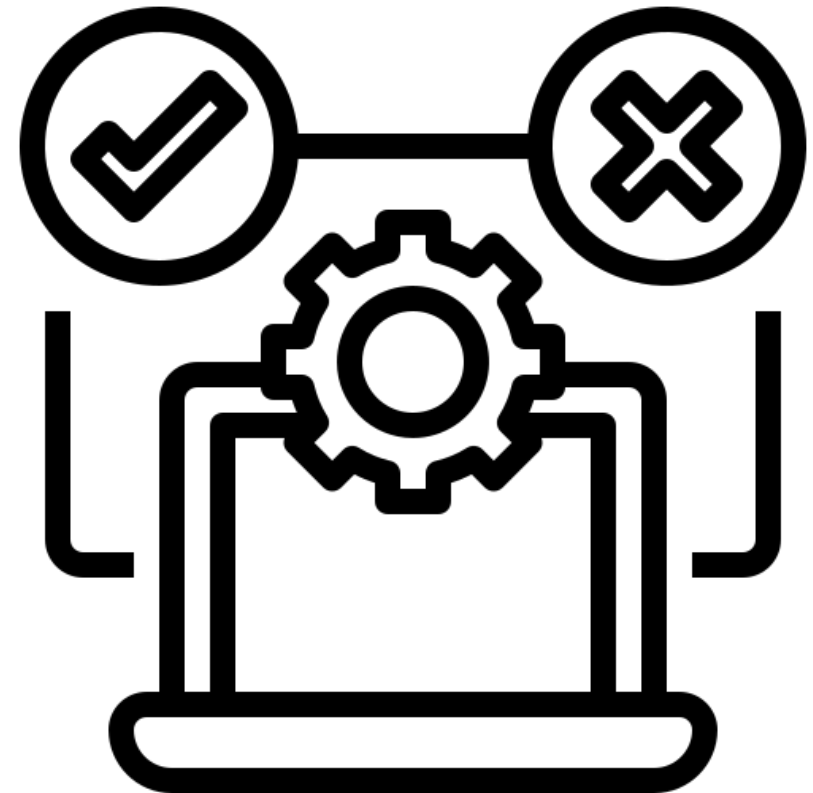


# АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ

Лекция 1: вводная

БАШАРИНА ЕКАТЕРИНА АЛЕКСАНДРОВНА



## ➤ **Введение в автоматизированное тестирование**

- **Общая информация. Что такое QA Automation?**
- **Ручное и автоматизированное тестирование**
- **Теория тестирования. Test case, issue, bug.**
- **Очередное знакомство с Java. Maven-проекты**
- **Система контроля версий Git**
- **Методика сдачи лабораторных работ**

# **Зачем нужно тестирование?**

- **Проверка соответствия требованиям**
- **Обнаружение проблем на ранних этапах разработки**
- **Взгляд на продукт со стороны пользователя**
- **Повышение лояльности к компании и продукту**

# Принципы тестирования

1. Тестирование демонстрирует наличие дефектов
2. Исчерпывающее тестирование невозможно
3. Раннее тестирование
4. Скопление дефектов
5. Парадокс пестицида
6. Тестирование зависит от контекста
7. Заблуждение об отсутствии ошибок



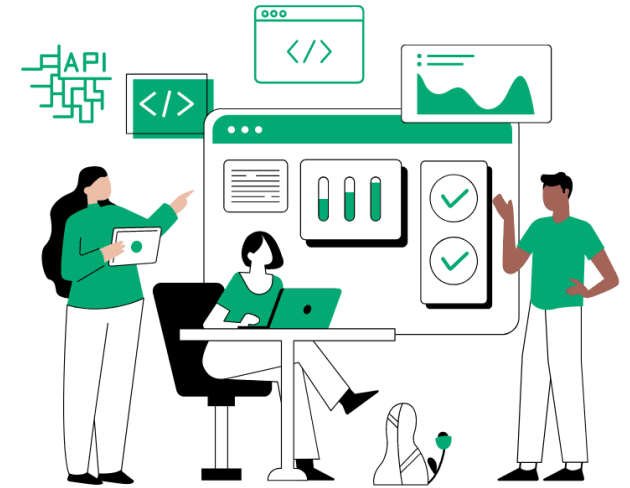
# Ручное тестирование

- Быстро и просто
- Низкий порог входа
- Не требует специальных программных средств
- Импровизация!



# Зачем автоматизировать?

- Убрать человеческий фактор
- Уменьшить трудозатраты
- Смоделировать большую нагрузку



Manual



Automated

# Автоматизированное тестирование: плюсы

- **Нагрузка на приложение**
- **Временной фактор**
- **Повторяемость**



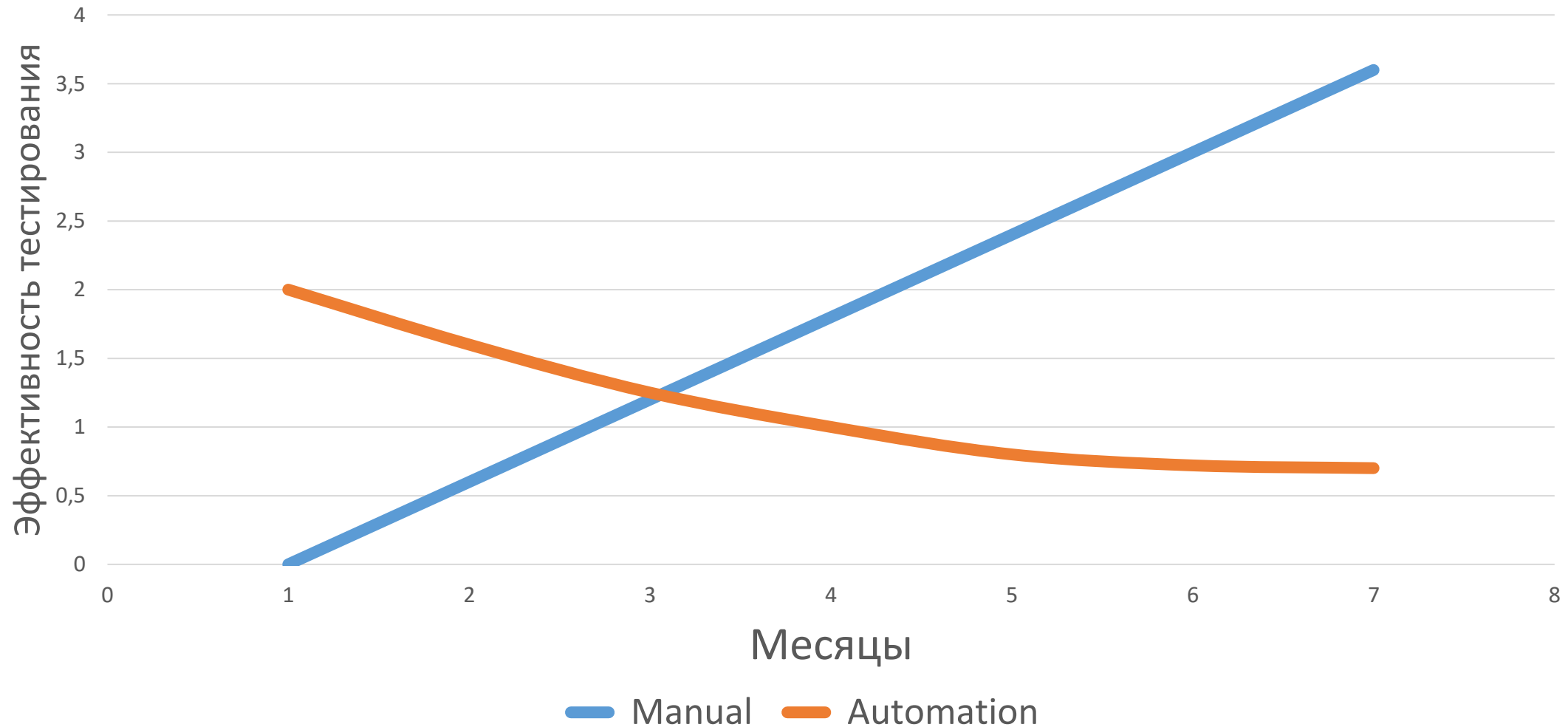
# Автоматизированное тестирование: минусы

- **Отсутствие тестирования глазами пользователя**
- **Отсутствие обратной связи**
- **Ненадежность**
- **Стоимость**





# Автоматизированное VS ручное



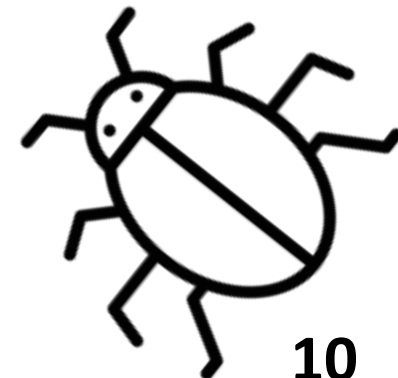
# Тестовый случай (test case)

- **Test case:**

**Action > Expected Result > Test Result**

**Позитивный тест кейс: передали  
«хорошее», ожидаем «правильное»**

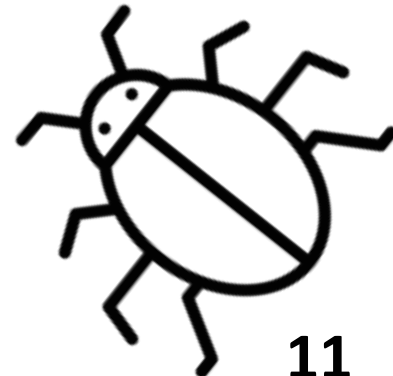
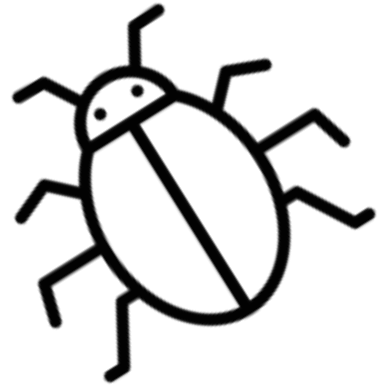
**Негативный тест кейс: передали  
«плохое», ожидаем «неправильное»**



# Issue и bug

- **Issue – то, что обнаружил тестировщик**
- **Bug – принятая «проблема»**

**Каждая issue мечтает стать bug'ом. Но не каждая становится...**



# Пример применения тестов

```
int result = Calc.sum(0, 0);  
Assert.areEquals(result, 0);
```

```
int result = Calc.sum(2, 2);  
Assert.areEquals(result, 4);
```

1. Open [www.epam.com](http://www.epam.com)
2. Fill login form
3. Press login button
4. Fill big form
5. Verify results
6. Search expected row
7. Open DB
8. Verify results in DB

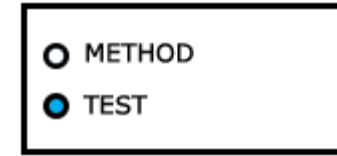
# Какие бывают тесты?

- **Модульные (Unit tests)**
- **Интеграционные (Service/API tests)**
- **Системные (System tests)**
- **Приёмочные (End-to-end tests)**

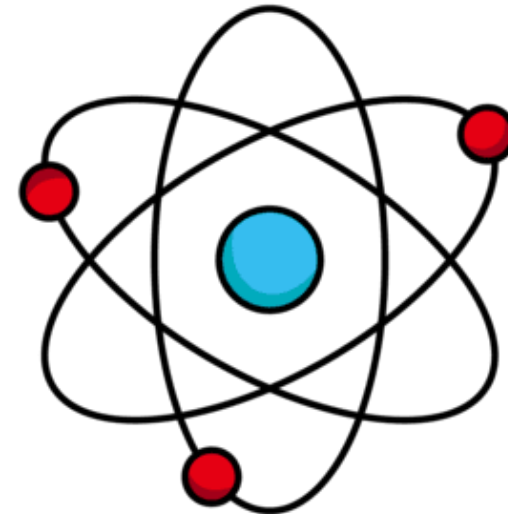
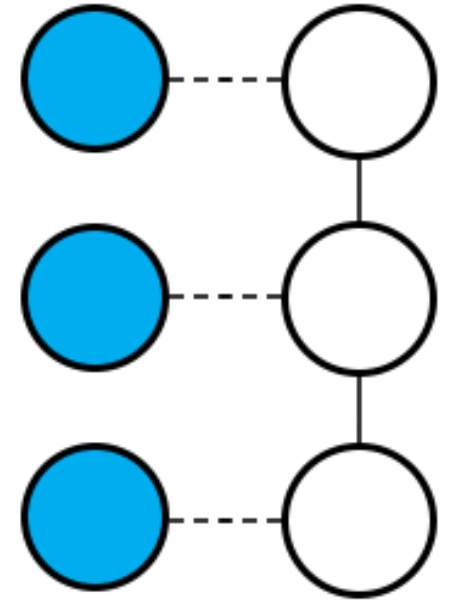


# Юнит тестирование

- Тестирование атомарных частей кода
- Большое количество тестов
- Всегда автоматизированы
- Независимы

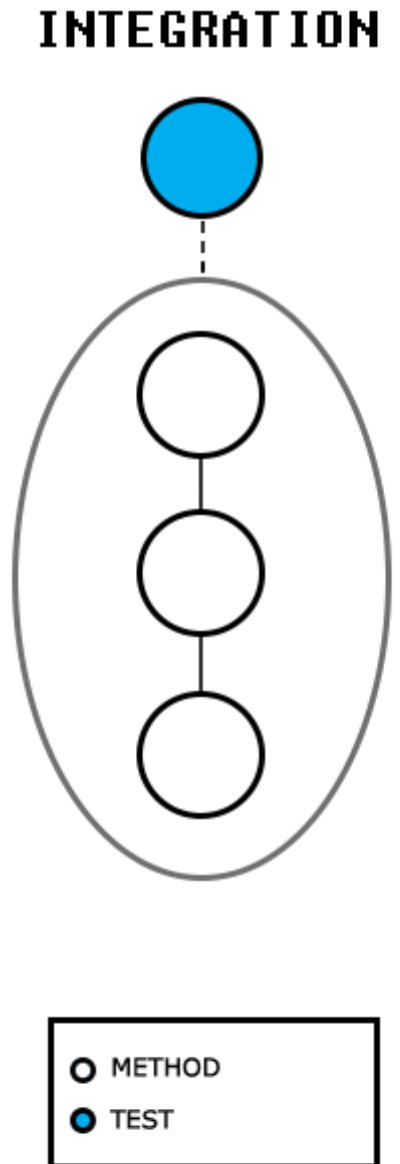


UNIT



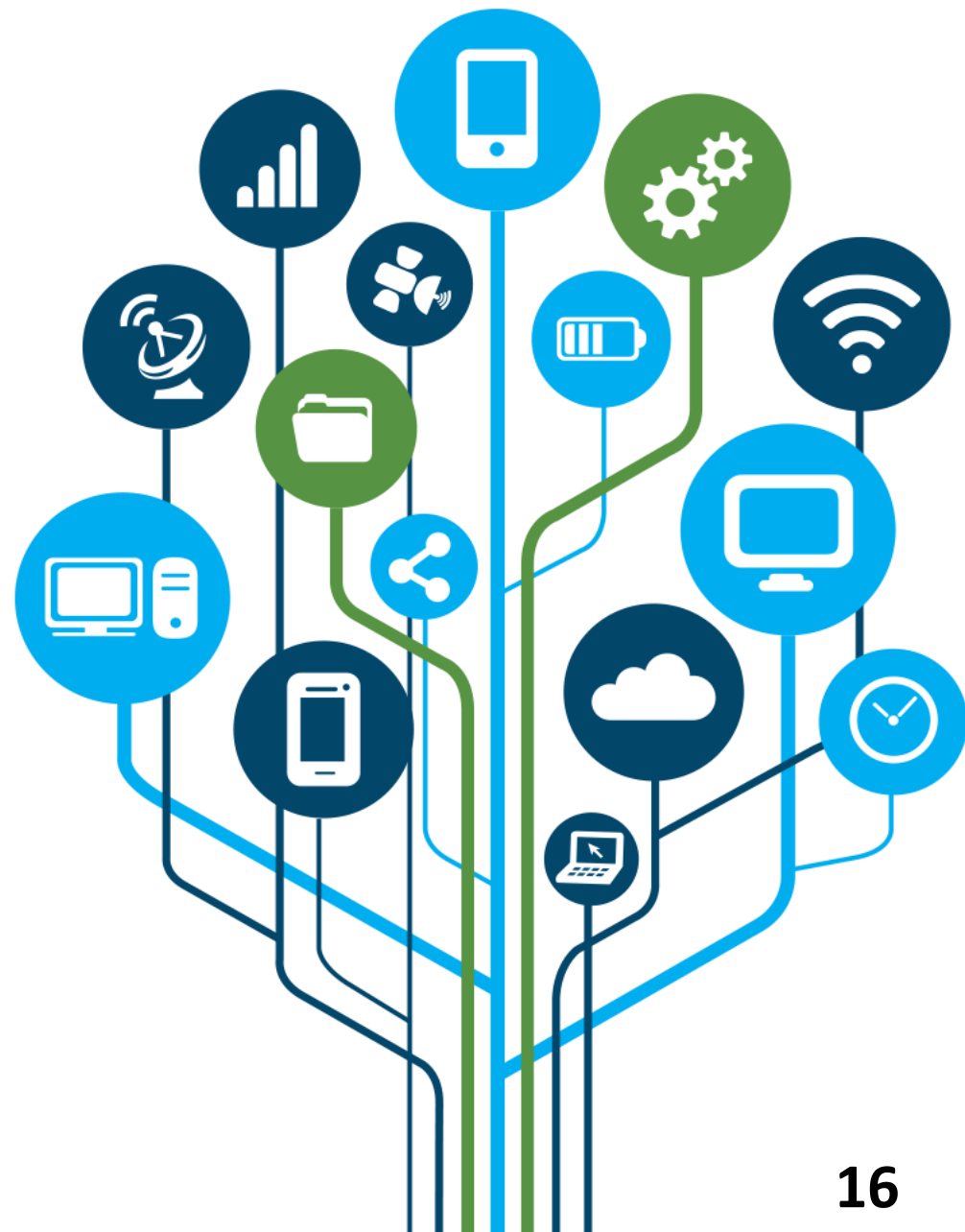
# Интеграционное тестирование

- Тестирование взаимосвязей компонент
- Проверка работы сервисов
- Можно тестировать сверху вниз, снизу вверх, либо по методу «большого взрыва»



# Системное тестирование

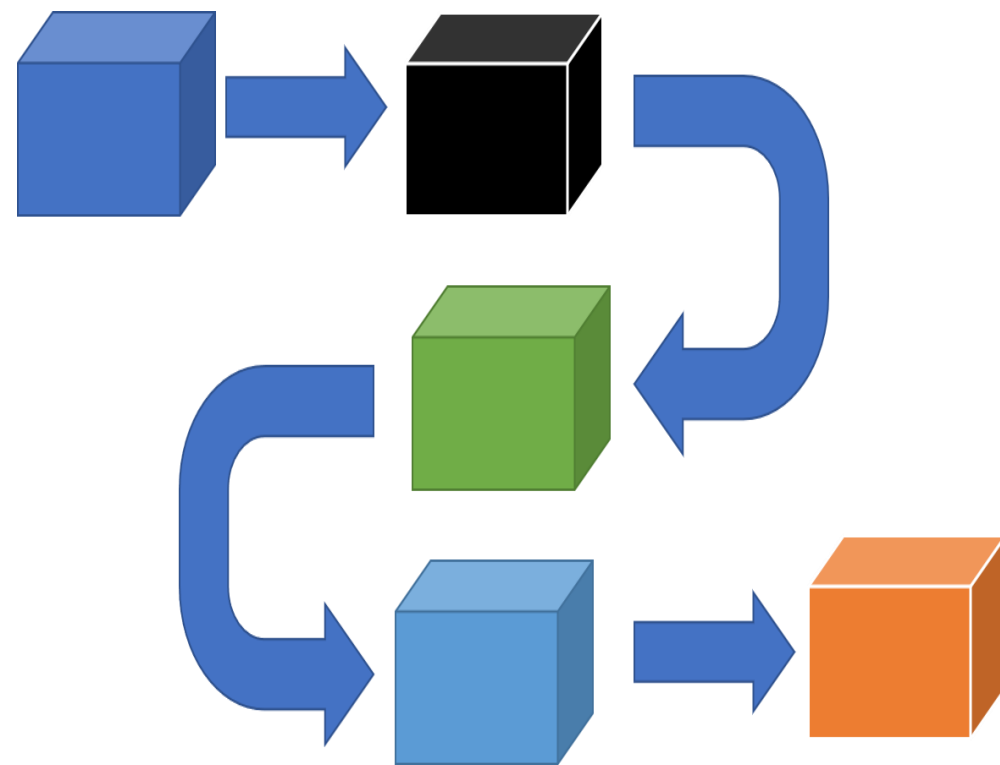
- Тестирование взаимодействия ПО с системой
- Окружение схоже с тем, что будет у пользователя
- Метод чёрного ящика



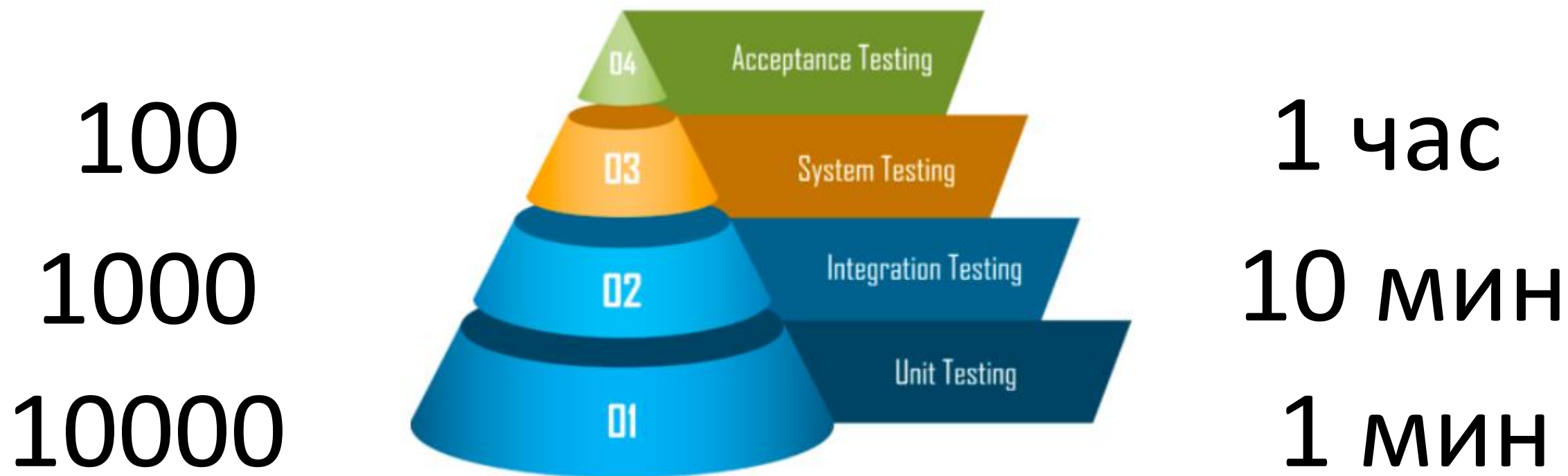


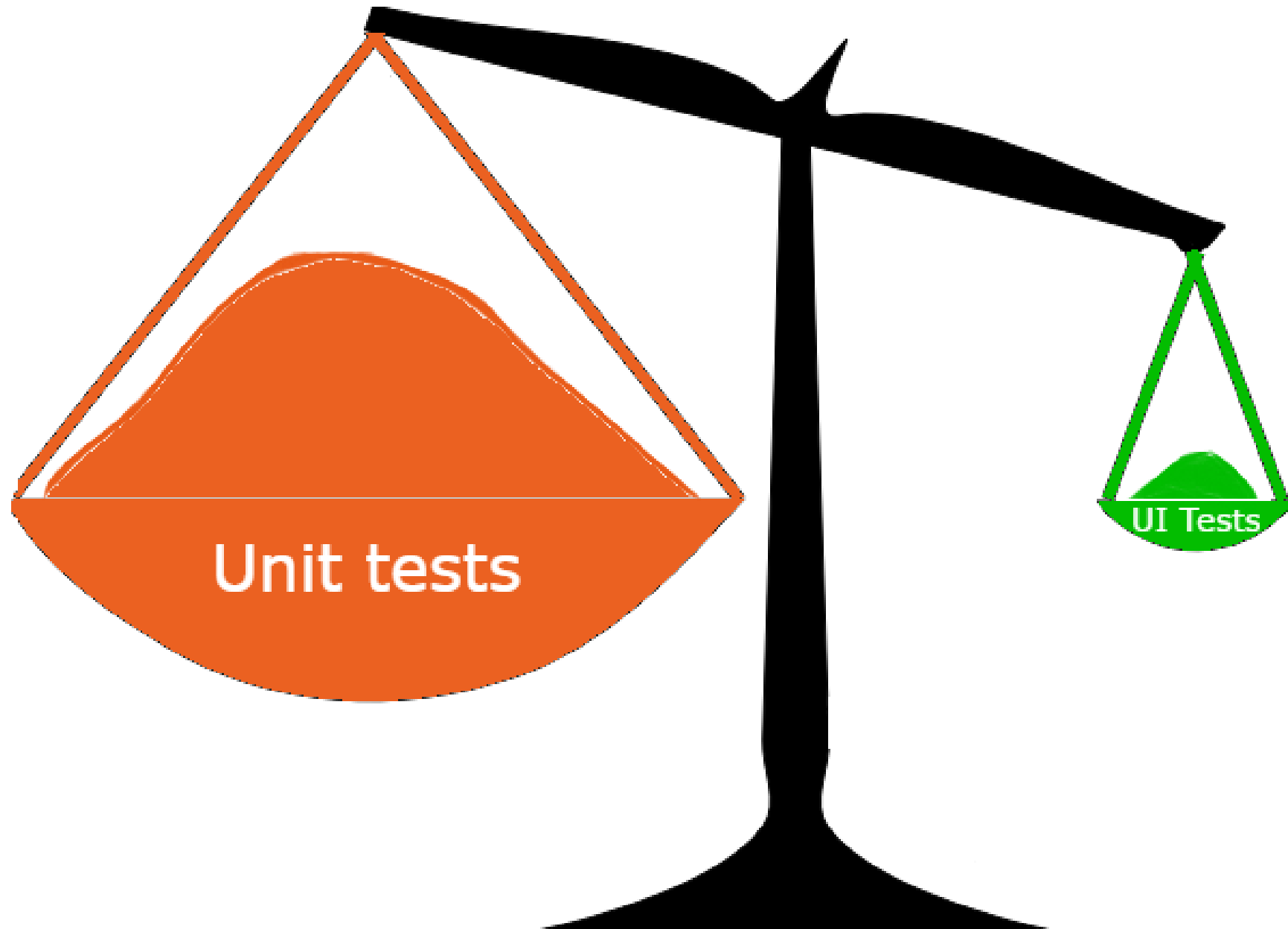
# Приемочное тестирование

- Валидация требований
- Только для ПО, достигшего необходимого качества
- Сложно автоматизировать
- Небольшое количество



# Пирамида тестирования





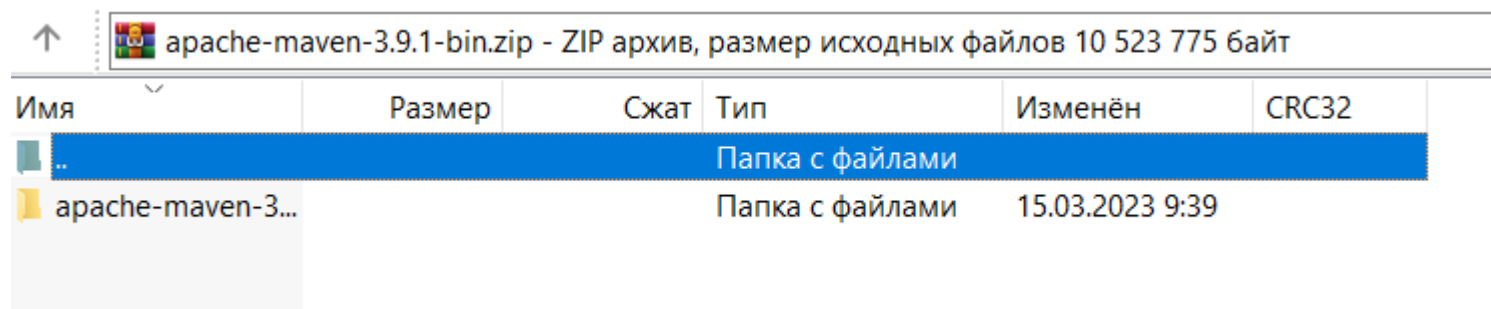
# Немного о Java и *Maven*<sup>™</sup>



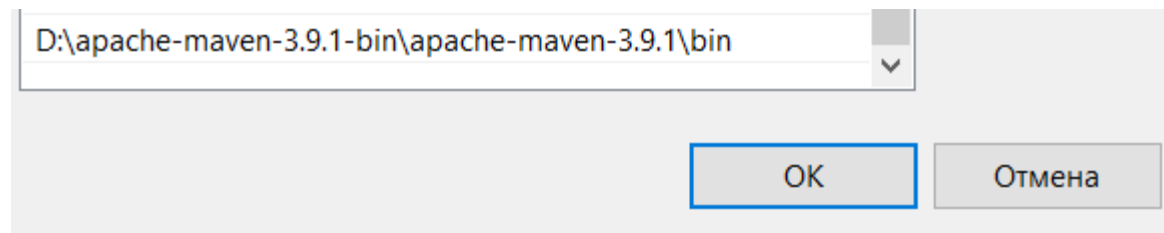
- **Сайт:** <https://maven.apache.org/index.html>
- **Откуда скачивать:** <https://maven.apache.org/download.cgi>
- **Инструкция по установке:** <https://maven.apache.org/install.html>
- **Краткий «гайд»:** <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

# Maven по шагам

- Скачать и распаковать архив



- Добавить путь к папке bin в Path



# Maven по шагам

- Создать папку для проекта и выполнить команду:

```
mvn archetype:generate
```

```
-DgroupId=org.example
```

```
-DartifactId=test-app
```

```
-DarchetypeArtifactId=maven-archetype-quickstart
```

```
-DarchetypeVersion=1.4
```

```
-DinteractiveMode=false
```

# Maven по шагам

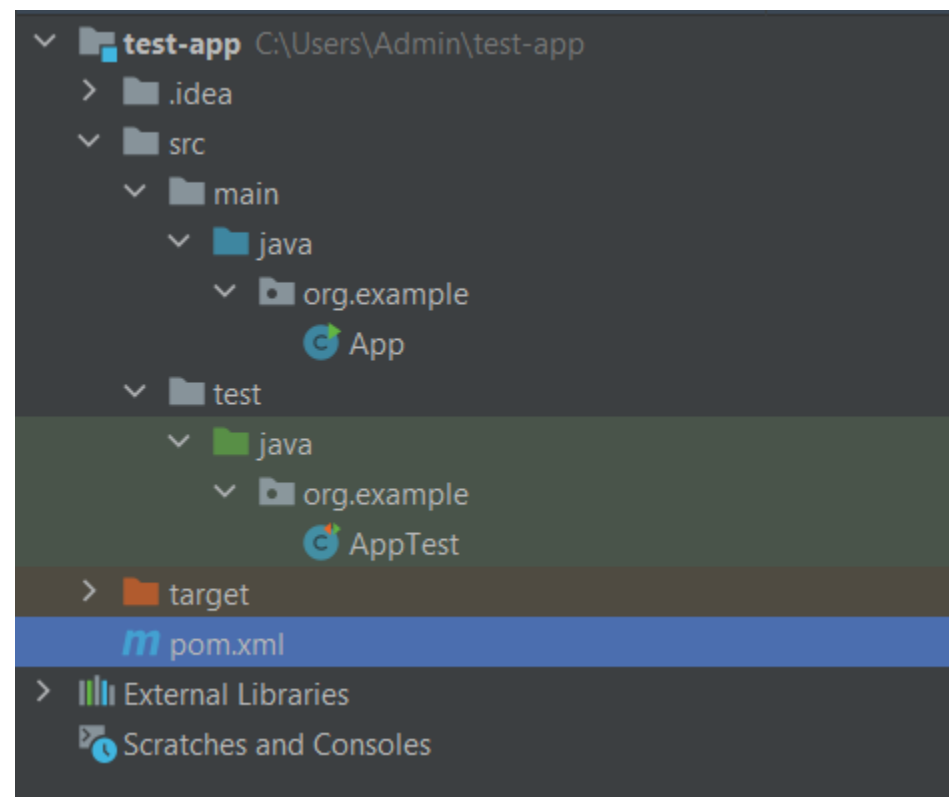
- В результате будет сгенерирован проект:

```
C:\Users\Admin\test-app>dir
Том в устройстве C не имеет метки.
Серийный номер тома: D6D7-4AE2

Содержимое папки C:\Users\Admin\test-app

27.03.2023  14:46    <DIR>        .
27.03.2023  14:46    <DIR>        ..
27.03.2023  14:46                2 737 pom.xml
27.03.2023  14:46    <DIR>        src
                1 файлов                2 737 байт
                3 папок  11 734 196 224 байт свободно

C:\Users\Admin\test-app>
```



# Maven по шагам

- В корне проекта выполнить команду `mvn package`

```
C:\Users\Admin\test-app>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.example:test-app >-----
[INFO] Building test-app -DinteractiveMode=false
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
```

- Получите информацию о сборке и прохождении тестов



# Maven по шагам

```
[INFO]
[INFO] -----
[INFO]  T E S T S
[INFO] -----
[INFO] Running org.example.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.062 s - in org.example.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.0.2:jar (default-jar) @ test-app ---
```

```
[INFO] Building jar: C:\Users\Admin\test-app\target\test-app--DinteractiveMode=false.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.817 s
[INFO] Finished at: 2023-03-27T14:49:11+03:00
[INFO] -----

C:\Users\Admin\test-app>
```

# Пара слов о Git и GitHub

- **Git – это система контроля версий**  
**Скачать и установить:** <https://git-scm.com/downloads>
- **GitHub – это веб-сервис для хранения репозиториев**  
**Создать аккаунт:** <https://github.com/>



# Как загрузить проект в GitHub?

- В папке проекта инициализировать репозиторий командой `git init`:

```
PS C:\Users\Admin\test-app> git init
Initialized empty Git repository in C:/Users/Admin/test-app/.git/
PS C:\Users\Admin\test-app> 
```

- Добавить необходимые файлы командой `git add`:

```
PS C:\Users\Admin\test-app> git add .
```

# Как загрузить проект в GitHub?

- При необходимости – создать новую ветку командой `git checkout -b branch-name`

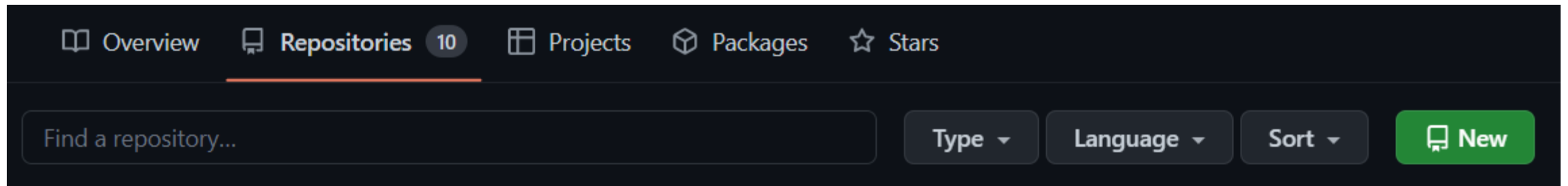
```
PS C:\Users\Admin\test-app> git checkout -b new-branch
Switched to a new branch 'new-branch'
PS C:\Users\Admin\test-app> 
```

- Сделать коммит командой `git commit -m "...":`

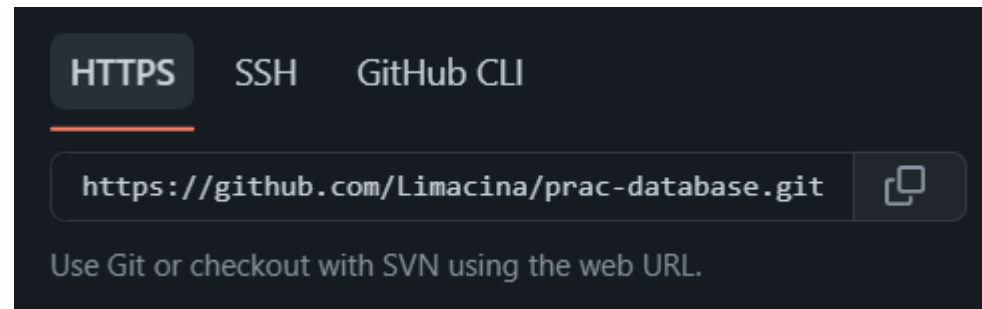
```
PS C:\Users\Admin\test-app> git commit -m "Add all files"
[master (root-commit) 54e6fb4] Add all files
22 files changed, 266 insertions(+)
```

# Как загрузить проект в GitHub?

- В разделе «Repositories» профиля GitHub нажать New



- Ввести данные репозитория, создать его и скопировать ссылку



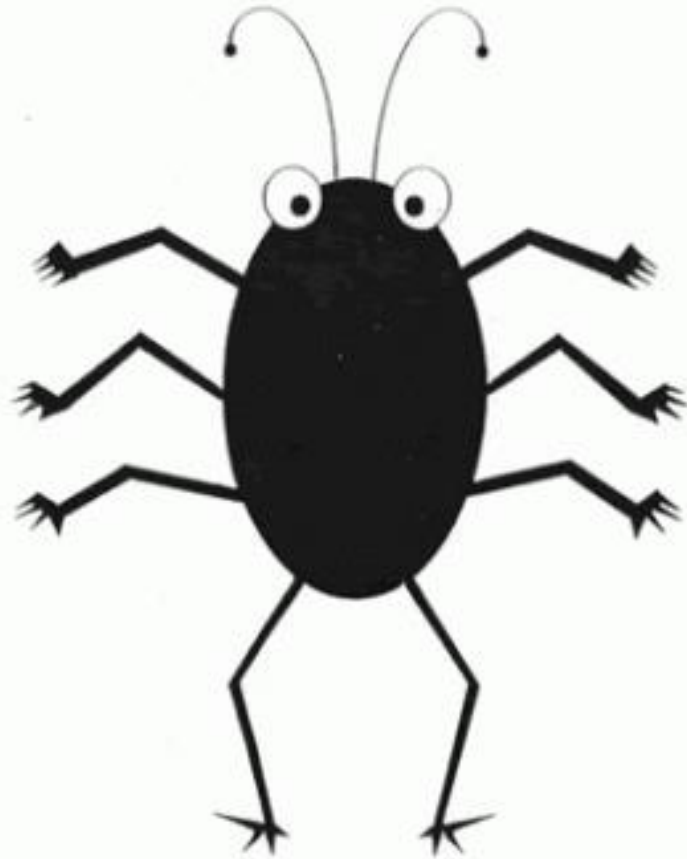
# Как загрузить проект в GitHub?

- Отправить изменения командой `git push -u <ссылка на репозиторий>`

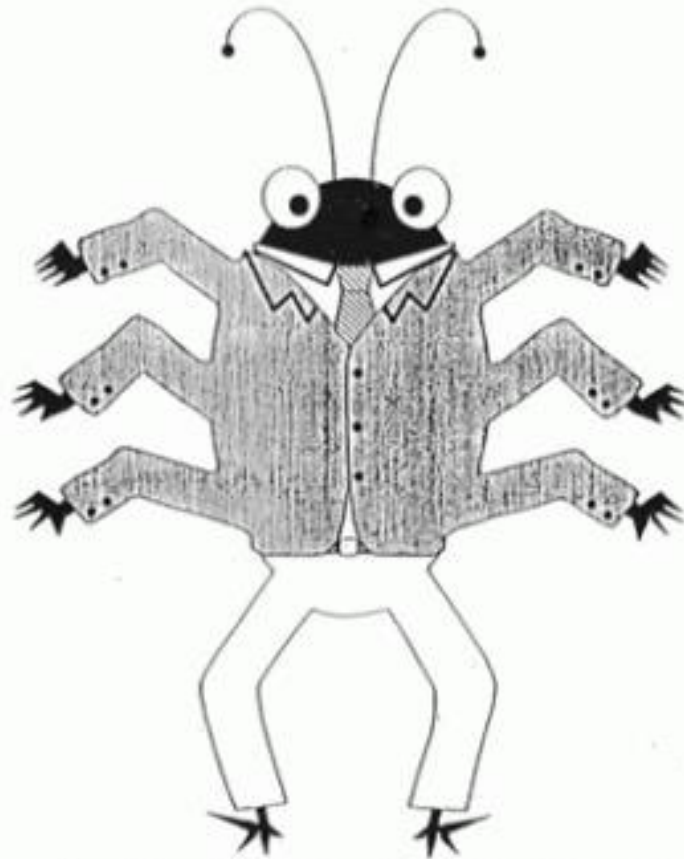
```
PS C:\Users\Admin\test-app> git push -u https://github.com/Limacina/test-repo.git
```

- Profit!

# Спасибо за внимание!



**BUG**



**FEATURE**