

Отчет по системному тестированию  
по дисциплине «Технология разработки качественного  
программного продукта»

Выполнили студенты гр. 3530904/80105:

Пинаев Н. Д.  
Распереза А. Д.

Руководитель:

Маслаков А. П.

Санкт-Петербург  
2022 г.

## Задание

На данном уровне необходимо протестировать готовый продукт по бизнес-требованиям, сформированным перед началом непосредственного проектирования программного продукта. В качестве тестовых сценариев выбираются основные сценарии использования ПО в полностью рабочем окружении. Целесообразно использование инструментов веб-тестирования или UI-тестирования, таких как selenium, puppeteer и т. д.

### **Обязательные требования:**

1. Предварительное формирование документа, описывающего тестовые сценарии (утверждение сценариев у преподавателя)
2. Поднятие сервера непрерывной интеграции и запуск задачи системного тестирования по временному триггеру или в ручном режиме (кроме тестирования производительности)

### **Отчёт по системному тестированию должен содержать:**

1. Отчёт о выполненной работе, использованных инструментах.
2. Тест-план со словесным описанием тестовых сценариев
3. Отчёт о прохождении тестов с результатами на сервере непрерывной интеграции.

## Ход работы

Для выполнения системных тестов был реализован полностью готовый продукт, инструмент для скачивания apk и ipa файлов из Google Play и App Store соответственно. Он имеет интерфейс в виде командной строки и предназначен для автоматизации различных вещей или построения CI/CD пайплайнов, например для проверок на безопасность. Распространение инструмента было автоматизировано и идет в виде PyPi пакета и докер-образа в Docker Hub. Подразумевается, что конечный пользователь будет использовать именно такие варианты работы с Poly App Downloader'ом. В виде тестов были реализованы все основные e2e сценарии использования, а также было выполнено тестирование производительности по критериям:

- Load testing
- Stability testing

Об этом подробнее дальше в отчете

## Использованные инструменты

- Pytest
- Github actions
- PyPi
- Docker hub

В соответствии с заданием системные тесты выполняются ежедневно, для этого был создан отдельный пайплайн. Также, они запускаются при пуше в мастер.

Тесты имитируют работу обычного пользователя, который используя докер образ или питоновский пакет пытается скачать приложения. Запросы отправляются на Google Play API или на iTunes/App Store API в зависимости от типа системы дистрибуции (параметр `–distribution_system` `appstore/google_play`)

Модули:

- User Cli Interface
- PyPi package
- Docker Hub image
- Google Play API
- AppStore

## Тест план

№	Сценарий	Ожидаемое поведение	Тестируемые модули	Шаги
1	Test_install_pypi_package (установка pypi репозитория)	Пользователь успешно получает последнюю версию пакета	PyPi User CLI interface	1. Отправить команду на скачивание последней версии пакета 2. Проверить, что пакет скачан успешно, последняя версия установлена
2	Test_download_package (скачивание докер-образа)	Пользователь успешно получает докер образ	Docker User CLI interface	1. Отправить команду на скачивание последней версии докер-образа 2. Проверить, что образ скачан успешно, последняя версия установлена
3	Test_pypi_download_from_google_play (скачивание приложения из гугл плэя через pypi)	Пользователь успешно скачивает приложение, в логах отображаются ожидаемые значения	PyPi User CLI interface Google Play API	1. Получить креды из конфига 2. Получить валидный package name 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и package name 4. Проверить, что приложение скачено успешно, exit code = 0, в логах отображена информация о процессе.
4	Test_docker_download_from_google_play (скачивание приложения из гугл плэя через docker)	Пользователь успешно скачивает приложение, в логах	Docker Hub User CLI interface Google Play API	1. Получить креды из конфига 2. Получить валидный package name 3. Отправить команду на скачивание приложения через

		отображаются ожидаемые значения		docker с полученными кредитами и package name 4. Проверить, что приложение скачено успешно, в логах отображена информация о процессе.
5	Test_pypi_download_paid_app_from_google_play (попытка скачивания платного приложения из гугл плэя через pypi)	Пользователь не сможет установить платное приложение, в логах будет отображен код ошибки = 4	PyPi User CLI interface Google Play API	1. Получить кредиты из конфига 2. Получить валидный package name 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и package name 4. Проверить, что скачивание приложения не удалось, exit code = 4, в логах отображена информация о процессе.
6	Test_docker_download_paid_app_from_google_play (попытка скачивания платного приложения из гугл плэя через docker))	Пользователь не сможет установить платное приложение, в логах будет отображен код ошибки = 4	Docker Hub User CLI interface Google Play API	1. Получить кредиты из конфига 2. Получить валидный package name 3. Отправить команду на скачивание приложения через docker с полученными кредитами и package name 4. Проверить, что скачивание приложения не удалось, exit code = 4, в логах отображена информация о процессе.
7	Test_pypi_download_banned_app_from_google_play (попытка скачивания забаненного приложения из гугл плэя через pypi)	Пользователь не сможет установить забаненное приложение, в логах будет отображен код ошибки = 4	PyPi User CLI interface Google Play API	1. Получить кредиты из конфига 2. Получить невалидный package name 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и package name 4. Проверить, что скачивание приложения не удалось, exit code = 4, в логах отображена информация о процессе.
8	Test_docker_download_banned_app_from_google_play	Пользователь не сможет установить	Docker Hub User CLI interface	1. Получить кредиты из конфига 2. Получить невалидный package name

	(попытка скачивания забаненного приложения из гугл плей через docker)	забаненное приложение, в логах будет отображен код ошибки = 4	Google Play API	3. Отправить команду на скачивание приложения через docker с полученными кредитами и package name 4. Проверить, что скачивание приложения не удалось, exit code = 4, в логах отображена информация о процессе.
9	test_pypi_lookup_info_about_app_by_bundle_from_app_store (поиск информации о приложении через бандл в эпсторе, через pypi)	Пользователь успешно получает информацию о приложении, в логах отображаются ожидаемые значения	PyPi User CLI interface AppStore	1. Получить кредиты из конфига 2. Получить валидный bundle id 3. Отправить команду на получение информации о приложении через pypi пакет с полученными кредитами и bundle id 4. Проверить, что информация получена успешно, exit code = 0, в логах отображена информация о процессе.
10	test_docker_lookup_info_about_app_by_bundle_from_app_store (поиск информации о приложении через бандл в эпсторе, через docker)	Пользователь успешно получает информацию о приложении, в логах отображаются ожидаемые значения	Docker Hub User CLI interface AppStore	1. Получить кредиты из конфига 2. Получить валидный bundle id 3. Отправить команду на получение информации о приложении через docker с полученными кредитами и bundle id 4. Проверить, что информация получена успешно, в логах отображена информация о процессе.
11	test_pypi_download_app_by_bundle_from_app_store (скачивание приложения из эпстор по бандлу через pypi)	Пользователь успешно скачивает приложение, в логах отображаются ожидаемые значения	PyPi User CLI interface AppStore	1. Получить кредиты из конфига 2. Получить валидный bundle id 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и bundle id 4. Проверить, что приложение скачано успешно, exit code = 0, в логах отображена информация о процессе.
12	test_docker_download_app_by_bundle_from_app_store	Пользователь успешно	Docker Hub	1. Получить кредиты из конфига 2. Получить валидный bundle id

	re (скачивание приложения из эпстор по бандлу через docker)	скачивает приложение, в логах отображаются ожидаемые значения	User CLI interface  AppStore	3. Отправить команду на скачивание приложения через docker с полученными кредитами и bundle id 4. Проверить, что приложение скачано успешно, в логах отображена информация о процессе.
13	test_pypi_download_app_by_id_from_app_store (скачивание приложения из эпстор по app_id через pypi)	Пользователь успешно скачивает приложение, в логах отображаются ожидаемые значения	PyPi  User CLI interface  AppStore	1. Получить кредиты из конфига 2. Получить валидный app id 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и app id 4. Проверить, что приложение скачано успешно, exit code = 0, в логах отображена информация о процессе.
14	test_pypi_download_app_by_id_from_app_store (скачивание приложения из эпстор по app_id через docker)	Пользователь успешно скачивает приложение, в логах отображаются ожидаемые значения	Docker Hub  User CLI interface  AppStore	1. Получить кредиты из конфига 2. Получить валидный app id 3. Отправить команду на скачивание приложения через docker с полученными кредитами и app id 4. Проверить, что приложение скачано успешно, в логах отображена информация о процессе.
15	test_pypi_download_paid_app_from_app_store (попытка скачивания платного приложения из эпстор по app_id через pypi)	Пользователь не сможет установить платное приложение, в логах будет отображен код ошибки = 1	PyPi  User CLI interface  AppStore	1. Получить кредиты из конфига 2. Получить валидный app id 3. Отправить команду на скачивание приложения через pypi пакет с полученными кредитами и app id 4. Проверить, что скачивание приложения не удалось, exit code = 1, в логах отображена информация о процессе.
16	test_pypi_download_paid_app_from_app_store (попытка скачивания платного приложения из	Пользователь не сможет установить платное	Docker Hub  User CLI interface	1. Получить кредиты из конфига 2. Получить валидный app id 3. Отправить команду на скачивание приложения через

	эпстор по app_id через ruri)	приложение, в логах будет отображен код ошибки = 1	AppStore	docker с полученными кредитами и app id 4. Проверить, что скачивание приложения не удалось, exit code = 1, в логах отображена информация о процессе.
17	test_stability_testing_for_a_ppstore_logins (тестирование стабильности для логинов в эпстор)	Многократный успешный поиск приложения по бандлу в течение минуты, все разы проходят успешно, без единой ошибки/неожиданного поведения	PyPi User CLI interface AppStore	1)Получить из конфига нужны кредиты и bundle_id 2)В течении минуты многократно повторять запрос на получение информации о приложении, каждый раз вызывая новый логин в App Store 3)Проверить что не было никаких исключений/ошибок логина
18	Test_load_testing_google_play_download (нагрузочное тестирование скачивания приложений из гугл плэя)	Успешное скачивание нескольких (10) объемных приложений, все разы проходят успешно, без единой ошибки/неожиданного поведения	PyPi User CLI interface Google Play API	1)Получить список из package name'ов для скачивания 2)Поочередно запускать команду для скачивания приложения через ruri пакет с помощью системы дистрибуции Google Play и имени пакета 3)Проверить что не было никаких исключений/ошибок и все приложения скачались

## Описание работы

Для запуска системных тестов был создан специальный ежедневный пайплайн, запускающий тесты каждый день:



main poly\_app\_downloader / .github / workflows / daily\_system\_tests.yml View runs Go to file ...

npinaev rename pipeline Latest commit f0193db 10 minutes ago History

1 contributor

30 lines (26 sloc) 863 Bytes Raw Blame

```
1 name: Run system tests everyday
2
3 on:
4   schedule:
5     - cron: "0 0 * * *"
6
7 jobs:
8
9   run-daily-system-tests:
10    runs-on: ubuntu-latest
11    strategy:
12      matrix:
13        python-version: ["3.6", "3.8", "3.9"]
14
15    steps:
16      - uses: actions/checkout@v3
17      - name: Set up Python ${ matrix.python-version }
18        uses: actions/setup-python@v3
19        with:
20          python-version: ${ matrix.python-version }
21      - name: Install dependencies
22        run: |
23          python -m pip install --upgrade pip
24          pip install flake8 pytest pytest_testconfig
25          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
26      - name: Test with pytest
27        run: |
28          pytest tests/system_tests --tc-file=config/creds.ini
29      env:
30        PYTHONPATH: ""
```

И также был добавлен шаг в обычном пайплайне:

```
run-system-tests:
  needs: run-integration-tests
  runs-on: ubuntu-latest
  strategy:
    matrix:
      python-version: [ "3.9" ]

  steps:
    - uses: actions/checkout@v3
    - name: Set up Python ${ matrix.python-version }
      uses: actions/setup-python@v3
      with:
        python-version: ${ matrix.python-version }
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install flake8 pytest pytest_testconfig
        if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
    - name: Test with pytest
      run: |
        pytest tests/system_tests -vv --tc-file=config/creds.ini
  env:
    PYTHONPATH: ""
```

В ежедневном пайплайне есть матрица, которая прогоняет тесты на разных версиях питона.

Давайте чуть подробнее остановимся на тестах для тестирования стабильности и нагрузочного тестирования:

- Тестирование стабильности

```
def test_stability_testing_for_appstore_logins(capfd):
    apple_id = config['as']['apple_id']
    password2FA = config['as']['password2FA']
    bundle_id = config['as']['bundle_id']
    t_end = time.time() + 60
    while time.time() < t_end:
        lookup_info = subprocess.run(
            ['poly_app_downloader', '--distribution_system', 'appstore', '--appstore_apple_id',
            f'{apple_id}', '--appstore_password2FA', f'{password2FA}', '--appstore_bundle_id',
            f'{bundle_id}', '--lookup'])
        out, err = capfd.readouterr()
        assert lookup_info.returncode == 0
        assert "INFO Successfully logged in" in out
        assert "INFO Successfully found application by bundle id (ru.gazprombank.ios.mobilebank) with name:" in out
        assert "app_id: 1406492297" in out
```

Проверка что в течение минуты будет много успешных попыток логина и состояние ни разу не изменится, не будет неуспешных попыток

- Нагрузочное тестирование

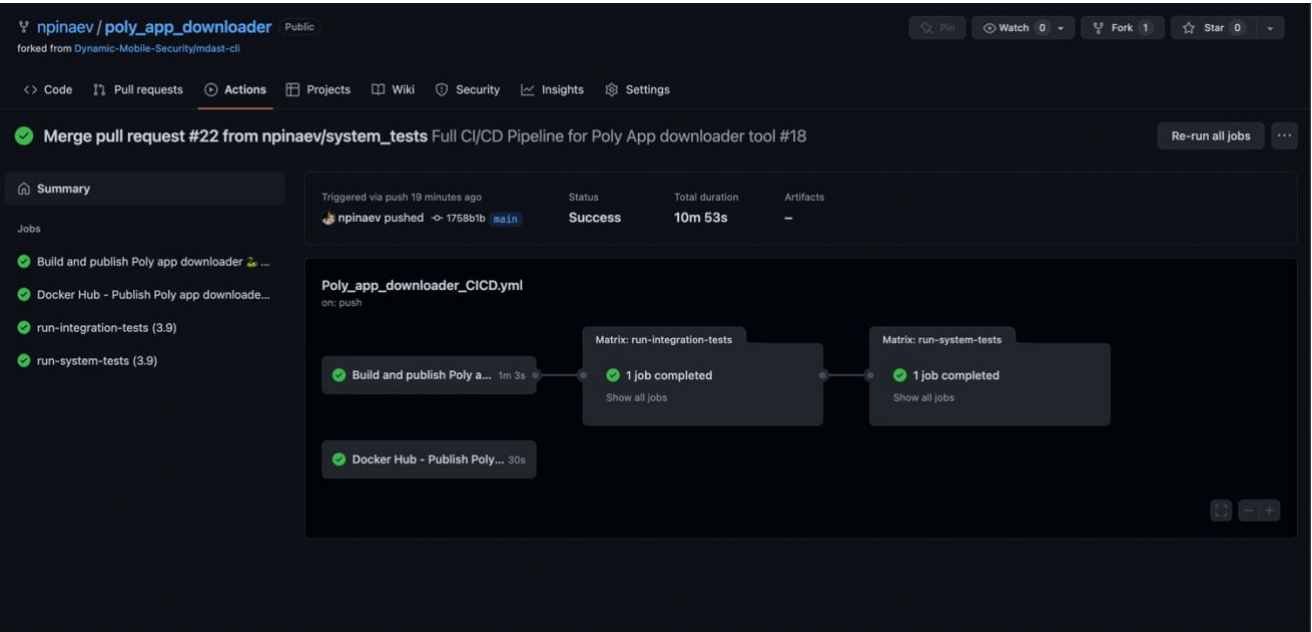
```
def test_load_testing_google_play_download(capfd):
    gsfid = int(config['gp']['gsfid'])
    authSubToken = config['gp']['authSubToken']
    for package_name in PACKAGE_NAMES:
        download_from_gp = subprocess.run(
            ['poly_app_downloader', '--distribution_system', 'google_play', '--google_play_package_name',
            f'{package_name}', '--google_play_gsfid', f'{gsfid}', '--google_play_auth_token',
            f'{authSubToken}'])
        out, err = capfd.readouterr()
        assert download_from_gp.returncode == 0
        assert "INFO Your application was downloaded!" in out
        assert "INFO Google Play - Logging in with gsfid and auth token" in out
        assert "INFO Google Play - Successfully logged in Play Store" in out
        assert f'{package_name}' in out
        assert "INFO Your application was downloaded!" in out
```

Проверка что приложения из списка, который подан на вход (в сумме >500mb)

были успешно скачаны поочередно и не возникло никаких ошибок.

Отчёт о прохождении тестов.

Скрин из общего пайплайна:



Детали прохождения:

Summary

Jobs

Build and publish Poly app downloader

Docker Hub - Publish Poly app downloade...

run-integration-tests (3.9)

run-system-tests (3.9)

run-system-tests (3.9)

succeeded 11 minutes ago in 9m 0s

Set up job

Run actions/checkout@v3

Set up Python 3.9

Install dependencies

Test with pytest

```
1 ▶ Run pytest tests/system_tests -vv --tc-file=config/creds.ini
8 ===== test session starts =====
9 platform linux -- Python 3.9.12, pytest-7.1.2, pluggy-1.0.0 -- /opt/hostedtoolcache/Python/3.9.12/x64/bin/python
10 cachedir: .pytest_cache
11 rootdir: /home/runner/work/poly_app_downloader/poly_app_downloader
12 plugins: testconfig-0.2.0
13 collecting ... collected 18 items
14
15 tests/system_tests/test_system.py::test_install_pypi_package PASSED [ 5%]
16 tests/system_tests/test_system.py::test_download_docker_image PASSED [ 11%]
17 tests/system_tests/test_system.py::test_pypi_download_from_google_play PASSED [ 16%]
18 tests/system_tests/test_system.py::test_docker_download_from_google_play PASSED [ 22%]
19 tests/system_tests/test_system.py::test_pypi_download_paid_app_from_google_play PASSED [ 27%]
20 tests/system_tests/test_system.py::test_docker_download_paid_app_from_google_play PASSED [ 33%]
21 tests/system_tests/test_system.py::test_pypi_download_banned_app_from_google_play PASSED [ 38%]
22 tests/system_tests/test_system.py::test_docker_download_banned_app_from_google_play PASSED [ 44%]
23 tests/system_tests/test_system.py::test_pypi_lookup_info_about_app_by_bundle_from_app_store PASSED [ 50%]
24 tests/system_tests/test_system.py::test_docker_lookup_info_about_app_by_bundle_from_app_store PASSED [ 55%]
25 tests/system_tests/test_system.py::test_pypi_download_app_by_bundle_from_app_store PASSED [ 61%]
26 tests/system_tests/test_system.py::test_docker_download_app_by_bundle_from_app_store PASSED [ 66%]
27 tests/system_tests/test_system.py::test_pypi_download_app_by_id_from_app_store PASSED [ 72%]
28 tests/system_tests/test_system.py::test_docker_download_app_by_id_from_app_store PASSED [ 77%]
29 tests/system_tests/test_system.py::test_pypi_download_paid_app_from_app_store PASSED [ 83%]
30 tests/system_tests/test_system.py::test_docker_download_paid_app_from_app_store PASSED [ 88%]
31 tests/system_tests/test_system.py::test_stability_testing_for_appstore_logins PASSED [ 94%]
32 tests/system_tests/test_system.py::test_load_testing_google_play_download PASSED [100%]
33
34 ===== 18 passed in 522.93s (0:08:42) =====
```

Post Set up Python 3.9

Post Run actions/checkout@v3

Complete job

С кодом можно ознакомиться в репо:

[https://github.com/npinaev/poly\\_app\\_downloader/tree/main/tests/system\\_tests](https://github.com/npinaev/poly_app_downloader/tree/main/tests/system_tests)

с прохождением тестов:

[https://github.com/npinaev/poly\\_app\\_downloader/actions/workflows/Poly\\_app\\_downloader\\_CICD.yml](https://github.com/npinaev/poly_app_downloader/actions/workflows/Poly_app_downloader_CICD.yml)