**Тест-план**

**для контроля системы «SILUR»**

*Версия 0.2*

История исправлений

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 01.09.2021 | 0.1 | Начало разработки | Данильчук К. М. |
| 17.09.2021 | 0.2 | Изменение требований | Данильчук К. М. |

ОГЛАВЛЕНИЕ

[1 ВВЕДЕНИЕ 2](#_Toc90708900)

[1.1 Цель 2](#_Toc90708901)

[1.2 Исходные данные 2](#_Toc90708902)

[1.3 Обзор системы 2](#_Toc90708903)

[1.4 Применяемые документы 2](#_Toc90708904)

[1.5 Главный план-график работ 2](#_Toc90708905)

[2 РАСПРЕДЕЛЕНИЕ РОЛЕЙ И ОБЯЗАННОСТЕЙ 3](#_Toc90708906)

[2.1 Структура проекта 3](#_Toc90708907)

[2.2 Роли и обязанности в проекте 3](#_Toc90708908)

[2.3 Состав группы тестирования 3](#_Toc90708909)

[3 ПРОГРАММА ТЕСТИРОВАНИЯ 3](#_Toc90708910)

[3.1 Рамки программы тестирования 3](#_Toc90708911)

[3.2 Стратегии тестирования 3](#_Toc90708912)

[3.3 Средства автоматизации 4](#_Toc90708913)

[3.4 Проектирование тестов 4](#_Toc90708914)

[3.4.1 Схема тестирования 4](#_Toc90708915)

[3.4.2 Соглашение по именованию тестовых процедур 4](#_Toc90708916)

[3.5 Разработка тестов 4](#_Toc90708917)

[3.5.1 Методы эквивалентного разбиения 4](#_Toc90708918)

[3.5.2 Метод анализа граничных условий 6](#_Toc90708919)

[3.5.3 Метод покрытия операторов 7](#_Toc90708920)

[3.5.4 Метод покрытия переходов 9](#_Toc90708921)

# 1 ВВЕДЕНИЕ

## 1.1 Цель

Цель документа “План тестирования системы” - координация усилий участников проекта в части контроля качества. Документ предназначен руководству проекта для согласования планов и оценки затрат. Документ предназначен группе тестирования для ознакомления с характером предстоящих работ, анализа и разбиения на подзадачи.

## 1.2 Исходные данные

Исходными данными являются:

1. Спецификация требований к ПО.
2. Результаты проектирования в виде формальных моделей, соответствующих выбранной стратегии проектирования.
3. Программный код продукта.

## 1.3 Обзор системы

Программный продукт в виде серверной части приложения разрабатывается с целью автоматизации поиска новостей из разных телеграмм каналах и их рассылке в Telegram каналы с фильтрацией по темам.

## 1.4 Применяемые документы

Применяется ГОСТ ISO/IEC 12207:1995

## 1.5 Главный план-график работ

Тестирование проекта – 18.12.2021

# 2 РАСПРЕДЕЛЕНИЕ РОЛЕЙ И ОБЯЗАННОСТЕЙ

## 2.1 Структура проекта

Серверная часть SILUR 0.2 представляет собой независимый программный продукт, который используется приложениями-клиентами посредствам межпроцессорного взаимодействия.

## 2.2 Роли и обязанности в проекте

Управление проектом, функциональные требования, разработка программ, проектирование системы, контроль продукта – Данильчук К. М.

## 2.3 Состав группы тестирования

Основная группа тестирования:

1. Данильчук К. М.

# 3 ПРОГРАММА ТЕСТИРОВАНИЯ

## 3.1 Рамки программы тестирования

Программа тестирования должна проверять:

* входные данные;
* выходные данные;
* алгоритм работы;
* функциональные требования.

## 3.2 Стратегии тестирования

Для данного проекта используется стратегия модульного тестирования.

## 3.3 Средства автоматизации

Для автоматизации тестирования используется среда xUnit выраженная в библиотеке GTest.

## 3.4 Проектирование тестов

## 3.4.1 Схема тестирования

Используется следующая схема тестирования:

1. Подготовить тестовые входные данные.
2. Подготовить ожидаемые выходные данные.
3. Подать тестовые выходные данные в тестируемый модуль и получить выходные данные.
4. Сравнить полученные данные с ожидаемыми.

## 3.4.2 Соглашение по именованию тестовых процедур

Все тестовые процедуры должны иметь имя теста отражающего суть теста, а также имя множества тестов, к которому данный тест принадлежит.

## 3.5 Разработка тестов

## 3.5.1 Методы эквивалентного разбиения

Внешние спецификации тестируемого модуля:

Модуль должен проверять корректность названий Telegram-каналов. Имя задается неограниченной строкой. Имя считается корректным если: первый символ - @, далее символы английского алфавита, цифры или нижние подчеркивания. Имя должно быть длиннее 1 символа и короче 25 символов. Пример корректного имени: «@channe\_name123\_34».

Выделяем классы эквивалентности:

|  |  |  |
| --- | --- | --- |
| Функциональные характеристики | Правильные классы | Неправильные классы |
| Длина | В диапазоне от 2 до 25 (1) | Строка < 2 (2)  Строка > 25 (3) |
| Первый символ @ | Да (4) | Нет (5) |
| Символы в названии канала | Английские буквы (6),  цифры (7),  нижние подчеркивания (8),  Английские буквы и цифры (9)  Английские буквы и нижние подчеркивания (10)  Нижние подчеркивания и цифры (11)  Английский, цифры и нижние подчеркивания (12) | Любой другой символ (13) |

Тесты правильных классов

|  |  |  |
| --- | --- | --- |
| № | Тест | Номера классов |
| 1 | «@silur\_test\_channel\_name»  Эталон: true | 1 |
| 2 | «@silur\_test»  Эталон: true | 4 |
| 3 | «@english»  Эталон: true | 6 |
| 4 | «@1231234»  Эталон: true | 7 |
| 5 | «@\_\_»  Эталон: true | 8 |
| 6 | «@number123»  Эталон: true | 9 |
| 7 | «@line\_\_x»  Эталон: true | 10 |
| 8 | «@123123\_\_123123»  Эталон: true | 11 |
| 9 | «@test\_123»  Эталон: true | 12 |

Тесты неправильных классов

|  |  |  |
| --- | --- | --- |
| № | Тест | Номера классов |
| 10 | «@s»  Эталон: false | 2 |
| 11 | «@abcdefghijklmnopqrstuvwxyz»  Эталон: false | 3 |
| 12 | «no»  Эталон: false | 4 |
| 13 | «@ыфыва»  Эталон: false | 13 |

## 3.5.2 Метод анализа граничных условий

Из анализа выявлены следующие граничные условия:

1. Количество символов в названии <= 2.
2. Количество символов в названии >= 25.
3. Необходимо проверить, что первый символ это @.
4. Пустая строка. Длина строки равно 0.

Появляются тесты:

14. Строка состоит из символа @ и английской буквы (покрывает 2, 3, 5). Покрывается тестом 10.

15. Строка состоит из символа @ и 2х английских букв (покрывает 1, 3, 5). Тест: «@ab». Ответ: true.

16. Строка состоит из символа @ и 3х английских букв (покрывает 1, 3, 5). Тест: «@abc». Ответ: true.

17. Строка состоит из символа @ и 24х английских букв (покрывает 1, 3, 5). Тест: «@abcdefghijklmnopqrstuvwx». Ответ: true.

18. Строка состоит из символа @ и 25 английских букв (покрывает 1, 3, 5). Тест: «@abcdefghijklmnopqrstuvwxy». Ответ: true.

19. Строка состоит из символа @ и 26х английских букв (покрывает 2, 3, 5). Покрывается тестом 3.

20. Первый символ строки @. Покрывается тестам 2.

21. Первый символ строки не @. Покрывается тестом 12.

22. На вход подается пустая строка (покрывает 2, 4, 6). Тест 7: «». Ответ: false.

## 3.5.3 Метод покрытия операторов



В данном алгоритме невозможно за один тест сразу проверить все операторы, т.к. у функции несколько выходов.

Составим несколько покрытий, чтобы в объединении получилось множество всех операторов.

Первый критерий покрытия операторов удовлетворяет путь: a.

Путь a.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Оператор | Значение предиката | Данные |
| 1 | P1 | P1 == false |  |
| 2 | a |  | Return value = false |

Предикат пути:

!P1

Тогда:

channel\_name.front() != «@»

Решением может быть набор название канала (Тест 23): «test\_channel». Ответ: false.

Второй критерий покрытия операторов удовлетворяет путь: b, c

|  |  |  |  |
| --- | --- | --- | --- |
| № | Оператор | Значение предиката | Данные |
| 1 | P1 | P1 == true |  |
| 2 | b |  |  |
| 3 | P2 | P2 == true |  |
| 4 | c |  | Return value = false |

Предикат пути:

P1 && P2 == true

(channel\_name.front() == «@») == true

!(1 < channel\_name.size() && channel\_name.size() < 25) == true

Решением может быть набор название канала (Тест 24): «@test\_channel». Ответ: true.

Третий критерий покрытия операторов удовлетворяет путь: b, e

|  |  |  |  |
| --- | --- | --- | --- |
| № | Оператор | Значение предиката | Данные |
| 1 | P1 | P1 == true |  |
| 2 | b |  |  |
| 3 | P2 | P2 == false |  |
| 4 | P3 | P3 == true |  |
| 5 | P4 | P4 == true |  |
| 6 | e |  | Return value = false |

Предикат пути:

P1 && !P2 && P3 && P4

(channel\_name.front() == «@») == true

!(1 < channel\_name.size() && channel\_name.size() < 25) == false

(auto&& symbol : channel\_name) == true

!(IsEnglishSymbol(symbol) || isdigit(symbol)

|| IsUnderscore(symbol)) == true

Решением может быть набор название канала (Тест 25): «@ыъ». Ответ: false.

## 3.5.4 Метод покрытия переходов

## Критерию соответствуют 4 пары путей

В предыдущем методе были покрыты все операторы, а также условные блоки, но этого недостаточно, т.к. существует путь, когда P4 ложно. Метод покрытия переходов решает данную задачу.

Для критерия необходимо протестировать оставшийся путь: b, d, когда условие P4 ложно.

Путь b, e:

P1 && !P2 && P3 && !P4 && P3 && !P4 && P3

Тогда:

(channel\_name.front() == '@') == true

!(1 < channel\_name.size() && channel\_name.size() < 25) == false

auto&& symbol : channel\_name == true

!(IsEnglishSymbol(symbol) || isdigit(symbol) || IsUnderscore(symbol)) == false

auto&& symbol : channel\_name == true

!(IsEnglishSymbol(symbol) || isdigit(symbol) || IsUnderscore(symbol)) == false

auto&& symbol : channel\_name == false

Решением может быть название канала (Тест 26): «@ab». Ответ true.