**План Тестирования: ИС Ресторан**

**Версия: Итерация 1**

**1.Участники**

Итерация выполняется командой, состоящей из разработчиков, тестеров. Разработчики в конечном счете подчиняются руководителю разработки (Дмитрий), а тестеры руководителю департамента качества (Тимур).

**2.Риски**

Итерация выполняется командой, состоящей из разработчиков, тестеров. Разработчики в конечном счете подчиняются руководителю разработки (Дмитрий), а тестеры руководителю департамента качества (Тимур).

**3.Стратегия тестирования**

Нужно помнить, что необходимо:

- Создать автоматизированные тестирования на основе историй, прежде чем начать кодирование, проверить новый код и интеграцию с текущей версией системы перед тем, как отметить завершение истории.

- Повторно тестировать каждый раз. когда что-то было изменено в результате предыдущих итераций, а также в результате текущей итерации, и применить регрессионное тестирование всего результата этой итерации перед встречей для презентации.

- Оценивать усилия и стоимость тестирования и разработки, чтобы соответствовать договоренности по данной итерации в начале итерации и возвращать в портфель все незавершенные элементы, которые не могут быть завершены, включая любой накопленный технический долг (ошибки), который не может быть разрешен в данной итерации.

- Использовать методики проектирования тестирования, наиболее подходящие к критериям приемки, учитывая то. что истории с более высокими рисками требуют более тщательною тестирования, чем истории с рисками ниже.

- Обеспечить и удостовериться, что покрытие тестированием операторов достигает, по крайней мере 90 % всего кода, а покрытие ветвления — 80 % для историй с высоким риском и 60 % для историй с низким риском.

- Перед интеграцией гарантировать, что в реализации истории отсутствуют невыясненные дефекты с уровнем серьезности 1 или 2.

- Определить тестирование с участием потребителя ATDD (приемку) в итерации с соглашением и участием потребителя/пользователя.

- Перед встречей для презентации протестировать результат итерации в формальных средах тестирования и презентации.

- Представлять покрытие элементов тестирования на ежедневных встречах, включая действия низкого уровня по Плану Тестирования и документирование рисков на доске обсуждения.

- Сохранять все сценарии тестирования в инструменте АВС таким образом, чтобы они. по мере необходимости. были доступны для повторного тестирования и регрессионного тестирования.

- В конце каждой итерации создавать краткий отчет о тестировании и размещать его на портале проекта.

**Расшифровка тестовых информационных полей**

## Расшифровка тестовых информационных полей:

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | ИС Ресторан |
| **Рабочая версия** | Тест 1.0 |
| **Имя тестирующего** | Лукьяненко Дмитрий Максимович |
| **Дата(ы) теста** | 02.04.2025- |
| **Тестовый пример №1** | Тестовый случай пользовательского интерфейса №1 — Тестирование функциональности, пользовательского интерфейса. |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | **Проверка Производительности и безопасности**. Требования пользователей(клиентской базы).Низкий уровень тестирования |
| **Заголовок/название теста** | Проведение ручного тестирования работы и отображения результатов вывода информации |
| **Краткое изложение теста** | Проверить работоспособность приложения |
| **Этапы теста** | 1. Запустить приложение 2. Нажать кнопку «GO» 3. Проверка вывода информации на главный экран 4. Закрыть приложение 5. Повторить запуск для закрепления результата |
| **Тестовые данные** | • Проверка кнопки "GO" на нажатие и отображение.  • Проверка выведенной информации.  • Проверка задержки при открытии. |
| **Ожидаемый результат** | Функционирование главной страницы- приложение должно показать работоспособность кнопок и вывода информации на главный экран |
| **Фактический результат** | - |
| **Предварительное условие** | Запуск программного продукта в Visual Studio и проверить через UnitTest. |
| **Постусловие** | Главная страница должна остаться в рабочем состоянии и не зависнуть. |
| **Статус**  *(Зачет/Незачет)* |  |
| **Примечания/комментарии** |  |

**Отчет о Завершении Тестирования**

**Приложение: ИС Ресторан**

Отчет о Завершении Тестирования по ГОСТ Р 56922—2016

**Н.1 Пример 1 — Корпорация «Lykirov»**

Корпорация «Lykirov» — большая организация, осуществляющая разработку и тестирование приложений.

Этот отчет доступен на портале проекта, и новейшая версия также вывешена в правом нижнем углу доски

истории в комнате разработки, где это разрабатывается и обновляется.

**1.Отчет тестирования для**: Новая система подписки (NSS) Vers.: Итерация 1.

2.**Покрытие**: Результат заключительной итерации NSS. включая результаты предыдущих итераций, в рамках подготовки к основной поставке потребителю (для использования).

**3.Риски**: Риск живых данных был устранен путем создания модельной базы данных с использованием исторических живых данных, «подчищенных» командой тестирования и потребителем.

**4.Результаты тестирования**: Заказчик принимает эту версию продукта на основе следующего:

* 16 историй пользователя были реализованы успешно, включая одну добавленную после последнего отчета о состоянии.
* 100 % покрытия операторов было достигнуто при технологическом тестировании с одной историей высокого риска, а для других было достигнуто в среднем 72 % покрытия операторов.
* Команда приняла портфель с четырьмя дефектами серьезности 1.
* Презентация была принята потребителем без обнаруженных проблем. Итеративные функции презентации взаимодействовали через интерфейс с «живыми» данными.
* Производительность итеративных функций, по мнению команды и потребителя, была приемлема.

**5.Новые, измененные и остаточные риски**: Защищенность системы может стать проблемой в будущих реализациях на основании информации о действиях, полученной от потребителей.

**6.Примечания для будущей работы из ретроспективы :**

* Итеративная команда ощущает потребность в новом члене команды для обработки возможных новых рисков, по которым ни у кого нет знаний.
* Дефекты серьезности 3, помещенные в портфель, нужно рассмотреть в следующей версии, чтобы
* уменьшить технический долг.
* Измененные живые данные работали хорошо и должны быть сохранены.
* Автоматизация тестирования и исследовательское тестирование работают, однако необходимо рассмотреть дополнительные методики проектирования тестирования, например, методики тестирования защищенности и комбинаторного тестирования.