## Руководство системного администратора по разработке программного продукта для учета заявок на ремонт бытовой техники

### Введение

Это руководство предназначено для системных администраторов, которые будут заниматься установкой, поддержкой и администрированием программного продукта для учета заявок на ремонт бытовой техники. Включает в себя требования к оборудованию, программному обеспечению, а также рекомендации по настройке и обслуживанию.

### 1. Требования к оборудованию

1.1. Минимальные системные требования:

- Процессор: 2 ядра, 2.0 GHz или выше

- Оперативная память: 4 ГБ (рекомендуется 8 ГБ)

- Накопитель: SSD объемом 100 ГБ или более

- Сетевая карта: 1 Гбит/с

- Резервное копирование: внешний накопитель или NAS

1.2. Рекомендуемая конфигурация:

- Процессор: 4 ядра, 3.0 GHz

- Оперативная память: 16 ГБ

- Накопитель: SSD объемом 250 ГБ

- Сетевая карта: 1 Гбит/с с возможностью подключения к VLAN

### 2. Программное обеспечение

2.1. Операционная система:

- Windows Server 2019 или 2022

- Ubuntu Server 20.04 или 22.04

2.2. Необходимые компоненты:

- СУБД: PostgreSQL или MySQL

- Веб-сервер: Nginx или Apache

- Язык программирования: Python, PHP или Java

- Фреймворк: Django, Laravel или Spring

### 3. Установка

3.1. Подготовка сервера:

- Установить операционную систему.

- Настроить сетевое подключение и проверить доступ к интернету.

- Установить обновления системы.

3.2. Установка необходимых пакетов:

- Установить и настроить СУБД (PostgreSQL/MySQL).

- Установить веб-сервер (Nginx/Apache).

- Установить интерпретатор языка программирования и фреймворк.

3.3. Развертывание приложения:

- Загрузить исходный код приложения из репозитория.

- Настроить виртуальные окружения (если требуется).

- Импортировать базу данных и настроить конфигурации.

### 4. Настройка безопасности

4.1. Настройка брандмауэра:

- Открыть необходимые порты (например, 80, 443 для HTTP/HTTPS, 5432 для PostgreSQL).

- Ограничить доступ к административным интерфейсам по IP-адресам.

4.2. Управление пользователями:

- Создать пользователей с ограниченными правами.

- Настроить аутентификацию и авторизацию.

4.3. Регулярные обновления:

- Настроить автоматическое обновление компонентов системы и приложений.

- Проводить регулярные проверки на уязвимости.

### 5. Мониторинг и резервное копирование

5.1. Мониторинг системы:

- Установить инструменты мониторинга (например, Zabbix, Prometheus).

- Настроить оповещения о сбоях и превышении пороговых значений.

5.2. Резервное копирование:

- Настроить регулярное резервное копирование базы данных и файлов.

- Проверять возможность восстановления данных из резервных копий.

### Заключение

Системный администратор должен обеспечить стабильную работу системы, следить за безопасностью и доступностью приложения, а также поддерживать документирование всех процессов.

---

## Руководство по сопровождению программного обеспечения для учета заявок на ремонт бытовой техники

### Введение

Это руководство предназначено для команды сопровождения, ответственной за поддержку и развитие программного продукта для учета заявок на ремонт бытовой техники. Включает в себя информацию о функционале, документации, процессе отчетности и обновлениях.

### 1. Описание функционала

1.1. Основные функции:

- Регистрация и отслеживание заявок на ремонт.

- Уведомления пользователю о статусе ремонта.

- Система оценки работы сервисного центра.

- Отчеты по количеству и типам заявок.

1.2. Целевые пользователи:

- Клиенты, обращающиеся с заявками.

- Служба технической поддержки.

- Администраторы системы.

### 2. Документация

2.1. Пользовательская документация:

- Инструкции по регистрации в системе.

- Руководство по созданию и отслеживанию заявок.

- Описание системы уведомлений.

2.2. Техническая документация:

- Архитектура приложения.

- API для интеграции с другими системами.

- Инструкции по установке и настройке среды разработки.

### 3. Процесс отчетности

3.1. Общение с пользователями:

- Создание канала обратной связи: email, телефон, чат.

- Регулярные опросы пользователей для сбора обратной связи.

- Ведение журнала обращений и решений.

3.2. Извинения об ошибках:

- Регистрация ошибок в системе управления задачами (например, Jira).

- Установка приоритетов для быстрого реагирования на критические ошибки.

### 4. Обновления

4.1. Периодические обновления:

- Регулярные обновления с улучшениями и исправлениями (не реже одного раза в месяц).

- Тестирование обновлений на тестовой среде перед развертыванием.

4.2. Планирование новых функций:

- Определение и приоритизация новых функций на основе отзывов пользователей.

- Проведение встреч для обсуждения плана развития программы.

### Заключение

Команда сопровождения должна обеспечивать постоянное улучшение системы, внимание к пользователям и быстрое реагирование на проблемы. Важно поддерживать актуальность документации и следить за развитием технологий, чтобы адаптировать систему под новые требования.

---

Эти руководства являются основой для работы с программным продуктом и его сопровождения, и могут быть адаптированы в зависимости от конкретных условий и потребностей организации.

## Базовый план по обеспечению качества (QA) для системы учета заявок на ремонт бытовой техники

### Введение

Обеспечение качества (QA) является неотъемлемой частью разработки программного обеспечения, особенно для систем, которые будут использоваться в реальной среде. Данный план описывает стратегию тестирования и обеспечит высокое качество программы для учета заявок на ремонт бытовой техники.

### 1. Цели QA

- Обеспечить высокое качество программного продукта.

- Выявить и устранить дефекты на ранних этапах.

- Минимизировать риски, связанные с развертыванием приложения.

- Повысить удовлетворенность пользователей.

### 2. Объем работы

- Автоматизированное тестирование.

- Ручное тестирование.

- Удобство интерфейса (UI/UX).

- Тестирование безопасности.

- Нагрузочное и производительное тестирование.

### 3. Этапы тестирования

#### 3.1. Планирование тестирования

- Определение объема тестирования и его целей.

- Разработка тестовой документации (план тестирования, тест-кейсы, тестовые сценарии).

- Определение ресурсов и инструментов для тестирования.

#### 3.2. Подготовка тестовой среды

- Настройка тестовой среды, которая будет идентична рабочей.

- Установка всех необходимых программных компонентов, включая СУБД, веб-сервер и само приложение.

#### 3.3. Разработка тест-кейсов

- Создание функциональных тест-кейсов на основе спецификаций требований системы.

- Разработка тест-кейсов для негативного тестирования (например, некорректные входные данные).

- Подготовка тест-кейсов для проверки производительности и нагрузки.

#### 3.4. Автоматизированное тестирование

- Выбор фреймворков для автоматизации (например, Selenium для UI, JUnit/TestNG для функциональных тестов).

- Разработка скриптов для автоматического тестирования основных функций приложения.

- Настройка непрерывной интеграции (CI) для автоматического запуска тестов при каждом изменении кода.

#### 3.5. Проведение тестирования

- Выполнение ручного тестирования согласно разработанным тест-кейсам.

- Запуск автоматических тестов и анализ их результатов.

- Проведение тестов на совместимость, производительность и безопасность.

### 4. Управление дефектами

- Использование системы отслеживания ошибок (например, Jira, Bugzilla) для регистрации и управления дефектами.

- Регулярная отчетность о найденных дефектах и их статусе.

- Планирование работ по устранению обнаруженных проблем.

### 5. Отчетность и анализ

- Подготовка отчетов о результатах тестирования: количество пройденных и не пройденных тестов, найденные дефекты, анализ покрытия тестами.

- Оценка качества приложения на основе тестирования.

- Рекомендации по улучшению продукта на основе результатов тестирования.

### 6. Обратная связь и улучшение процесса

- Сбор отзывов от тестировщиков и конечных пользователей.

- Анализ производительности тестирования и выявление возможных улучшений.

- Обновление плана по обеспечению качества на основе полученного опыта и рекомендаций.

### Заключение

Разработка плана по обеспечению качества для системы учета заявок на ремонт бытовой техники поможет организовать процесс тестирования, выявить дефекты и обеспечить высокое качество готового продукта. Регулярная оценка и улучшение процессов QA будут способствовать разработке более эффективных и надежных систем.