**Лабораторная работа 2**

**Организационные меры по внедрению программного обеспечения**

*Задания*

1Задание

Для обсуждения перечня работ по созданию программного обеспечения для проекта внедрения программного комплекса "Выдача рассрочки в магазине" рассмотрим модель Жизненного Цикла Программного Обеспечения (ЖЦПО) в соответствии с классическим подходом:

1. Анализ и планирование:

- Идентификация и анализ требований для программного комплекса "Выдача рассрочки в магазине".

- Постановка целей проекта и формирование бизнес-кейса.

- Разработка плана проекта, включая оценку ресурсов и расписания работ.

2. Проектирование:

- Создание архитектуры программного комплекса "Выдача рассрочки в магазине" с учетом функциональных и нефункциональных требований.

- Разработка дизайна интерфейса пользователя и пользовательских сценариев.

- Определение требований к базе данных и способов взаимодействия с другими системами.

3. Разработка:

- Реализация программного кода на основе определенной архитектуры и дизайна.

- Создание модулей для работы с базой данных, бизнес-логики, интерфейсом пользователя и другими компонентами.

- Тестирование и отладка разрабатываемых модулей, включая модульные и интеграционные тесты.

4. Тестирование и верификация:

- Планирование и проведение тестирования программного комплекса, включая функциональное тестирование, тестирование производительности, нагрузочное тестирование и тестирование безопасности.

- Проверка соответствия выполненных работ требованиям и спецификациям проекта.

- Вовлечение пользователей и получение обратной связи для уточнения и улучшения продукта.

5. Внедрение и сопровождение:

- Подготовка и проведение внедрения программного комплекса "Выдача рассрочки в магазине" в среде пользователя.

- Обеспечение постоянной поддержки и сопровождения программного комплекса, включая исправление ошибок, устранение неполадок и модификации в соответствии с требованиями пользователей.

Отметим, что представленный перечень работ является обобщенным и может дополняться в зависимости от специфических требований и контекста проекта. Важно также учитывать практики и стандарты в области разработки программного обеспечения, а также оставаться внимательным к потребностям пользователей и изменениям в бизнес-среде.

2Задание

Распределение работы между участниками группы может зависеть от их ролей, навыков и опыта. Вот предполагаемое распределение работ для проекта внедрения программного комплекса "Выдача рассрочки в магазине" с учетом различных этапов:

1. Анализ и планирование:

- Анализ требований: Группа аналитиков или бизнес-аналитиков.

- Планирование проекта: Проектный менеджер или технический лидер.

2. Проектирование:

- Архитектура программного комплекса: Технический лидер или архитектор программного обеспечения.

- Дизайн интерфейса пользователя: Графический дизайнер или разработчик интерфейса.

3. Разработка:

- Разработка модуля базы данных: Разработчик баз данных.

- Разработка бизнес-логики: Команда разработчиков с опытом в разработке бизнес-логики.

- Разработка интерфейса пользователя: Команда разработчиков с опытом в разработке пользовательского интерфейса.

4. Тестирование и верификация:

- Тестирование функциональности: Команда тестировщиков.

- Тестирование производительности и нагрузки: Специалисты по тестированию производительности.

- Тестирование безопасности: Специалисты по тестированию безопасности.

5. Внедрение и сопровождение:

- Внедрение программного комплекса: Команда разработчиков или инженеров для поддержки внедрения.

- Сопровождение и поддержка: Команда технической поддержки или сопровождения с опытом в поддержке программных продуктов.

Ключевым аспектом распределения работ является коммуникация и согласование между участниками группы. Важно обеспечить баланс и гармоничное сотрудничество между различными ролями и участниками проекта.

Обратите внимание, что это только предположительное распределение работ, и реальное распределение может варьироваться в зависимости от особенностей проекта и командных ролей. Не забывайте учитывать индивидуальные навыки, предпочтения и опыт участников группы при распределении работ.

3Задание

План работ по созданию программного продукта "Выдача рассрочки в магазине"

1. Анализ и планирование

- Идентификация и анализ требований для программного продукта.

- Постановка целей проекта и формирование бизнес-кейса.

- Определение ожидаемых результатов и области применения продукта.

- Разработка плана проекта, включая оценку ресурсов, расписание и бюджет.

2. Проектирование

- Создание архитектуры программного продукта, определение основных компонентов и связей между ними.

- Разработка дизайна пользовательского интерфейса, включая визуальное оформление и использование наилучших практик юзабилити.

- Определение структуры базы данных и модели данных для хранения информации о рассрочке и клиентах.

3. Разработка

- Реализация компонентов программного продукта на основе разработанной архитектуры и дизайна.

- Разработка функциональности, связанной с оформлением рассрочки, проверкой кредитного рейтинга клиента и расчетом суммы платежей.

- Создание модулей для взаимодействия с базой данных и интеграции с системами учета.

4. Тестирование и качество

- Проведение модульного тестирования для каждого компонента и модуля.

- Интеграционное тестирование для проверки взаимодействия между компонентами.

- Проведение функционального и пользовательского тестирования для проверки соответствия требованиям и проверки юзабилити.

- Отладка и исправление обнаруженных ошибок.

5. Внедрение

- Подготовка окружения для развертывания программного продукта.

- Установка и настройка программы на сервере или в сети магазина, обеспечение стабильной работы программы.

- Обеспечение поддержки пользователей и участников процесса оформления рассрочки, проведение обучения персонала.

6. Сопровождение и поддержка

- Предоставление технической поддержки пользователей, решение возникающих проблем и вопросов.

- Регулярное обновление программного продукта в соответствии с требованиями и отзывами пользователей.

- Проведение аудита безопасности и обеспечение защиты данных клиентов.

Это примерный план работ, который может быть дополнен и настроен для конкретных требований и условий проекта "Выдача рассрочки в магазине".

План работ в электронном виде можно оформить в виде таблицы или документа, например в формате Microsoft Excel или Google Sheets. Такой документ может содержать столбцы с описанием работ, сроками выполнения и ответственными лицами. Кроме того, можно использовать графики Ганта или другие инструменты для визуализации и управления планом работ.

*Контрольные вопросы*

1. Модели жизненного цикла программного обеспечения (ЖЦПО) представляют различные подходы к организации и управлению разработкой программного обеспечения. Ниже перечислены некоторые популярные модели ЖЦПО:

- Waterfall (Водопадная модель)

- Spiral (Спиральная модель)

- Iterative and Incremental (Итеративная и инкрементная модель)

- Agile (Гибкая модель)

- DevOps (Разработка и эксплуатация)

2. Ниже приведена сравнительная характеристика каждой модели:

- Waterfall (Водопадная):

- Линейная последовательность этапов разработки

- Водопадный, некорректируемый процесс

- Требования формализованы и фиксированы заранее

- Ограниченные возможности реагирования на изменения

- Рекомендуется для проектов с четко определенными требованиями и минимальной вероятностью изменений

- Spiral (Спиральная):

- Итеративный и инкрементный подход

- Управление рисками конкретно в каждом цикле

- Оценка и анализ требований в начале каждого цикла

- Подходит для проектов со значительными рисками, изменяющимися требованиями и требованиями быстрой доставки

- Iterative and Incremental (Итеративная и инкрементная):

- Последовательное повторение циклов разработки

- Упор на постепенное приращение функциональности

- Раннее предоставление рабочих версий

- Возможность внесения изменений в процессе разработки

- Применима к проектам с определенными, но гибкими требованиями

- Agile (Гибкая):

- Итеративный подход с акцентом на гибкость и сотрудничество

- Высокая вовлеченность заказчика в процесс разработки

- Прототипирование и постепенное развитие системы

- Возможность быстрой адаптации к изменениям

- Применима в проектах, где требуется большая гибкость, взаимодействие с заказчиком и быстрая поставка ценности

- DevOps (Разработка и эксплуатация):

- Интеграция разработки и эксплуатации в непрерывный процесс

- Автоматизация тестирования, развертывания и мониторинга

- Быстрое внедрение изменений в продукцию

- Улучшение коммуникации и сотрудничества между разработчиками и эксплуатационными командами

Каждая модель ЖЦПО имеет свои преимущества и ограничения, и выбор модели зависит от особенностей проекта, его требований и контекста. Это лишь общая характеристика каждой модели, и подходы могут быть различны в зависимости от конкретной ситуации.