию SDK в автотестах

✎ Редактировать

🔍 Добавить комментарий

Назначить

Еще... ▾

Return to New

Work

Close

Approve

### ▼ Детали задачи

Тип:  Задача

Приоритет: 🚩 Стандартный

Затронуты версии: Не заполнено

Компоненты: autotest, mytarget\_android

Метки: ui

Automation required: No

links to branches: [TASK-0001-update-sdk](#)

Статус: В РАБОТЕ (Посмотреть бизнес-процесс)

Решение: Нет решения

Исправить в версиях: Не заполнено

# Ruby код

- Определяем компонентный класс по полю из Jira задачи через регулярные выражения

```
def find_class(component)
  compClass = nil
  if component =~ /myTracker-ios|tracker-ios/
    compClass = MyTrackerIOS.new
  elsif component =~ /autotest/
    && component =~ /(A|a)ndroid/
    && component =~ /(T|t)arget/
    compClass = MyTargetAndroidAutotest.new
  end
  return compClass
end
```

# Компонентный класс

- Содержит методы, которые можно будет вызвать через CLI для полного управления всеми процессами
- Каждый метод запускает Jenkins джобу с своими параметрами
- Можно указать запуск релизной джобы, деплойной джобы, интеграционных тестов и т.д.

```
class MyTrackerIOS
  JENKINS_URL = 'https://cool-jenkins-url/'.freeze
  UNIT_JOB_NAME = 'Tracker_Unit_IOS'.freeze

  def run_jenkins_job(url:)
    #add some headers
    http = Curl.post(url)
  end

  def run_unit_tests(branch:)
    url = "#{JENKINS_URL}/job/#{UNIT_JOB_NAME}/
    buildWithParameters?token=#{token}&branch=#{branch}"
    run_jenkins_job(url: url)
  end
end
```

# Запуск действий на основе статуса задачи

- Получение статуса задачи и атрибутов для запуска действий

```
def run_by_status(task_info:)  
    status = task_info['fields']['status']['name']  
    branch = task_info['fields']['branch']
```

# Запуск действий на основе статуса задачи

```
def run_by_status(task_info:)  
    status = task_info['fields']['status']['name']  
    branch = task_info['fields']['branch']
```



MyTarget / TASK-0001

## [automation][myTarget][android] Обновить версию SDK в автотестах

✎ Редактировать

🔍 Добавить комментарий

Назначить

Еще... ▾

Return to New

Work

Close

Approve

### ▼ Детали задачи

Тип:  Задача  
Приоритет: 🚩 Стандартный  
Затронуты версии: Не заполнено  
Компоненты: autotest, mytarget\_android  
Метки: ui  
Automation required: No

links to branches: [TASK-0001-update-sdk](#)

Статус: **В РАБОТЕ** (Посмотреть бизнес-процесс)

Решение: Нет решения

Исправить в версиях: Не заполнено

# Запуск действий на основе статуса задачи

- Получение статуса задачи и атрибутов для запуска действий

```
def run_by_status(task_info:)
  status = task_info['fields']['status']['name']
  branch = task_info['fields']['branch']
  if status == "To Unit"
    unit_tests(task_branch: branch)
  elsif status == "To Autotest"
    qa_tests(task_branch: branch)
  end
end
```

# Запуск действий на основе статуса задачи

- Получение статуса задачи и атрибутов для запуска действий
- Основная логика запуска одноименных методов из переменной `@project`

```
def run_by_status(task_info:)
  status = task_info['fields']['status']['name']
  branch = task_info['fields']['branch']
  if status == "To Unit"
    unit_tests(task_branch: branch)
  elsif status == "To Autotest"
    qa_tests(task_branch: branch)
  end
end

def unit_tests(task_branch:)
  @project.run_unit_tests(task_branch)
end

def qa_tests(task_branch:)
  @project.run_qa_tests(task_branch)
end
```



# Ruby код

- Получаем задачи, назначенные на бота
- Получаем компонентный класс с доступными методами, которые можно будет вызывать на выбранном проекте

```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end

  def start_jira_component
    tasks = JSON.parse(get_bot_assigned_tasks)
    (0..tasks.length).each do |number|
      task = tasks[number]
      task_component = task['fields']['comp']
      @project = find_class(task_component)
      run_by_status(task)
    end
  end
end
```

# Создание CLI аргументов

- Создаем аргументы для запуска нужного метода через командную строку

```
module JiraHelper
  program :version, '1.0.0'
  program :description, 'Helper for Jira'

  command :check_new_task do |c|
    c.syntax = 'jirahelper check_new_task'
    c.description = 'Run Jenkins Job by status'
    c.action { JiraHelper.new.check_new_task }
  end
end
```

# Создание CLI аргументов

- Создаем аргументы для запуска нужного метода через командную строку
- Инициализируем нужный класс

```
module JiraHelper
  program :version, '1.0.0'
  program :description, 'Helper for Jira'

  command :check_new_task do |c|
    c.syntax = 'jirahelper check_new_task'
    c.description = 'Run Jenkins Job by status'
    c.action { JiraHelper.new.check_new_task }
  end
end
```

```
module JiraHelper
  class JiraHelper
    def check_new_task
      JiraListener.new.start_jira_component
    end
  end
end
```

# Ruby код

- Получаем задачи, назначенные на бота
- Получаем компонентный класс с доступными методами, которые можно будет вызывать на выбранном проекте

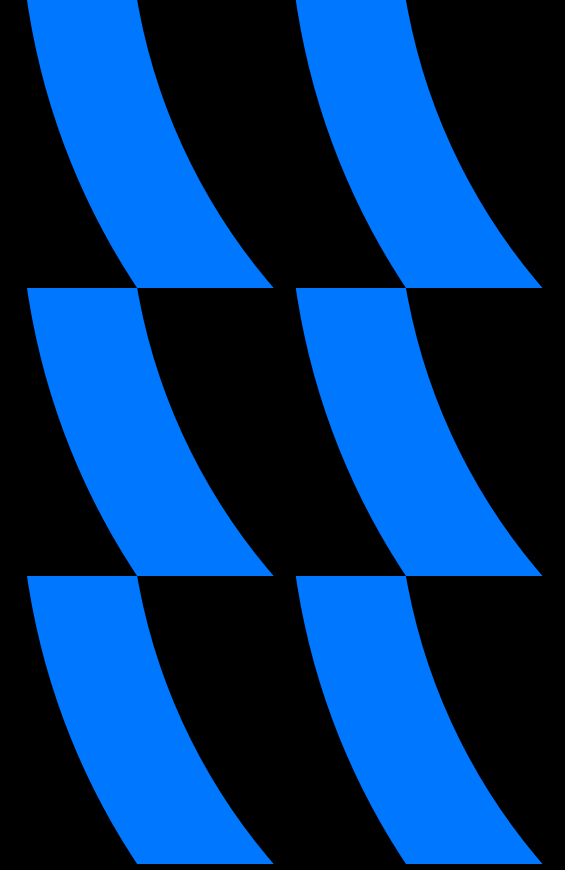


```
class JiraListener

  def get_bot_assigned_tasks
    pattern = 'assignee=currentuser()'
    tasks = client.Issue.jql(pattern)
    tasks.to_json
  end

  def start_jira_component
    tasks = JSON.parse(get_bot_assigned_tasks)
    (0..tasks.length).each do |number|
      task = tasks[number]
      task_component = task['fields']['comp']
      @project = find_class(task_component)
      run_by_status(task)
    end
  end
end
```

# Как устанавливать бота в тестовое окружение на CI?



# Создаем Gemfile

- Определяем откуда качать зависимости
- Указываем нужные основные библиотеки для установки

```
source "https://rubygems.org"
```

```
gem 'jira-ruby', :require => 'jira-ruby'
```

# Собираем проект

- Устанавливаем зависимости из Gemfile

```
bundle install
```

# Собираем проект

- Устанавливаем зависимости из Gemfile
- Используем сборщик rake, чтобы весь проект превратить в gem файл

```
bundle install
```

```
rake build
```



# Собираем проект

- Ис...
- вео...

Имя	Дата изменения	Разме
...	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	
Gemfile	22 марта 2023 г., 14:36	
> lib	22 марта 2023 г., 14:36	
▼ pkg	Сегодня, 13:45	
jirahelper-0.0.1.gem	Сегодня, 13:38	
Rakefile	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	
...	22 марта 2023 г., 14:36	

# Собираем проект

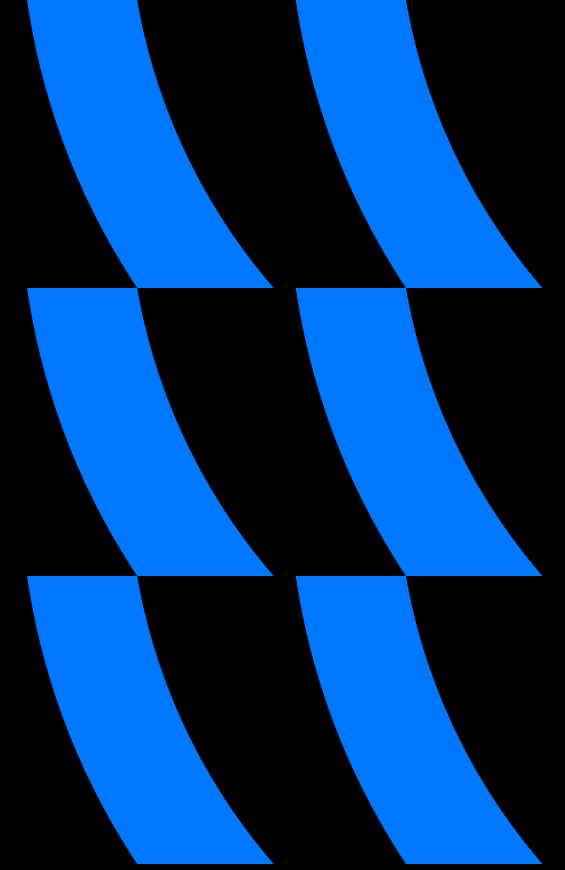
- Устанавливаем зависимости из Gemfile
- Используем сборщик rake, чтобы весь проект превратить в gem файл
- Устанавливаем gem в окружение

```
bundle install
```

```
rake build
```

```
gem install pkg/jirahelper*.gem
```

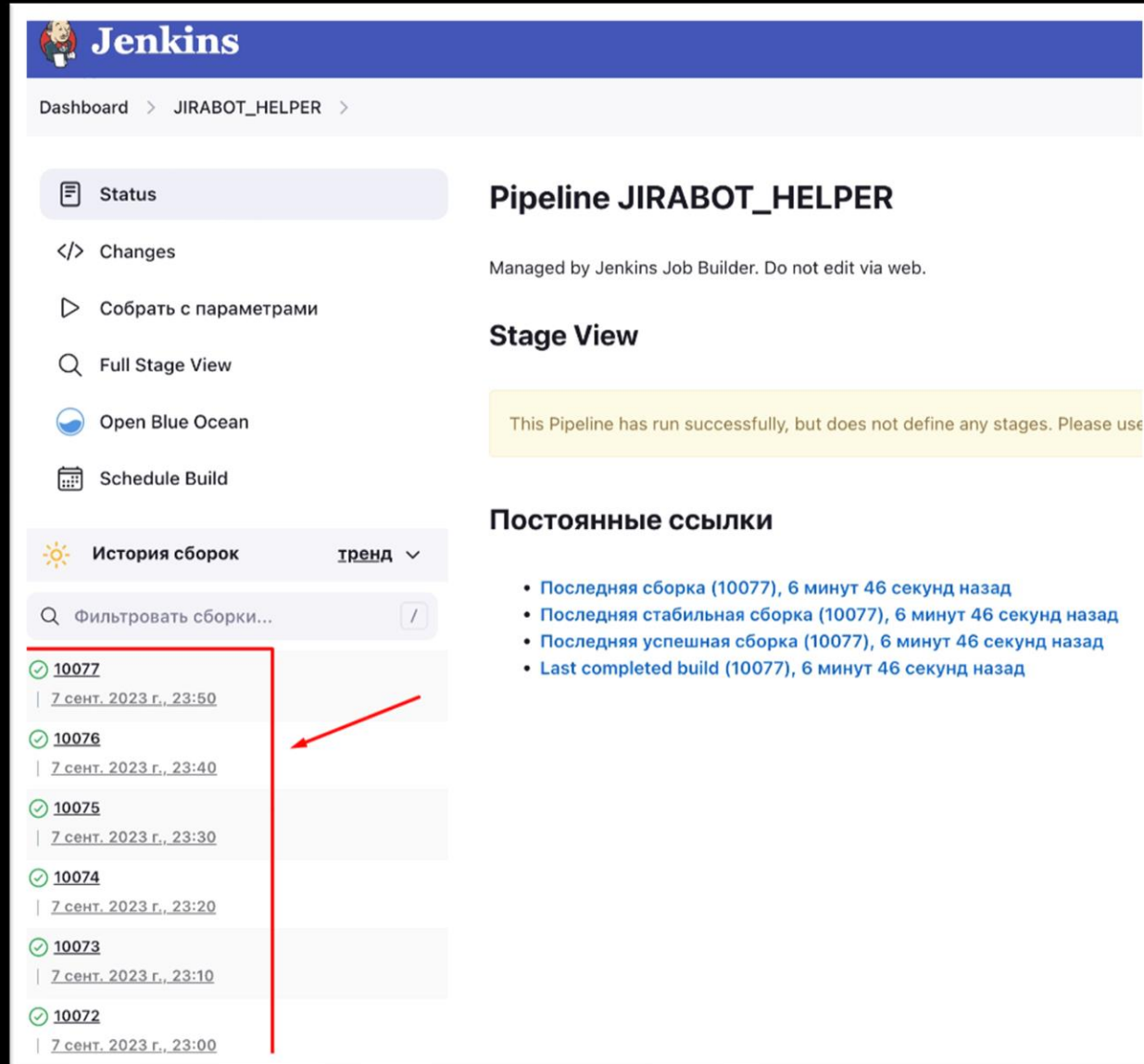
На выходе получаем 1  
команду для  
мониторинга статусов



## Команда бота для CI/CD скрипта в Jenkins

```
jirahelper check_new_task
```

# Создаем Jenkins джобу



**Jenkins**

Dashboard > JIRABOT\_HELPER >

Status

Changes

Собрать с параметрами

Full Stage View

Open Blue Ocean

Schedule Build

**История сборок** Тренд ▾

Фильтровать сборки...

- ✓ **10077**  
| [7 сент. 2023 г., 23:50](#)
- ✓ **10076**  
| [7 сент. 2023 г., 23:40](#)
- ✓ **10075**  
| [7 сент. 2023 г., 23:30](#)
- ✓ **10074**  
| [7 сент. 2023 г., 23:20](#)
- ✓ **10073**  
| [7 сент. 2023 г., 23:10](#)
- ✓ **10072**  
| [7 сент. 2023 г., 23:00](#)

## Pipeline JIRABOT\_HELPER

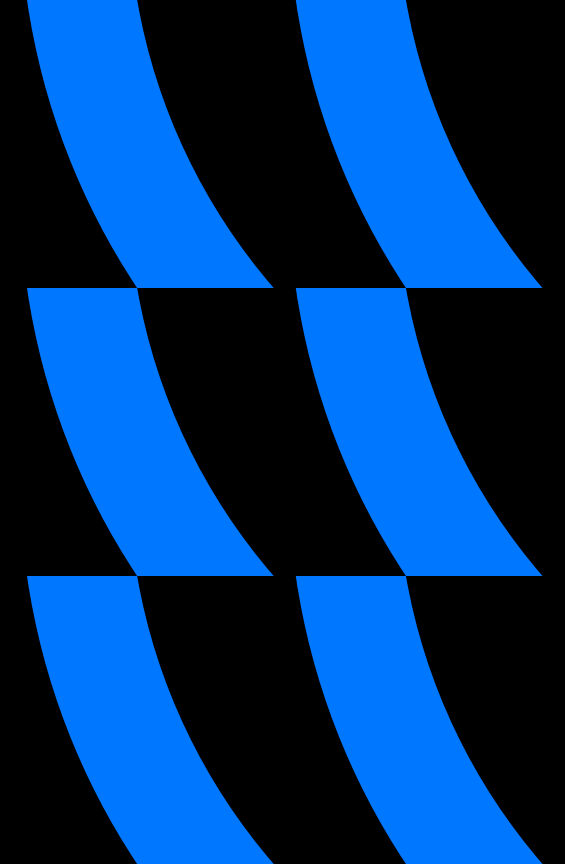
Managed by Jenkins Job Builder. Do not edit via web.

### Stage View

This Pipeline has run successfully, but does not define any stages. Please use

### Постоянные ссылки

- [Последняя сборка \(10077\), 6 минут 46 секунд назад](#)
- [Последняя стабильная сборка \(10077\), 6 минут 46 секунд назад](#)
- [Последняя успешная сборка \(10077\), 6 минут 46 секунд назад](#)
- [Last completed build \(10077\), 6 минут 46 секунд назад](#)



# Пример реальной задачи

**Самый интересный проект / TASK-9999**  
**Покрыть тестами новую фичу для Tracker iOS**

Редактировать | Добавить комментарий | Назначить | Еще... | Close | Approve | To Autotest | Электронная почта | Экспорт

**Детали задачи**

Тип:	<input checked="" type="checkbox"/> Задача	Статус:	<b>ТЕСТИРОВАНИЕ</b> (Посмотреть бизнес-процесс)
Приоритет:	Стандартный	Решение:	Нет решения
Затронуты версии:	Не заполнено	Исправить в версиях:	Не заполнено
Компоненты:	autotest, tracker-ios-sdk		
Метки:	Не заполнено		
Branch:	TASK-9999-cool-feature		
links to branches:	\$link.branch		

**Люди**

Исполнитель: Пендрак Олег  
Автор: Пендрак Олег  
Голоса: 0 [Голосовать за эту задачу](#)  
Наблюдатели: 1 [Начать наблюдение за этой задачей](#)

**Даты**

Создано: 2025-01-12 17:30  
Обновлено: 2025-01-15 17:17

**Сотрудники** +

**Описание**

У нас добавилась новая апишка по сбору информации о пользователе, нужно покрыть тестами все статусы /api/v2/user/collect

# Переводим в статус автотестов

Самый интересный проект / TASK-9999

## Покрыть тестами новую фичу для Tracker iOS

Редактировать | Добавить комментарий | Назначить | Еще... | Close | Approve | **To Autotest** ← | Электронная почта | Экспорт

**Детали задачи**

Тип:	<input checked="" type="checkbox"/> Задача	Статус:	<b>ТЕСТИРОВАНИЕ</b>
Приоритет:	Стандартный		(Посмотреть бизнес-процесс)
Затронуты версии:	Не заполнено	Решение:	Нет решения
Компоненты:	autotest, tracker-ios-sdk	Исправить в версиях:	Не заполнено
Метки:	Не заполнено		
Branch:	TASK-9999-cool-feature		
links to branches:	\$link.branch		

**Люди**

Исполнитель:	Пендрак Олег
Автор:	Пендрак Олег
Голоса:	0 <a href="#">Голосовать за эту задачу</a>
Наблюдатели:	1 <a href="#">Начать наблюдение за этой задачей</a>

**Даты**

Создано:	2025-01-12 17:30
Обновлено:	2025-01-15 17:17

**Описание**

У нас добавилась новая апишка по сбору информации о пользователе, нужно покрыть тестами все статусы /api/v2/user/collect

**Сотрудники** +

# Бот назначается исполнителем

The screenshot shows a task management interface for a project named "Самый интересный проект / TASK-9999". The task title is "Покрыть тестами новую фичу для Tracker iOS". The task status is "AUTOTESTING", which is highlighted with a red box and a red arrow. The task is assigned to "Томас Андерсон" (Thomas Anderson), also highlighted with a red box and a red arrow. The interface includes various action buttons like "Редактировать", "Добавить комментарий", "Назначить", "Close", "Approve", and "To Merge". The task details section shows the type as "Задача", priority as "Стандартный", and components as "autotest, tracker-ios-sdk". The description mentions a new API endpoint for user information collection.

Самый интересный проект / TASK-9999

## Покрыть тестами новую фичу для Tracker iOS

Редактировать | Добавить комментарий | Назначить | Еще... | Close | Approve | To Merge

Электронная почта | Экспорт

### Детали задачи

Тип:	<input checked="" type="checkbox"/> Задача	Статус:	<b>AUTOTESTING</b> (Посмотреть бизнес-процесс)
Приоритет:	Стандартный	Решение:	Нет решения
Затронуты версии:	Не заполнено	Исправить в версиях:	Не заполнено
Компоненты:	autotest, tracker-ios-sdk		
Метки:	Не заполнено		
Branch:	TASK-9999-cool-feature		
links to branches:	<a href="#">\$link.branch</a>		

### Люди

Исполнитель:	<b>Томас Андерсон</b>
Автор:	Пендрак Олег
Голоса:	0 <a href="#">Голосовать за эту задачу</a>
Наблюдатели:	1 <a href="#">Начать наблюдение за этой задачей</a>

### Даты

Создано:	2025-01-12 17:30
Обновлено:	2025-01-15 17:17

### Сотрудники

+

### Описание

У нас добавилась новая апишка по сбору информации о пользователе, нужно покрыть тестами все статусы `/api/v2/user/collect`



# Ждем пока бот подхватит задачу

**Jenkins**

Dashboard > JIRABOT\_HELPER >

Status

Changes

Собрать с параметрами

Full Stage View

Open Blue Ocean

Schedule Build

История сборок **ТРЕНД** ▾

Фильтровать сборки...

10077	7 сент. 2023 г., 23:50
10076	7 сент. 2023 г., 23:40
10075	7 сент. 2023 г., 23:30
10074	7 сент. 2023 г., 23:20
10073	7 сент. 2023 г., 23:10
10072	7 сент. 2023 г., 23:00

## Pipeline JIRABOT\_HELPER

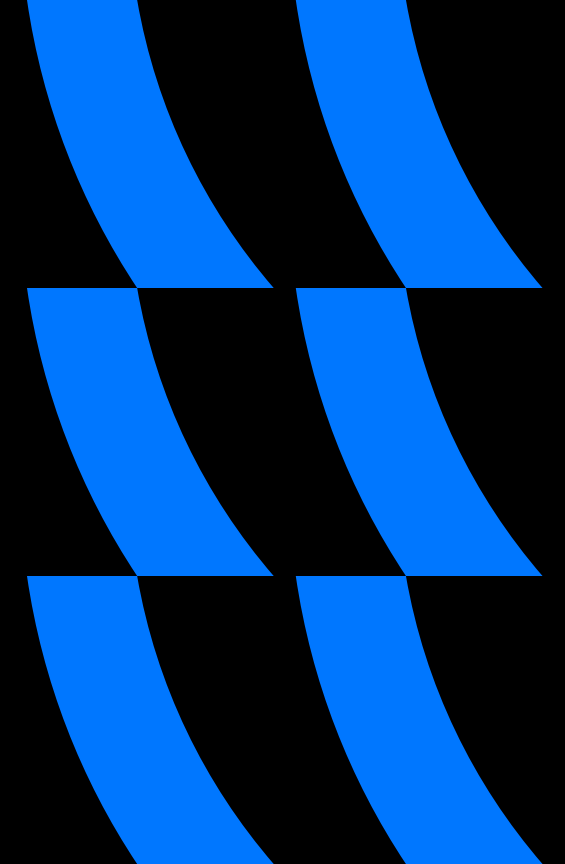
Managed by Jenkins Job Builder. Do not edit via web.

### Stage View

This Pipeline has run successfully, but does not define any stages. Please use

### Постоянные ссылки

- Последняя сборка (10077), 6 минут 46 секунд назад
- Последняя стабильная сборка (10077), 6 минут 46 секунд назад
- Последняя успешная сборка (10077), 6 минут 46 секунд назад
- Last completed build (10077), 6 минут 46 секунд назад




# Бот оставляет комментарий в задаче после выполнения джобы в Jenkins

## ▼ Активность

Все Комментарии Журнал работ История изменений Активность Письма Переходы

Estimate (async mode)

---

▼  Томас Андерсон добавил(а) комментарий - 2023-01-23 17:29

 iOS Unit Tests прошли успешно, время выполнения составило 7 минут 22 секунд

Reply · Редактировать · Удалить · 

---



▼  Томас Андерсон добавил(а) комментарий - 2023-01-23 17:36

 Appium iOS Tests упали, время выполнения составило 27 минут 13 секунд

Reply · Редактировать · Удалить · 

# Расширенные комментарии к задаче

Томас Андерсон добавил(а) комментарий - 2023-07-14 13:35

Name	Tracker iOS Appium Tests
Длительность	33 минут
Ветка тестов	master
Ветка приложения	TASK-9999-cool-feature
Версия приложения	1.12.1.61
Ссылка на логи джобы	
Ссылка на Allure	

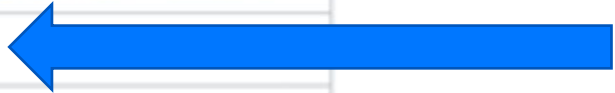
Reply · Редактировать · Удалить · 

# Расширенные комментарии к задаче

Томас Андерсон добавил(а) комментарий - 2023-07-14 13:35



Name	<a href="#">Tracker iOS Appium Tests</a>
Длительность	33 минут
Ветка тестов	master
Ветка приложения	<a href="#">TASK-9999-cool-feature</a>
Версия приложения	1.12.1.61
Ссылка на логи джобы	<a href="#">i</a>
Ссылка на Allure	<a href="#">-</a>

Reply · Редактировать · Удалить · 😊



# Расширенные комментарии к задаче

Томас Андерсон добавил(а) комментарий - 2023-07-14 13:35

Name	Tracker iOS Appium Tests
Длительность	33 минут
Ветка тестов	master
Ветка приложения	TASK-9999-cool-feature
Версия приложения	1.12.1.61
Ссылка на логи джобы	
Ссылка на Allure	

Reply · Редактировать · Удалить · 



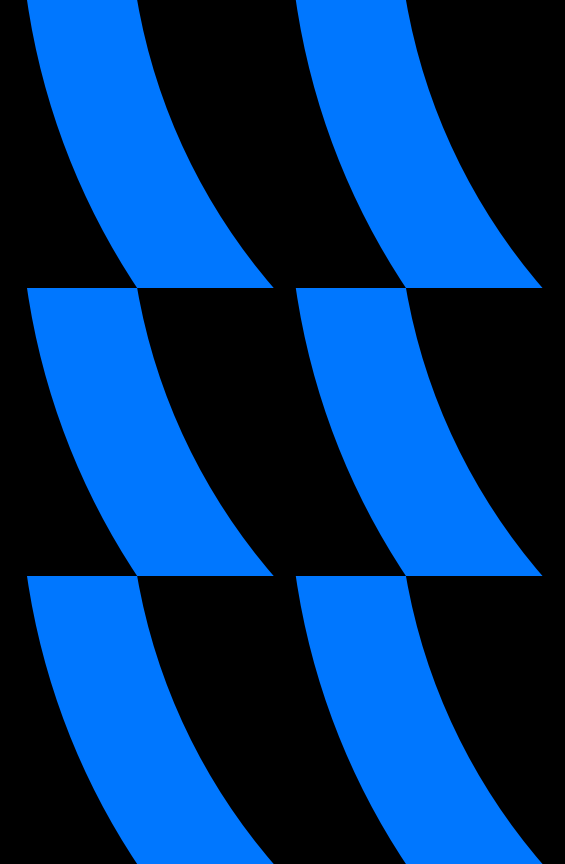
# Разработчик пушит изменения

☑️  Крутой разработчик добавил(а) комментарий - 2023-07-14 16:42

Для QA: фикс новых ошибок

- по нажатию кнопки Get теперь не отправляются логи
- добавили проверку на https при получении ссылок

Reply · Редактировать · Удалить · 😊



# Разработчик пушит изменения

Крутой разработчик добавил(а) комментарий - 2023-07-14 16:42

Для QA: фикс новых ошибок

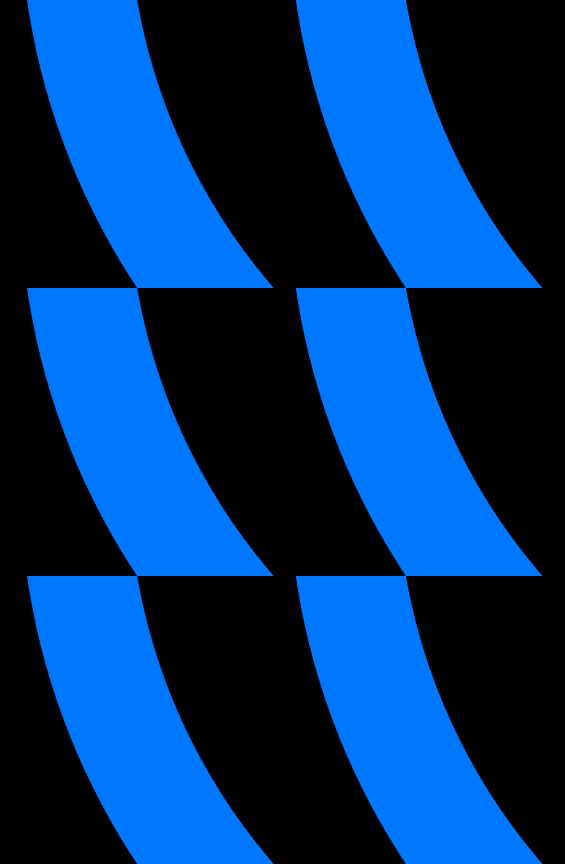
- по нажатию кнопки Get теперь не отправляются логи
- добавили проверку на https при получении ссылок

Reply · Редактировать · Удалить · 🗨️

Томас Андерсон добавил(а) комментарий - 2023-07-14 16:43

Name	Tracker iOS Appium Tests
Длительность	31 минута
Ветка тестов	master
Ветка приложения	<a href="#">TASK-9999-cool-feature</a>
Версия приложения	1.12.1.62
Ссылка на логи джобы	<a href="#">i</a>
Ссылка на Allure	<a href="#">+</a>

Reply · Редактировать · Удалить · 🗨️



# Разработчик пушит изменения

Крутой разработчик добавил(а) комментарий - 2023-07-14 16:42

Для QA: фикс новых ошибок

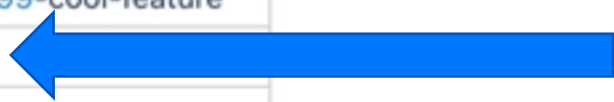
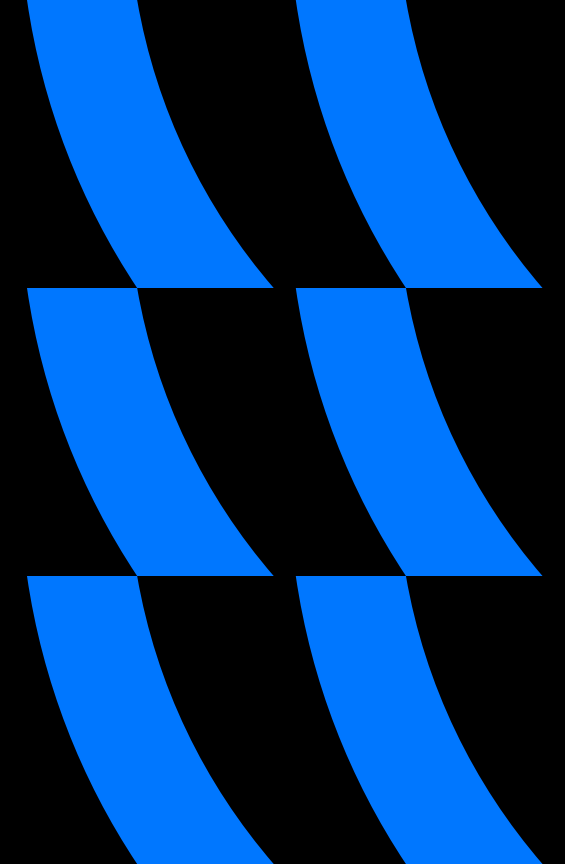
- по нажатию кнопки Get теперь не отправляются логи
- добавили проверку на https при получении ссылок

Reply · Редактировать · Удалить · 🗨️

Томас Андерсон добавил(а) комментарий - 2023-07-14 16:43

Name	Tracker iOS Appium Tests
Длительность	31 минута
Ветка тестов	master
Ветка приложения	TASK-9999-cool-feature
Версия приложения	1.12.1.62
Ссылка на логи джобы	<a href="#">i</a>
Ссылка на Allure	<a href="#">+</a>

Reply · Редактировать · Удалить · 🗨️





# Разработчик пушит изменения

Крутой разработчик добавил(а) комментарий - 2023-07-14 16:42

Для QA: фикс новых ошибок

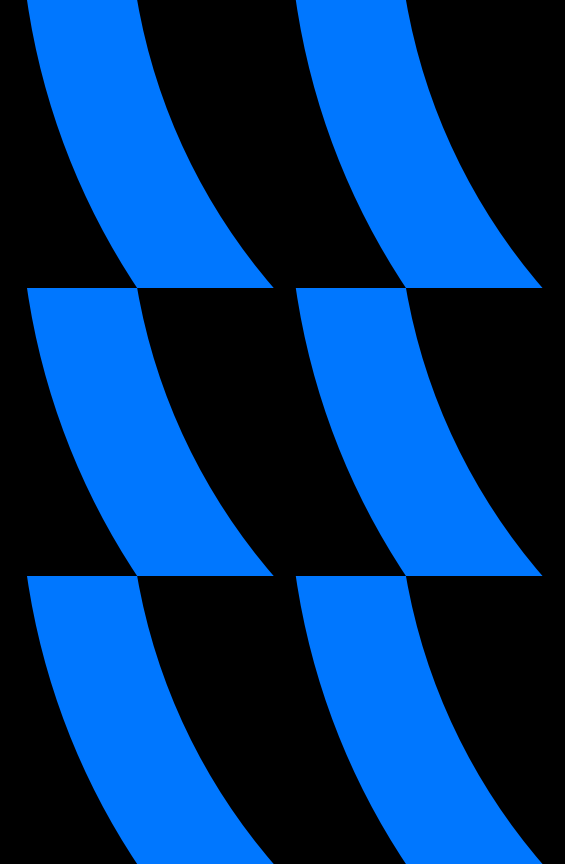
- по нажатию кнопки Get теперь не отправляются логи
- добавили проверку на https при получении ссылок

Reply · Редактировать · Удалить · 🗨️

Томас Андерсон добавил(а) комментарий - 2023-07-14 16:43

Name	Tracker iOS Appium Tests
Длительность	31 минута
Ветка тестов	master
Ветка приложения	TASK-9999-cool-feature
Версия приложения	1.12.1.62
Ссылка на логи джобы	<a href="#">i</a>
Ссылка на Allure	<a href="#">+</a>

Reply · Редактировать · Удалить · 🗨️





Чаты



Поиск



**Thomas Anderson Bot** вчера  
 На тебя назначена задача TASK-4912  
 разобрать с флаки тестами для ...



- Ирина...** 11:15
- Ирина...** 11:15
- Ирина...** 11:15
- Ирина...** 11:15
- Ирина...** 11:15

**Thomas Anderson Bot**  
 БОТ



7 авг.

@thomasAndersonBot

[TASK-9999](#)

Покрыть тестами новую фичу для Tracker iOS

(+) [iOS Appium Tests passed \(31m\)](#) 11:15

Комментировать задачу

Показать комментарии

Назначить

Изменить статус

Быстрый просмотр

Меню

@thomasAndersonBot

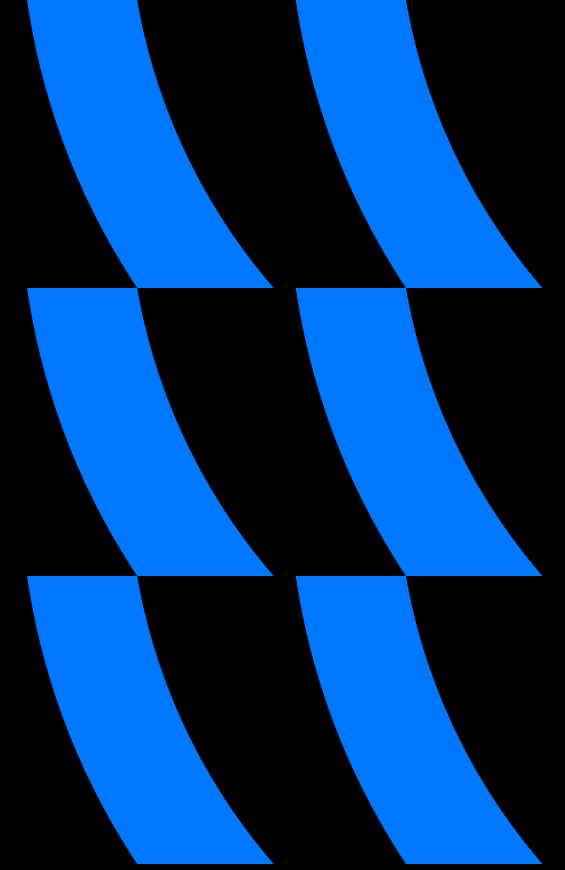
[TASK-9951](#)

Выкатить новый релиз для Target Android

(-) [Android Target release failed \(4m\)](#) 14:07

14:07

Что делать если бот  
выполняет команду с  
ошибкой?



# Почему нужны качественные логи?

▼  Томас Андерсон добавил(а) комментарий - 2020-04-16 11:44

Something went wrong

[Reply](#)

# Почему нужны качественные логи?

▼  Томас Андерсон добавил(а) комментарий - 2020-04-16 11:44

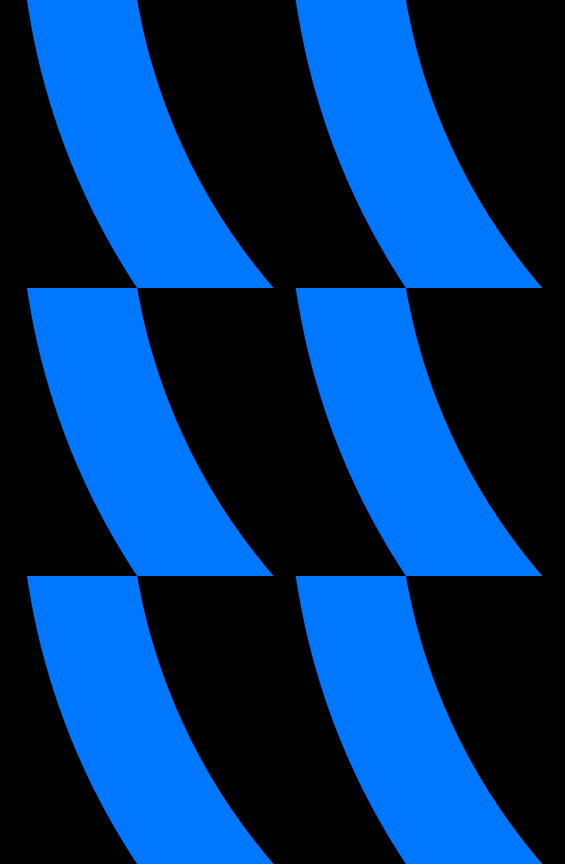
Something went wrong

Reply

Уйдет до 1-го дня на решение проблемы

## Почему нужны качественные логи?

[FATAL][git.rb] Can't checkout branch,  
it may not exist



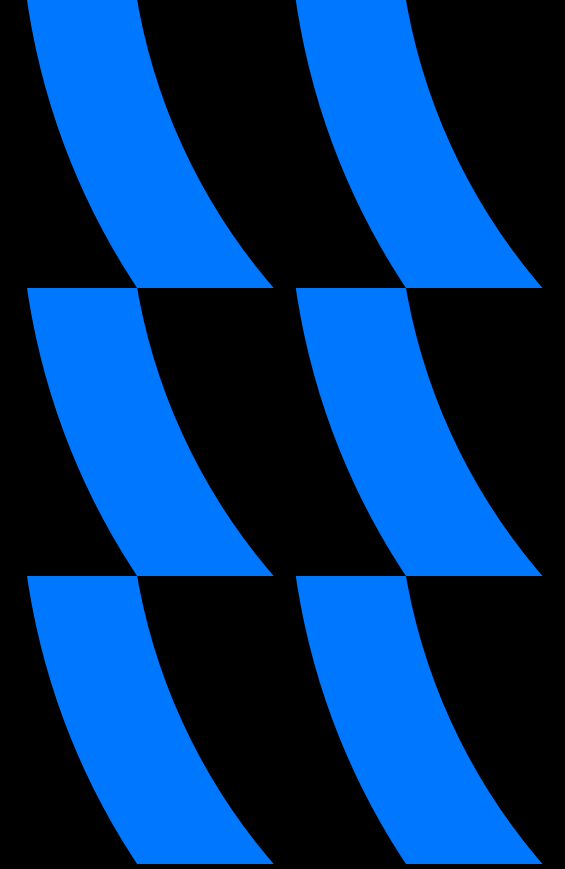
## Почему нужны качественные логи?

[FATAL][git.rb] Can't checkout branch,  
it may not exist

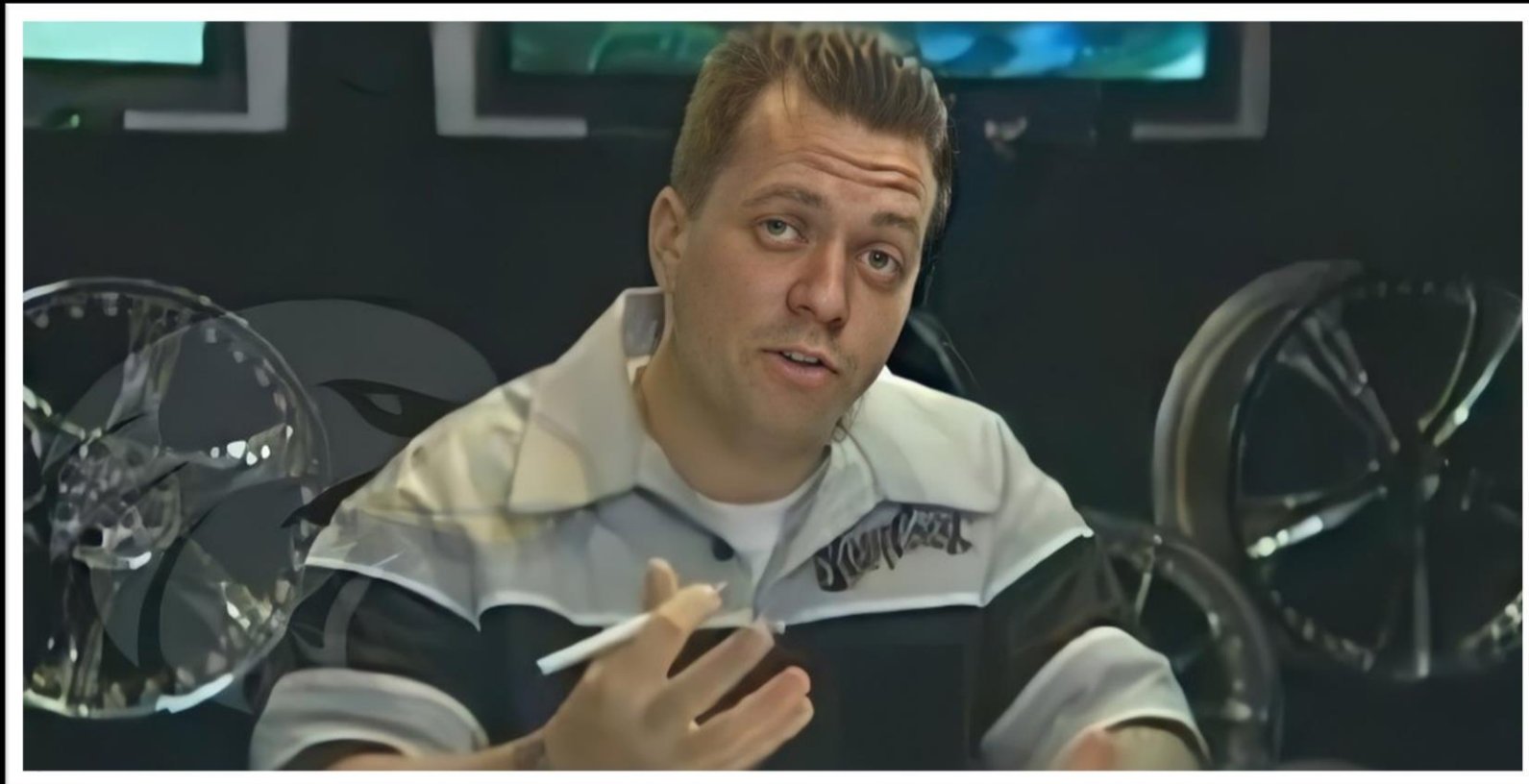


5 мин на нахождение проблемы

10 мин на решение и проверку, что все ок



# Мониторинг на CI/CD





Что нужно учесть в логах

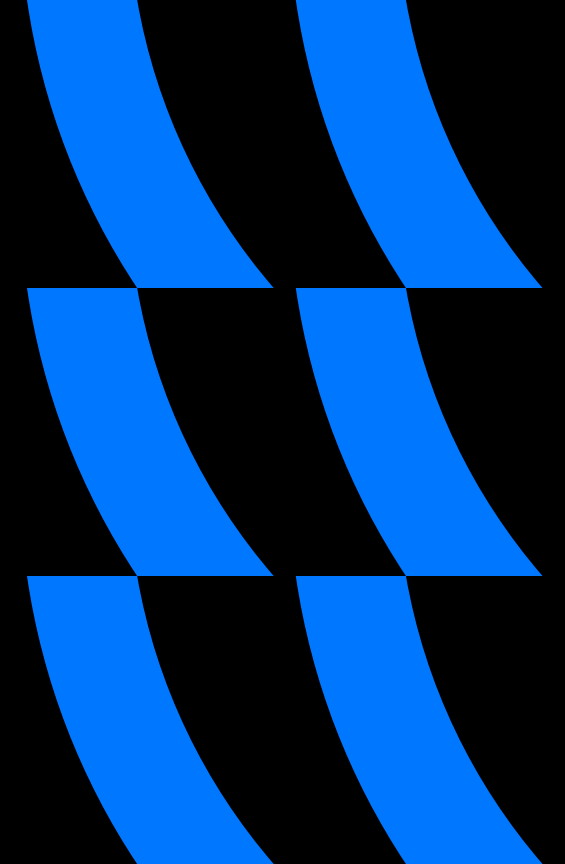
Сообщения должны быть читабельными



## Что нужно учесть в логах

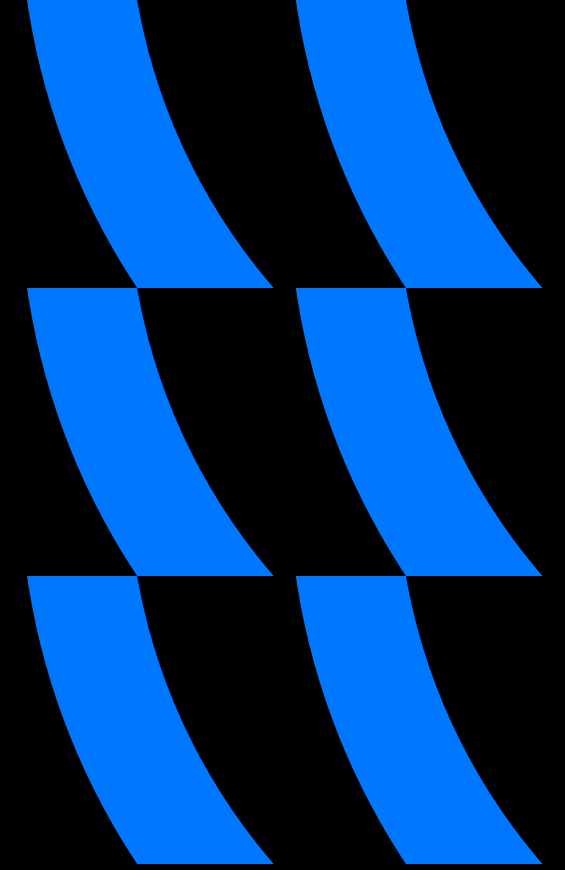
Сообщения должны быть читабельными

Логи должны указывать на проблему, и показывать, что к этому привело



## Способы логирования

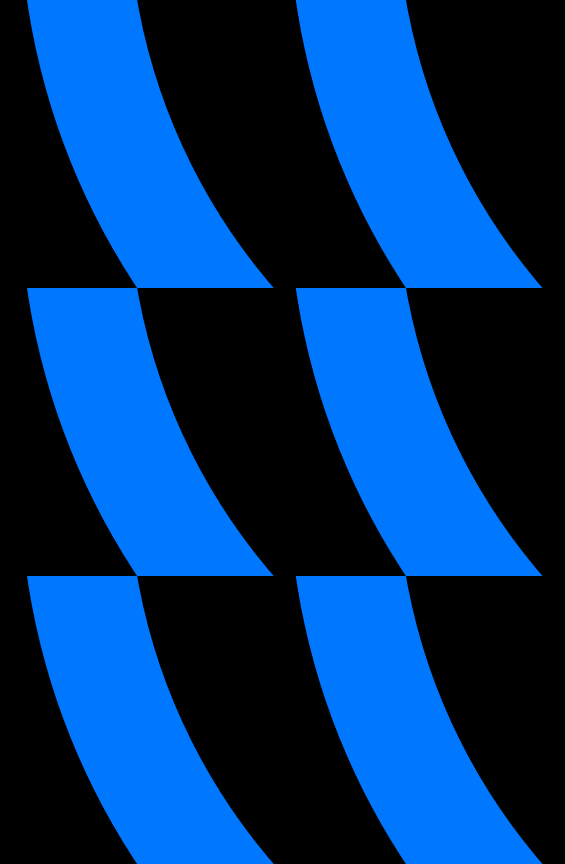
Писать логи во всех методах



## Способы логирования

Писать логи во всех методах

Использовать декораторы



# Уровни логирования на примере Merge ветки Target в Master

```
[INFO] [pid:94068] [lib/autobot/modules/gitlab/gitlab.rb:64] [merge] Target branch: master
[INFO] [pid:94068] [lib/autobot/modules/gitlab/gitlab.rb:65] [merge] Stage 1: merge target branch into source branch
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:48] [checkout] Checkout to origin/target
[INFO] [pid:94068] [lib/autobot/common/git/git.rb:60] [branch_exists] Check branch existing in origin
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:62] [branch_exists] 'git show-branch origin/target'
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:54] [checkout] 'git checkout origin/target'
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:101] [merge] merge branch origin/master
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:102] [merge] 'git merge origin/master'
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:66] [diff] Diff of branch with grep
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:71] [diff] 'git diff --name-only --diff-filter=U '
[DEBUG] [pid:94068] [lib/autobot/common/checks/checks.rb:7] [check_merge_conflict?] Checking merge conflict status
[INFO] [pid:94068] [lib/autobot/modules/gitlab/gitlab.rb:74] [merge] Stage 2: merge source branch into target branch
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:48] [checkout] Checkout to master
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:54] [checkout] 'git checkout master'
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:101] [merge] merge branch origin/target
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:102] [merge] 'git merge origin/target'
[DEBUG] [pid:94068] [lib/autobot/common/git/git.rb:116] [push] Pushing to origin
[EXPLAIN] [pid:94068] [lib/autobot/common/git/git.rb:117] [push] 'git push --all origin'
```

# Команды бота для CI/CD пайплайна

```
jirahelper merge --from master --to TASK-4712
```

Мержим одну ветку в другую одной командой

```
jirahelper comment --task TASK-4712
```

```
--message "Unit Success"
```

Пишем комментарии в задачу или в месенджер

```
jirahelper dev --task TASK-4712 --human
```

```
"vasya_pupkin@vk.com"
```

Устанавливаем исполнителя в задачу

```
jirahelper attach --task TASK-3141
```

```
--file "file path"
```

Прикрепляем файл к задаче

# Загружаем iOS SDK в Cocoapods

```
if("${UPLOAD_TO_COCOPODS}" == "true") {  
  stage("Push to Cocoapods") {  
    labelledShell(label: 'Загружаем SDK в Cocoapods', script: '''  
    jirahelper target_ios deploy publish_to_cocoapods --sdk_version $SDK_VERSION  
    ''')  
    sendJiraMessage(taskId: issue_id, message: "SDK загружен в Cocoapods")  
  }  
}
```

# Взаимодействуем с системой Git

```
if ("${READY TO RELEASE}" == "true") {
  stage("Postcondition") {
    labelledShell(label: 'Мержим релизную ветку в продакшен', script: '''
jirahelper target_ios gitlab merge --branch $DEV_BRANCH --target production
''')

    labelledShell(label: 'Устанавливаем релизный тэг', script: '''
jirahelper target_ios gitlab create_tag --tag $SDK_VERSION --branch $DEV_BRANCH --notes "$NEW_DOCS"
''')

    labelledShell(label: 'Мержим продакшн ветку в мастер', script: '''
jirahelper target_ios gitlab merge --branch production --target master
''')

    labelledShell(label: 'Удаляем релизную ветку', script: '''
jirahelper target_ios gitlab remove_branches --branch $DEV_BRANCH
''')

    setJiraStatus(taskId: issue_id, jiraStatus: JiraStatus.RELEASED)
  }
}
```



# Уведомляем по почте и в мессенджер

```
if ("${SEND_RELEASE_MAIL}" == "true") {
  stage("Send success report") {
    labelledShell(label: 'Отправляем уведомления на почту и в мессенджер', script: '''
      jirahelper target_ios reporter send_release_mail --sdk_version $SDK_VERSION --notes "$NEW_DOCS" --trace
      jirahelper target_ios reporter send_myteam_release --tag $SDK_VERSION --notes "$NEW_DOCS"
    ''')

    link = "${TARGET_IOS_GITLAB_LINK}/tags/$SDK_VERSION"
    message = "[myTargetSDK v$SDK_VERSION|$link] успешно зарелизен"
    sendJiraMessage(taskId: issue_id, message: message)
    setJiraStatus(taskId: issue_id, jiraStatus: JiraStatus.RELEASED)
  }
}
```

А что еще можно автоматизировать через бота?

# А что еще можно автоматизировать через бота?

## Релиз

- Загрузка в различные репозитории
- Cocoapods, Adman, Maven, JCentre

# А что еще можно автоматизировать через бота?

## Релиз

- Загрузка в различные репозитории
- Cocoapods, Adman, Maven, JCentre

## Уведомления

- На почту
- В мессенджер
- В задачу

# А что еще можно автоматизировать через бота?

## Релиз

- Загрузка в различные репозитории
- Cocoapods, Adman, Maven, JCentre

## GitLab

- Работа с ветками
- Выявление измененных модулей, чтобы запускать только нужные тесты
- Обновление документации

## Уведомления

- На почту
- В мессенджер
- В задачу

# А что еще можно автоматизировать через бота?

## GitHub

- Обновление changelog
- Загрузка публичных демоков
- Создание тегов и веток

## Релиз

- Загрузка в различные репозитории
- Cocoapods, Adman, Maven, JCentre

## GitLab

- Работа с ветками
- Выявление измененных модулей, чтобы запускать только нужные тесты
- Обновление документации

## Уведомления

- На почту
- В мессенджер
- В задачу

# 4 основные проблемы из-за человеческого фактора

1

Исполнитель задачи в Jira забывает менять статус и комментировать прогресс

2

Путаница в параметрах при ручном запуске джобы на CI/CD

3

Не понятно какие действия запускались на CI/CD для задачи

4

В релизных процессах слишком много кода



# ИТОГИ



Бизнес процессы автоматизированы  
Исключили ошибки, возникающие из-за  
человеческого фактора





# ИТОГИ

1 Бизнес процессы автоматизированы  
Исключили ошибки, возникающие из-за человеческого фактора

2 Выполнения CI/CD процессов  
логируются в задачу, параметры  
подставляются автоматически  
Любой человек может посмотреть  
статус задачи и текущие  
выполненные действия

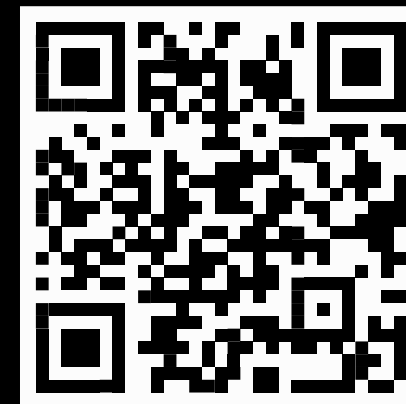
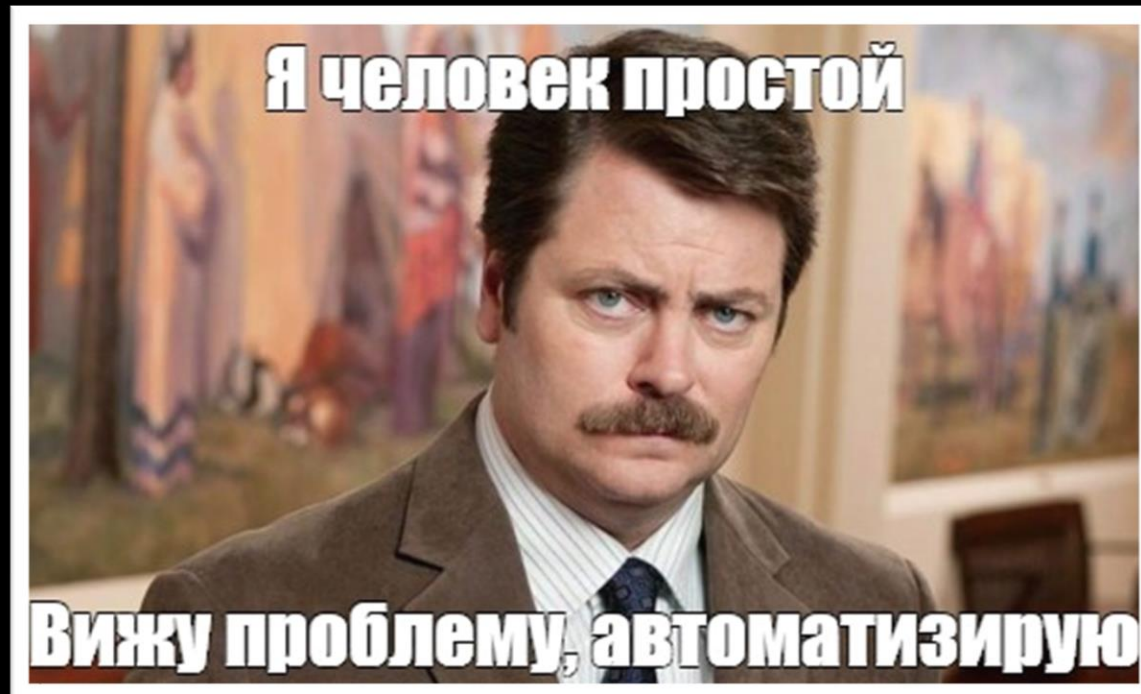


# ИТОГИ



- 1 Бизнес процессы автоматизированы  
Исключили ошибки, возникающие из-за человеческого фактора  
  
Выполнения CI/CD процессов логируются в задачу, параметры подставляются автоматически
- 2 Любой человек может посмотреть статус задачи и текущие выполненные действия
- 3 Релизные процессы вынесены в бота  
Сократили код на CI/CD

Спасибо за внимание!



Написать мне



Пример кода на  
Python



Пендрак Олег | VK Реклама