

До

QALight

Mykola Boboshko

Від**Groups 5&6****Дата**

Створено: 03 квітня 2024

Змінено востаннє: 06 квітня 2024

TEST PLAN

Версія 1.7

Тест план розроблений для тестування програми List Boxer version 1.98.

Документ описує підходи до тестування, планування та ресурси, критерії та ризики.

Версія	Дата	Автор	Зміни
1.0	03 квітня 2024	Руслан Салманов	Створено, додано розділи: Призначення, 1 та 2
1.1	03 квітня 2024	Олександра Шматко	Додано розділ 3
1.2	04 квітня 2024	Вікторія Строга	Додано розділ 4
1.3	04 квітня 2024	Катерина Сухорукова	Додано розділ 5
1.4	05 квітня 2024	Галина Сидоренко	Додано розділ 6
1.5	05 квітня 2024	Марина Слабеняк	Додано розділ 7
1.6	06 квітня 2024	Альона Швець	Додано розділи 8, 9
1.7	06 квітня 2024	Ніна Труфанова	Додано розділ 10

Зміст

Призначення	3
1. ПОСИЛАННЯ.....	3
2. МЕЖІ РЕЛІЗУ	3
3. ЩО НЕ ВХОДИТЬ В ТЕСТУВАННЯ	4
4. ПЛАНУВАННЯ	5
4.1. Milestones	5
4.2. Спринти.....	7
4.3. Спринт розробки.....	7
4.4. Спринти тестування	8
5. ПІДХІД ДО ТЕСТУВАННЯ	9
5.1. Загальний підхід	9
5.2. Аналіз вимог	9
5.3. Підготовка тестового плану.....	9
5.4. Вибір методології та інструментів тестування.....	10
5.5. Розробка тестових сценаріїв та кейсів.....	10
5.6. Затвердження та налаштування тестових середовищ.....	11
5.7. Аналіз результатів тестування.....	11
5.8. Підготовка звіту про тестування	11
6. ПРОЦЕС УПРАВЛІННЯ ТЕСТУВАННЯМ	12
6.1. Порядок тестування	12
6.2. Перевірка та управління дефектами	14
7. КРИТЕРІЇ ТЕСТУВАННЯ	14
7.1. Критерії входу в тестування (Enter Criteria).....	14
7.2. Критерії прийняття (Acceptance Criteria)	14
7.3. Критерії часу тестування (Time Criteria)	16
8. КРИТЕРІЇ ПРИПИНЕННЯ ТЕСТУВАННЯ (STOP CRITERIA)	17
9. КРИТЕРІЇ ЗАВЕРШЕННЯ ТЕСТУВАННЯ (EXIT CRITERIA).....	17
10. РИЗИКИ.....	18

Призначення

Цей документ створений для детального опису процесу тестування програми List Boxer версії 1.98. Мета цього документа - забезпечити ретельне тестування за допомогою оптимальних технік, методів та підходів, що дозволять провести ґрунтовну перевірку функціональності, відповідно до основних принципів тестування програмного забезпечення.

Призначення програми List Boxer – надання користувачам зручного та ефективного інструменту для організації та керування даними, шляхом створення списків, що можуть бути використані у наукових дослідженнях, управлінні проектами або при особистому плануванні.

Завдяки широкому спектру функцій користувачі можуть ефективно керувати своїми даними.

Основні функції, які будуть тестуватися, включають:

- Створення універсальних алфавітно-цифрових списків.
- Модифікація списків, що існують.
- Збереження списків у вигляді файлів на диску.
- Відкриття збережених списків.
- Видалення списків.

Проведення детального тестування основних функцій програми List Boxer версії 1.98 допоможе досягти успішної реалізації запланованого функціонала.

1. ПОСИЛАННЯ

Документ	Опис
Вимоги до програми	Функціональні вимоги до List Boxer версія 1.98 (версія вимог 0.2 – доповнені від 06.04.2024)
Traceability Matrix	Матриця покриття функціональності програми тест кейсами

Табл. 1 Посилання на документи

2. МЕЖІ РЕЛІЗУ

Функції для тестування в запланованій ітерації:

- Тестування створення, модифікації, збереження, відкриття та видалення універсальних алфавітно-цифрових списків.
- Перевірка роботи з різними режимами відображення списків.

- Тестування елементів керування (кнопок) для зручності користувача.
- Тестування застосування фільтрів для керування відображенням введеної інформації.
- Тестування функції сортування інформації у списках.
- Тестування додавання нових елементів у списки.
- Тестування видалення інформації зі списків.
- Перевірка скасування останньої виконаної дії користувачем.
- Перевірка наявності контекстної довідки для підтримки користувача.
- Тестування можливості виклику опису програми для отримання додаткової інформації.
- Тестування підтримки гарячих клавіш для швидкого доступу до функцій.
- Тестування інсталяції/деінсталяція.

Припущення

Припускається, що всі вхідні дані для тестування будуть належним чином сформовані та доступні, а також всі необхідні ресурси будуть доступні протягом процесу тестування.

Обмеження

Можливі обмеження включають час, персонал, а також технічні обмеження, такі як обмеження апаратного та програмного забезпечення.

Детальний опис функцій, що тестуватимуться зазначено у документі Traceability Matrix.

3. ЩО НЕ ВХОДИТЬ В ТЕСТУВАННЯ

На основі наданих функціональних вимог до "List Boxer версії 1.98" командою було прийнято рішення винести за рамки тестування деякі нефункціональні характеристики програми. Згідно з рішенням в даній ітерації не підлягають тестуванню:

A. Тестування сумісності з застарілим обладнанням

Перевірка сумісності програми з мінімальними вимогами вказаними у функціональних вимогах: операційна система Windows XP SP2.

Windows XP SP2 - застаріла, більше не підтримується виробником. Тому тестування на цій операційній системі може бути непрактичним або неможливим з огляду на відсутність підтримки. Процесор Intel Celeron 440 @ 1.00 GHz, ОЗП 1 ГБ, відеоадаптер 64 - ці вимоги стосуються дуже застарілих компонентів, тому тестування на такому обладнанні майже неможливе та недоцільне.

В. Тестування стійкості

Програмне забезпечення не має вимог щодо стійкості до виникнення помилок, та вимог щодо поведінки при тривалій роботі для виявлення можливих падінь або витоків ресурсів/непередбачених ситуацій, через що, тестування на стійкість не проводимо.

С. Тестування безпеки

Програмне забезпечення не взаємодіє з мережею і зовнішніми джерелами даних, тестування безпеки не буде проводитись через відсутність вимог щодо безпеки або обмежень доступу до даних. Не перевіряємо на вразливості, зокрема, де програма зберігає або обробляє конфіденційні дані. Робота в програмі не передбачає захист, та конфіденційність даних.

Д. Тестування локалізації та міжнародної підтримки

Перевірка коректності відображення тексту та інших елементів інтерфейсу для різних мов та регіональних налаштувань не передбачено функціональними вимогами.

Е. Тестування відновлення після збоїв

Перевірка можливості відновлення роботи програми після непередбачених ситуацій або аварійного завершення роботи не передбачено функціональними вимогами.

F. Тестування масштабованості

Перевірка працездатності програми при обробці великої кількості даних або великої кількості користувачів одночасно не передбачено функціональними вимогами. Якщо програмне забезпечення призначено для використання багатьма користувачами одночасно, але вимоги не містять інформацію щодо масштабованості до великої кількості одночасних користувачів, тоді тестування на масштабованість не потрібне.

4. ПЛАНУВАННЯ

4.1. Milestones

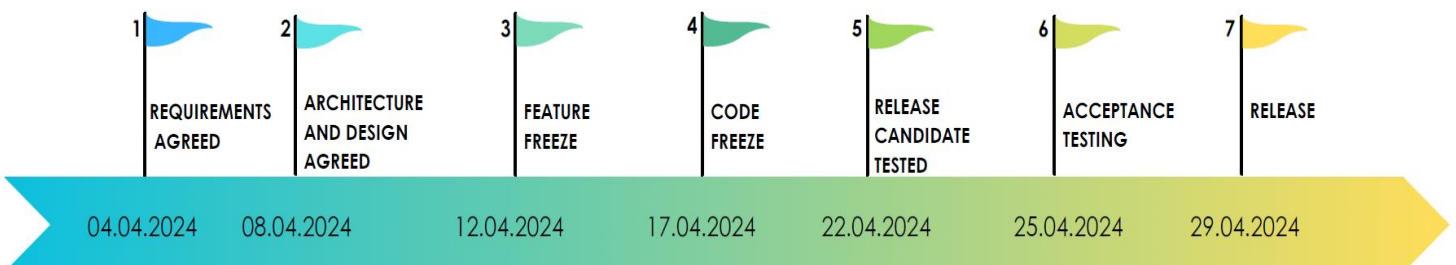
В процесі розробки визначені умовні Milestones (Віхи), які допоможуть синхронізувати роботу команди розробників (DEV) та тестувальників (QA).

Досягнення певного Milestone означає, що запланований обсяг робіт виконано, проект знаходиться в такому стані, який дозволяє переходити до наступного етапу роботи всім учасникам команди.

Графік Milestones є орієнтовним, передумовою досягнення того чи іншого Milestone є досягнення його цілей, а не настання певної дати.

На графік досягнення Milestone можуть впливати:

- a) будь-які проблеми в готовності системного середовища:
- b) будь-яка зміна обсягу/додавання обсягу
- c) будь-яка інша залежність, яка впливає на зусилля та часові рамки



Графік 1. Milestone проекту (орієнтовний)

Milestone	Опис
1. REQUIREMENTS AGREED	<p>Project/Product State: Вимоги надані замовником та/або зібрані та вивчені командою. Загальне бачення вимог до функціональності продукту погоджено командою та Product owner. Вимоги протестовані.</p> <p>Activities: QA на основі вимог починають написання тест-плану та іншої тестової документації. Команда DEV починає розробляти архітектуру ПЗ.</p>
2. ARCHITECTURE AND DESIGN AGREED	<p>Project/Product State: Команда DEV має розроблену архітектуру та дизайн продукту.</p> <p>Activities: QA завершують написання тест - плану, пишуть checklist, test cases. DEV починають написання коду та розробку функціоналу.</p>
3. FEATURE FREEZE	<p>Project/Product State: Розроблений основний функціонал програми. Unit та Integration тестування проведено DEV на стадії розробки. Build готовий до системного тестування. Тестова документація (checklist, test cases) готова.</p> <p>Activities: QA починають цикл функціонального тестування. DEV працюють над виправленням знайдених Blocker та Critical багів.</p>
4. CODE FREEZE	<p>Project/Product State: Всі Blocker та Critical баги виявлені та виправлені. Продукт стабільний.</p> <p>Activities: QA приступає до регресійного тестування, та Performance та Usability testing, щоб передати Build для User Acceptance Testing.</p>

5. RELEASE CANDIDATE TESTED	<p>Project/Product State: Build перевірено, нових багів не виявлено, продукт готовий для розгортання на тестовому середовищі для UAT. Traceability matrix повністю заповнена командою QA.</p> <p>Activities: QA та DEV готують build та технічну документацію для UAT.</p>
6. ACCEPTANCE TESTING	<p>Activities: Замовник проводить приймальне тестування.</p>
7. RELEASE	<p>Project/Product State: Traceability matrix заповнена і Замовником і QA. Всі заплановані роботи виконані.</p>

Табл. 2 Опис Milestones

4.2. Спринти

Розробка програми запланована в 1 спринт. Тестування буде проводитися в 2 спринти. Модульне та інтеграційне тестування проводиться командою DEV на стадії розробки. Спринт тестування 1 буде складатися з повного циклу функціонального тестування системи для перевірки основного функціоналу програми (critical path test). Метою першого спринту тестування є виявлення, в першу чергу, будь-яких Blocker, Critical та Major дефектів. Спринт тестування 2 буде включати в себе цикл регресійного тестування, тестування продуктивності та Usability test. Окремим циклом буде проведено User Acceptance Testing (UAT) на тестовому середовищі за участі Замовника.

Ітерація	Початок	Кінець	Тривалість
Development sprint	02.04.2020, 09:00 am	17.04.2024, 18:00 am	16 days
QA Sprint 1	02.04.2020, 09:00 am	16.04.2024, 18:00 am	15 days
QA Sprint 2	17.04.2024, 09:00 am	22.04.2024, 18:00 am	6 days

Табл. 3 Розклад DEV та QA Sprints

4.3. Спринт розробки

Робота команди тестувальників буде синхронізована з роботою команди розробників. Розробники, в свою чергу, протягом спринта розробки виконують наступні задачі:

TASK	DEV SPRINT	
	1	
Вивчення та аналіз вимог		✓
Розробка архітектури програми, знаходження технічних рішень для реалізації розробки		✓
Написання коду основного функціоналу програми		✓
Проведення Unit та Integration тестів		✓
Розробка User Interface		✓

Табл. 4 Завдання на DEV Sprint

4.4. Спринти тестування

Як зазначалось вище команда QA проведе тестування в два спринти. Цикли тестування, які будуть проведені включають в себе такі етапи: підготовка (Test Preparation), тестування (Testing), аналіз результатів (Test Result Analysis). В таблиці нижче літерою Т позначена активність, яка означає тестування, літерою Р позначена активність яка означає підготовку

TASK	QA SPRINT			
	1		2	
	T	P	T	P
Вивчення та аналіз вимог		•		
Тестування вимог	•	•		
Написання тест-плану		•		
Написання чек листів		•	•	•
Написання тест кейсів		•	•	•
Формування матриці покриття	•	•		•
Функціональне тестування	•	•	•	•
Регресійне тестування		•	•	•
Тестування продуктивності		•	•	•
Usability тестування		•	•	•

Табл. 5 Активності QA Sprints

5. ПІДХІД ДО ТЕСТУВАННЯ

5.1. Загальний підхід

Як зазначається в розділі запланованих етапів роботи (див. розділ №4), тестування програми List Boxer версії 1.98 включатиме наступні цикли:

1. Функціональне тестування (включно з регресією);
2. Тестування GUI;
3. Тестування продуктивності;
4. Приймальне тестування на тестовому середовищі.

Тестування програми List Boxer версії 1.98 буде проводитися за допомогою мануального тестування з метою перевірки функціональності, взаємодії користувача з програмою, а також відповідності програми встановленим вимогам і очікуванням користувачів. Тестувальники будуть виконувати тестові сценарії вручну, без використання автоматизованих інструментів. Це дозволить глибше зрозуміти користувацький досвід та виявити потенційні проблеми, які можуть не бути очевидними при автоматизованому тестуванні. Використання автоматизації на даному проекті вважається недоречним, через низьку складність програми та обмеженість бюджету.

Юніт та інтеграційне тестування проводитиметься розробниками на рівні написання коду.

Для тестування проекту List Boxer версії 1.98, який призначений для створення та управління алфавітно-цифровими списками, визначено етапи, інструменти та техніки які описані нижче.

5.2. Аналіз вимог

Першим етапом у процесі тестування програмного забезпечення є аналіз вимог. Це критично важливо, оскільки дозволяє зрозуміти, що саме має бути реалізовано у програмі та які критерії використовуватимуться для оцінки її роботи.

5.3. Підготовка тестового плану

Визначення цілей тестування. Цілями тестування програми List Boxer версії 1.98 є забезпечення того, що програма:

- Відповідає всім функціональним вимогам: переконатися, що програма може створювати, модифікувати, зберігати, відкривати та видаляти алфавітно-цифрові списки згідно з визначеними специфікаціями.
- Ефективна та продуктивна: оцінити продуктивність програми, особливо при роботі з великими даними та під час тривалого використання.
- Зручна у використанні: переконатися, що інтерфейс користувача інтуїтивно зрозумілий та легкий у використанні.
- Сумісна з системними вимогами: перевірити, що програма працює належним чином на операційній системі Windows з мінімальними системними вимогами.

У результаті тестування ми отримаємо:

- Звіт про тестування: Документ, який містить деталі про виконані тестові сценарії, виявлені помилки та проблеми, а також рекомендації щодо їх виправлення.
- Оцінку якості: Загальне розуміння якості програми, її стабільності та готовності до випуску.
- Рекомендації щодо поліпшення: Пропозиції щодо покращення функціональності, продуктивності та користувацького інтерфейсу.

5.4. Вибір методології та інструментів тестування

Так як в ході обговорення і пошуку оптимальної методології, було прийнято рішення про дотримання командою Agile-принципів та Scrum-фреймворку для розробки програми List Boxer версії 1.98, процес тестування починається на ранніх стадіях, а саме з аналізу вимог. Робота за принципами Agile з використанням Scrum-підходу до управління, дозволить нашій команді швидко адаптуватися до змін, регулярно отримувати зворотний зв'язок від користувачів та забезпечити постійне вдосконалення продукту. Крім того, Scrum допоможе організувати робочий процес, встановити чіткі ролі та відповідальності, а також спланувати короткі ітерації роботи, що сприятиме більш гнучкій та ефективній розробці.

При мануальному тестуванні програми List Boxer версії 1.98, командою будуть використовуватися такі інструменти:

- ❖ Трекер дефектів: Jira, для відстеження та управління помилками, що були виявлені під час тестування.
- ❖ Інструменти для спілкування: Discord, Zoom, Telegram, поштові голуби для комунікації між членами команди.
- ❖ Спредшіти: Google Sheets, для створення документації.
- ❖ Інструменти для знімків екрану та запису відео: Monosnap, FastStone Capture - для документування візуальних дефектів.

Ці інструменти допоможуть організувати процес мануального тестування, забезпечити точне документування та ефективну комунікацію між членами команди.

5.5. Розробка тестових сценаріїв та кейсів

Розробка тестових сценаріїв відбудуватиметься з використанням наступник технік тест-дизайну:

- Еквівалентне розбиття (Equivalence Partitioning): Ця техніка дозволяє ефективно визначити тестові випадки, зменшуючи кількість тестів, необхідних для покриття всіх можливих вхідних даних.
- Аналіз граничних значень (Boundary Value Analysis): Вона допомагає виявити помилки, які часто виникають на межах вхідних даних, і є важливою для перевірки коректності обробки даних програмою.
- Попарне тестування (Pairwise Testing): Ця техніка використовується для забезпечення того, що всі можливі комбінації параметрів були протестовані хоча б один раз, що є ефективним для виявлення взаємодій між параметрами.

Ці техніки допоможуть створити комплексний набір тестових сценаріїв, які забезпечать гарне покриття функціональності програми і допоможуть виявити потенційні дефекти.

5.6. Затвердження та налаштування тестових середовищ

№	QA	Тип ОС	Характеристики системи
1	Р. Салманов	Windows 10 Pro x64 22H2 Deutsche local	Intel Core i7 4770k 4.2 Ghz 32GB DDR3 2133 mhz RAM
2	В. Строга	Windows 11 x64 Home 23H2	Intel(R) Core(TM) i5-8300H CPU 2.30GHz 8 GB DDR4 2400 MHz
3	К. Сухорукова	Windows 10 Pro x64 22H2	Intel Core i7 6700 @ 3.40GHz 32GB Dual-Channel DDR4 1064MHz Intel(R) HD Graphics 530
4	А. Швець	Windows 7 x64	Proxmox Virtual Environment Common KVM processor 2.27 GHz Installed memory (RAM): 4.00 GB
5	М. Слабеняк	Windows 10 Pro x64 22H2	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz 16,0 GB
6	О. Шматко	Windows 10 Home Single Language x64 22H2	Intel(R) Core(TM) i7-7Y75 CPU @ 1.30GHz 1.61 GHz ОЗП 8,00 GB
7	Г. Сидоренко	Windows 11 x64 Home	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 2.90 GHz Installed RAM 8.00 GB (7.65 GB usable)
8	Н. Труфанова	Windows 7 SP1	AMD FX(tm)-8300 Eight-Core Processor 3.30 Gzh

Табл. 6 Тестові середовища

5.7. Аналіз результатів тестування

Оцінка виявлених дефектів та їхнього впливу на роботу програми.

Буде проведена оцінка тенденції та типів дефектів, що виявлені під час тестування. Шляхом аналізу кількості та серйозності критичних дефектів зроблено висновок про стабільність системи, та чи відповідає функціональність системи встановленим вимогам.

5.8. Підготовка звіту про тестування

На заключному етапі буде зроблено документування результатів тестування та пропозицій щодо покращення. Звіт про тестування включатимемо в себе:

- Використані техніки, інструменти, середовище та інші деталі, що стосуються тестування
- Зведену інформацію про обсяг тестування, включаючи кількість тест-кейсів, обсяг покриття функціональності, аспекти, які були протестовані.
- Огляд результатів тестування, включаючи кількість виявлених дефектів, їхній статус та серйозність, а також загальні висновки щодо стабільності та продуктивності системи.
- Загальні висновки щодо якості продукту, виявлені сильні та слабкі сторони, рекомендації для подальшого вдосконалення та випуску продукту.

6. ПРОЦЕС УПРАВЛІННЯ ТЕСТУВАННЯМ

6.1. Порядок тестування

Порядок тестування буде залежати від пріоритетності функції програми. Першочерговість виправлення дефектів буде залежати від ступеню їх впливу на роботу програми. Знайденим дефектам буде призначений ступінь критичності їхнього впливу на працевздатність програми згідно таблиці:

Критичність	Вплив
Blocker	Ця помилка достатньо критична, щоб привести до збою системи, спричинити пошкодження файлів або привести до потенційної втрати даних; Це викликає некоректне повернення до операційної системи (з'являється повідомлення про збій системи). Це призводить до зависання програми та потребує перезавантаження системи.
Critical	Відсутній будь-який шлях до запуску бажаної функціональності.
Major	Ця помилка погіршить якість системи. Однак є інтелектуальний обхідний шлях для досягнення бажаної функціональності - наприклад через інший екран. Ця помилка не дозволяє тестувати інші області продукту, однак інші області можна перевірити незалежно.
Minor	Графічні дефекти
Trivial	Дефекти, пов'язані з текстом

Табл. 7 Критичність багів

Покроковий процес виконання тестування і виправлення дефектів в роботі команди представлений на Рис.1

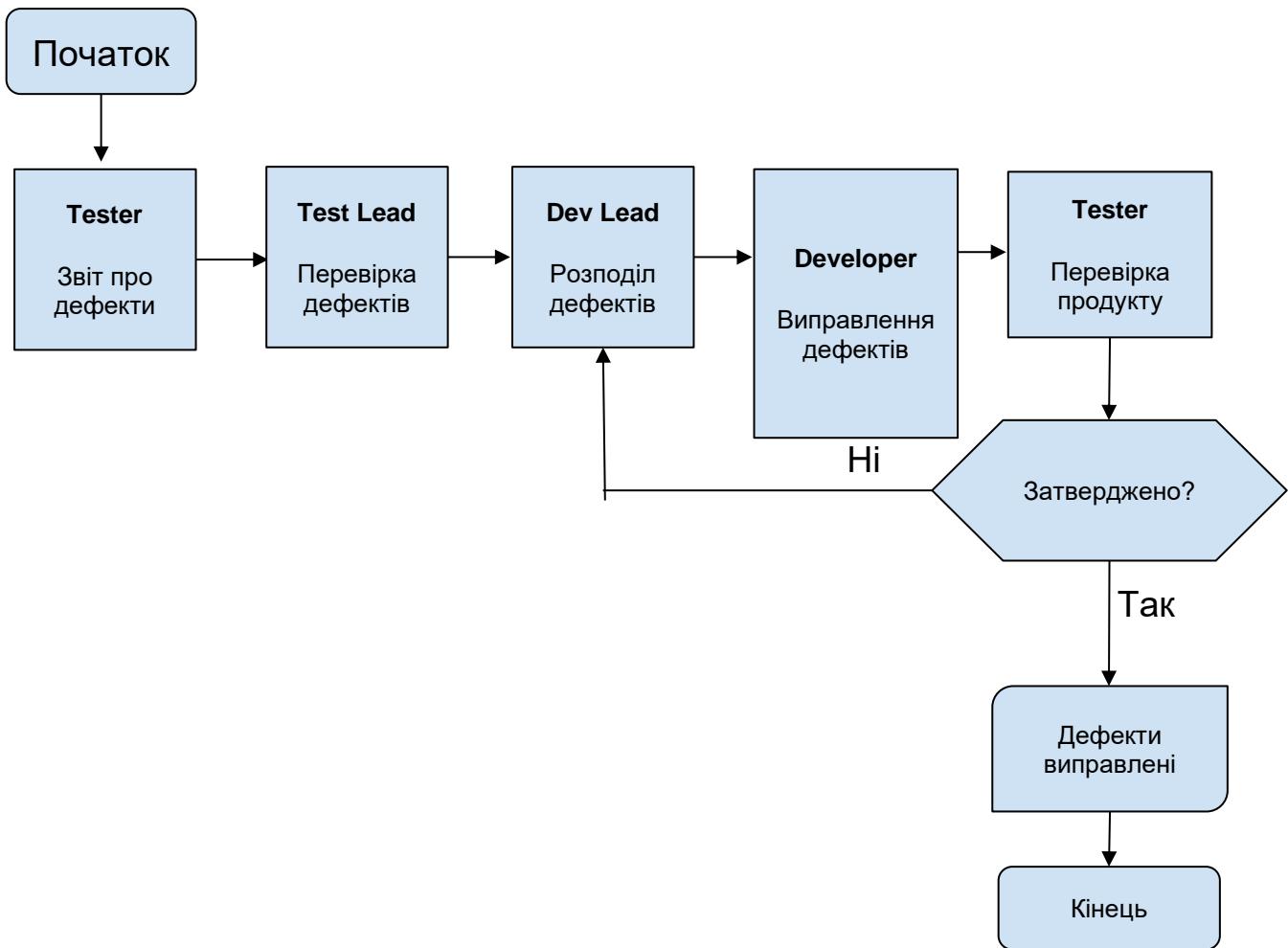


Рис.1 Процес виправлення дефектів

Кожен тестувальник матиме прямий доступ до відповідних призначених тестів і оновлюватиме статус (passed, failed, skipped,N/A). Під час тестування на виправлення дефектів дефекти повторно призначаються тестувальніку для перевірки виправлення дефекту.

Тестувальник перевіряє виправлення дефекту та оновлює статус (fixed, reopened або closed).

6.2. Перевірка та управління дефектами

Очікується, що тестувальники виконають усі сценарії в кожному з циклів тестування. Однак бізнес-аналітики можуть доєднатися до аналізу результатів тестування, оскільки Бізнес-Аналітики мають глибші знання бізнес-процесів. Якщо виявлено прогалину, сценарії та матриця відстеження будуть оновлені, а потім у сценаріях буде зареєстровано дефект.

Дефекти відстежуватимуться лише через Jira. Технічна команда збирає інформацію про дефекти щоденно і працює над виправленнями за підтримки команди розробників.

- Обов'язком тестувальника є відкриття дефектів, зв'язування їх із відповідним скриптом, призначити початковий ступінь тяжкості та статус, повторно протестувати та закрити дефект;
- Відповідальність Dev Lead - перевірити серйозність дефектів і допомогти технічній команді виправити їх, спілкуватися з тестувальниками з причин, коли тест можна продовжити або його слід зупинити;
- відповідальність технічної групи Developers - щоденна перевірка Jira на предмет виявленіх нових дефектів і усунення дефектів, повідомити команду про зроблені виправлення.

7. КРИТЕРІЇ ТЕСТУВАННЯ

7.1. Критерії входу в тестування (Enter Criteria)

1. Вимоги до програмного забезпечення List Boxer версії 1.98 наявні.
2. Тестова документація розроблена.
3. Наявний стабільний білд, готовий до тестування.
4. Налаштоване тестове середовище, яке відповідає мінімальним вимогам (див. "List Boxer requirements "- п. 1.2).
5. Команда тестувальників забезпечена доступом до баг-трекінгової системи Jira.
6. Необхідні ресурси для виконання тестів, включаючи людські, технічні та інші ресурси наявні.

7.2. Критерії прийняття (Acceptance Criteria)

Встановлення та погодження критеріїв прийняття гарантує відсутність критичних дефектів і несправностей, технічну готовність та функціональність згідно оцінки відповідності вимогам (див. "List Boxer requirements ").

Дляожної з основних вимог пропонуються наступні критерії прийняття:

Легенда результатів:

В - кількість знайдених Blocker,

С - кількість знайдених Critical,

Т - загальна кількість інших знайдених дефектів (Major, Minor, Trivial).

	Критерії до основних вимог	В	С	Т
1	Користувач може створювати універсальні алфавітно-цифрові списки.			
2	Користувач може модифікувати списки.			
3	Користувач може зберігати списки у вигляді файлів на диску.			
4	Користувач може відкривати списки.			
5	Користувач може видаляти списки.			
6	Користувач може використовувати режим роботи - алфавітний.			
7	Користувач може використовувати режим роботи - цифровий.			
8	Користувач може використовувати режим роботи - змішаний.			
9	Користувач може використовувати елементи управління - кнопки (у тому числі option buttons, checkbox).			
10	Користувач може управляти відображенням введеної інформації (фільтри).			
11	Користувач може сортувати інформацію, що відображається в списках.			
12	Користувач може додавати рядки до списку.			
13	Користувач може видаляти інформацію (у будь-який момент видалити всю введену інформацію).			
14	Користувач може скасувати останню дію (у будь-який момент за допомогою команди "Undo" в меню "Edit").			
15	Користувач може викликати контекстну довідку.			
16	Користувач може викликати опис програми (через меню "Help").			
17	Користувач може використовувати "гарячі" клавіші.			
	Загальна кількість			

Табл. 8 Приклад таблиці результатів

Щоб вважати функціонал та продукт в цілому прийнятним для замовника та кінцевого користувача, результати тестування мають відповідати наступним критеріям прийняття:

- 1)** Закриті всі баги з високою та середньою критичністю, які могли б значно вплинути на роботу програмного забезпечення або становити значний ризик для користувачів:
 - Blocker -0;
 - Critical - 0;
 - загальна кількість дефектів (Major, Minor, Trivial) в статусі "not fixed" - не більше 5.
- 2)** Користувач може успішно пройти будь-який сценарій з основної функціональності без блокувань та падінь системи.
- 3)** Інтерфейс програми інтуїтивно зрозумілий та зручний для користувача.
- 4)** Програмне забезпечення відповідає вимогам, які були визначені на початку проекту і відповідає очікуванням користувача.

7.3. Критерії часу тестування (Time Criteria)

Критерії часу виконує декілька важливих функцій:

- 1) Критерії часу допомагають визначити, скільки часу буде витрачено на виконання певних завдань або етапів тестування. Це дозволяє ефективно розподілити ресурси і забезпечити вчасне виконання всіх задач.
- 2) Критерії часу створюють вимоги до часу виконання кожного етапу тестування. Це дозволяє вчасно виявляти будь-які затримки або проблеми, що можуть виникнути під час процесу тестування, і вживати відповідних заходів.
- 3) Аналіз критеріїв часу після завершення тестування дозволяє оцінити ефективність тестувального процесу. Порівняння фактичного часу з визначеними критеріями допомагає виявити можливі області для покращення та оптимізації процесу.

	Етап	Критерії	Час, год.
1	Планування і підготовка до тестування.	Огляд вимог: перегляд документації, уточнення вимог; обговорення з розробниками; аналіз функціональності. Визначення стратегії тестування.	8
		Розробка детального плану тестування, визначення обсягу тестів, вибір типів тестування, створення тестових випадків.	8
		Підготовка тестових середовищ та необхідних даних.	5
2	Виконання тест-кейсів.	Проведення тестів відповідно до складених тест-кейсів (активне тестування).	30
		Реєстрація результатів тестування та виявлення помилок.	6
3	Аналіз результатів та внесення змін.	Аналіз результатів тестування. Підготовка звіту про тестування.	10
		Виявлення недійсності або функціональності та внесення змін у вимоги або програмне забезпечення.	10
4	Підготовка до релізу та фінальне тестування	Останнє тестування перед релізом. Підготовка та подання фінального звіту про тестування	15
		Проведення зустрічі з командою для обговорення результатів тестування та подальших кроків.	5
	Разом, год		104

Табл. 9 Оцінка часу на виконання задач (орієнтовно)

8. КРИТЕРІЇ ПРИПИНЕННЯ ТЕСТУВАННЯ (STOP CRITERIA)

Критерії припинення (Stop Criteria) визначають умови, за яких тестування або певний етап тестування слід припинити. Ці критерії встановлюються для того, щоб уникнути продовження тестування в разі досягнення певних негативних обставин або обмежень. Для тестування програми List Boxer version 1.98 командою визначено такі Stop Criteria:

- 1) **Часові обмеження.** Тестувальні дії можуть бути припинені, якщо вони перевищують строк, вказаний в плані проекту.
- 2) **Вичерпання ресурсів.** Тестування може бути припинене через вичерпання ресурсів, таких як бюджет або доступність персоналу.
- 3) **Перевищення обсягу дефектів.** Тестування може бути припинене, якщо кількість виявлених дефектів блокує та не дозволяє пройти основні користувачькі сценарії, що вказує на серйозні проблеми з якістю продукту.
- 4) **Недосяжність цілей тестування.** Тестування може бути припинене, якщо виявлено, що початкові цілі тестування не можуть бути досягнуті через технічні обмеження або інші обставини.
- 5) **Ухвалення рішення з перенесення проекту.** Якщо рішення про перенесення проекту прийнято з певних стратегічних або бізнес-причин, тестування може бути припинене.

9. КРИТЕРІЇ ЗАВЕРШЕННЯ ТЕСТУВАННЯ (EXIT CRITERIA)

Критерії завершення (Exit Criteria) визначають умови, за яких можна вважати, що тестування або певний етап тестування завершено успішно і можна переходити до наступних етапів розробки чи випуску продукту.

Для тестування програми List Boxer version 1.98 командою визначено, що тестування може бути завершене і вважатися успішно виконаним за наступних умов:

- 1) **Достатнє покриття тестування.** Всі критичні функціональності, робочі процеси та користувачькі сценарії були протестовані, а покриття тестування відповідає попередньо визначеним критеріям.
- 2) **Виконання критеріїв якості:** Певні метрики якості, такі як покриття коду, покриття тестування, кількість виявлених дефектів тощо, повинні досягти певних цілей.
- 3) **Відповідність термінам і бюджету:** Тестування повинно бути завершене в межах встановлених термінів і бюджету.
- 4) **Цілісність даних.** Додаток зберігає цілісність та послідовність даних під час різних операцій та маніпуляцій з даними.
- 5) **Усунення дефектів.** Всі ідентифіковані Blocker та Critical дефекти були виправлені і проведено повторне тестування для перевірки виправлень.
- 6) **Схвалення керівництвом.** Результати тестування були переглянуті та схвалені керівником проекту, що підтверджує відповідність програмного забезпечення вимогам та очікуванням.
- 7) **Прийняття користувачем.** Пройшло тестування прийняття користувачем (UAT) і застосунок був прийнятий кінцевими користувачами або клієнтами.

10. РИЗИКИ

В ході тестування програми, як і вході проекту загалом, існують певні ризики, які можуть вплинути на хід та терміни роботи. Ці ризики необхідно враховувати та мати попереднього узгоджений план з мінімізації їх впливу на проект.

Нижче подана таблиця, у якій відображені актуальні ризики, з якими може зіткнутися команда тестувальників програми List Boxer version 1.98, а також дії, які будуть виконані в разі, якщо той чи інший ризик матиме місце в реальності.

Ризик	Тригер	Ймовірність	Ступень впливу	Дії
Недооцінка обсягу робіт по тестуванню	Недооцінювання обсягу робіт по тестуванню може привести до затримок у виконанні проекту або незадовільної якості продукту.	середня	висока	Закласти додатковий час на доопрацювання та правки, налагодити комунікацію всередині команди.
Відключення електроенергії	Непередбачене відключення світла в області, де розгорнутий проект.	середня	середня	Забезпечення резервного живлення.
Втрата доступу до даних (проблеми з сервером)	Проблема, яка призвела до тимчасового падіння сервера або приводить в неробочий стан деяку частину системи.	низький	середня	Регулярне резервне копіювання даних у зашифрованому вигляді на зовнішні носії або в хмарні сховища для запобігання доступу до даних третім особам, застосування реплікації даних.
Непланове внесення змін у документацію/програму	Непланове або несвоєчасне оновлення документації членами команди. Нові функції, зміни або виправлення можуть бути впроваджені без відповідного оновлення документації.	середня	середня	Своєчасне оновлення та перегляд документації, автоматизація процесів управління версіями та систем документування, регулярний аудит документації.

Недоступність необхідного середовища тестування	Необхідні тестові середовища стають недоступними або несправними	низький	середня	Забезпечити резервними засобами для тестування.
Хвороба/звільнення члена команди	Хвороба/звільнення члена команди	низький	низький	Пріоритизація задач, залучення до команди кандидатів з бенчу (якщо строки дедлайну не можуть бути перенесені).
Форс-мажорні обставини пов'язані з військовим станом (Повітряні тривоги)	Частина команди тестиувальників знаходиться на території України	висока	висока	Забезпечити доступ команди до укриттів, які мають безперебійні джерела живлення, засоби зв'язку та всім необхідним для продовження роботи.
Відсутність доступу до інтернету	Технічні складнощі зі сторони провайдера	низький	низький	Забезпечити альтернативний доступ до мережі інтернет (мобільний інтернет, резервні лінії іншого провайдеру).

Табл. 10 Ризики

Запобігання цим ризикам та ретельне планування дозволять зменшити негативний вплив на процес тестування програми List Boxer version 1.98. На ризики командою закладається до 20% від загальної естимації проекту.