Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа № 6

По дисциплине «Разработка и анализ требований»

На тему «Управление требованиями»

Выполнила:

Студентка 2 курса 10 группы

Соленок Анастасия Александровна

Преподаватель: асс. Ромыш А.С.

2025, Минск

Тема проекта: программное средство книжный магазин.

**1. Список тест-кейсов**

На основе критериев приемки и сценариев — пронумеруем тест-кейсы (TC):

| **№ TC** | **Описание тест-кейса** |
| --- | --- |
| TC1 | Проверка отображения каталога книг с обложкой, названием, ценой, автором и жанром. |
| TC2 | Добавление товара в корзину и переход к оформлению заказа. |
| TC3 | Оформление заказа с заполнением всех обязательных полей. |
| TC4 | Оформление заказа без регистрации. |
| TC5 | Отображение сообщения и номера заказа после успешного оформления. |
| TC6 | Проверка ошибки при незаполненных обязательных полях. |
| TC7 | Проверка формата номера телефона (негативный сценарий). |
| TC8 | Проверка блокировки кнопки при отсутствии товара. |
| TC9 | Отображение ошибки при проблемах с сетью. |
| TC10 | Проверка времени открытия каталога (≤ 2 сек). |
| TC11 | Проверка удобства интерфейса (UX-тест). |
| TC12 | Проверка стабильности при вводе некорректных данных (без краша). |
| TC13 | Проверка отображения интерфейса на экранах от 1280×720 и выше. |

**2. Матрица трассировки требований**

| **Бизнес-требование** | **Функциональное требование** | **Тест-кейсы** |
| --- | --- | --- |
| Покупатель хочет просматривать каталог и оформлять заказ | Пользователь видит каталог с фото, названием, ценой, описанием | TC1, TC10 |
|  | Пользователь может добавить товар в корзину и оформить заказ | TC2, TC3, TC4 |
|  | После оформления выводится сообщение с номером заказа | TC5 |
|  | Возможно оформление без регистрации | TC4 |
|  | Валидация обязательных полей (имя, телефон, адрес) | TC3, TC6, TC7 |
|  | Нельзя оформить заказ, если товара нет в наличии | TC8 |
|  | При ошибке сети показывается сообщение и заказ не отправляется | TC9 |
|  | Интерфейс работает на экранах от 1280×720 и выше | TC13 |
|  | Интерфейс удобен для пользователей с минимальными знаниями ПК | TC11 |
|  | Приложение не должно крашиться при ошибке | TC12 |

**1.Выберем незначительное изменение**

Допустим, заказчик вносит **изменение в функциональное требование**:

**Изменение:**  
Добавить новое обязательное поле при оформлении заказа — **электронную почту (e-mail)**.

**2. Анализ влияния по матрице трассировки**

Это изменение затронет:

* Функциональные требования: необходимо уточнение валидации и списка обязательных полей.
* Тест-кейсы: необходимо обновить или добавить тесты с проверкой обязательности e-mail и его формата.
* UI/UX: форма оформления заказа должна быть расширена.
* Возможные изменения в архитектуре (модель данных, валидация, обработка ошибок).

**3. Таблица анализа влияния изменений**

| **Объект проекта** | **Действие по изменению** |
| --- | --- |
| Функц. требование FR-03 | Расширить требование: добавить поле "e-mail" как обязательное |
| Тест-кейсы TC3, TC6, TC7 | Обновить: добавить проверки на заполнение и валидацию e-mail |
| Новый тест-кейс TC14 | Добавить: проверка на неверный формат e-mail |
| UI/UX: форма оформления заказа | Обновить: добавить поле ввода e-mail, адаптировать макет |
| Архитектура (модель Order) | Изменить: добавить поле Email, обновить логику сохранения |
| Валидация | Расширить правила проверки на e-mail (формат, пустое значение) |

**1. Пример RACI-матрицы**

Пример RACI-матрицы для проекта «Книжный магазин»:

| **Задача / Роль** | **Аналитик** | **Разработчик** | **Тестировщик** | **Менеджер проекта** | **Дизайнер** |
| --- | --- | --- | --- | --- | --- |
| Сбор требований | R |  |  | A | C |
| Проектирование интерфейса | C |  |  | A | R |
| Разработка функционала заказа | C | R |  | A | I |
| Написание и выполнение тестов | I | C | R | A |  |
| Утверждение изменения требований | R |  | I | A |  |
| Внедрение системы | I | R | C | A |  |

**2. Что отражает RACI-матрица**

**RACI-матрица** определяет **роли и зоны ответственности** членов команды по каждой задаче проекта:

* **R (Responsible)** — отвечает за выполнение задачи.
* **A (Accountable)** — несет конечную ответственность за результат.
* **C (Consulted)** — консультирует, даёт экспертную оценку.
* **I (Informed)** — информируется о ходе выполнения, но не участвует.

Она помогает избежать недопонимания в команде и чётко распределить обязанности.

**3. Проверка RACI-матрицы**

**По горизонтали (по задачам):**

* У **каждой задачи** должна быть хотя бы **одна R (исполнитель)** и **одна A (ответственный)**.
* ✅ Все строки содержат как R, так и A — проверка пройдена.

**По вертикали (по ролям):**

* Ни у одной роли не должно быть **только C или I** — хотя бы одна R или A нужна, чтобы роль была действительно задействована.
* ✅ Все роли имеют хотя бы одну R или A — проверка пройдена.