Отчёт №2

**Организационные меры по внедрению программного обеспечения**

**Модель жизненого цикла ПП «Спиральная модель»**

**Задание**

**План работ по созданию программного продукта (ПП):**

**«Выдача кредитов в банке»**

1. **Определение требований:**

Провести встречи с заказчиком компании “Сбербанк” выяснить функциональные и технические требования ПП

Документировать полученные требования.

1. **Анализ рынка и конкурентов:**

Существующие аналогии ПП для выдач кредитов в банке:

**Finastra Fusion Loan IQ:**

Особенности:

Мощная система учета кредитов и управления рисками.

Многофункциональный и гибкий интерфейс.

Недостатки:

Высокая стоимость внедрения и обслуживания.

Требует опытных специалистов для настройки и поддержки.

**Temenos Transact:**

Особенности:

Широкий спектр функций для управления кредитами.

Поддержка цифровых каналов обслуживания клиентов.

Недостатки:

Высокие затраты на внедрение.

Возможны сложности с интеграцией с существующими системами банка.

**Nucleus Software FinnOne:**

Особенности:

Интегрированная система для учета и управления кредитами.

Аналитика и мониторинг кредитных рисков.

Недостатки:

Сложность настройки и внедрения.

Высокие требования к аппаратному обеспечению.

**Ellie Mae Encompass:**

Особенности:

Автоматизация процесса выдачи ипотечных кредитов.

Интеграция с партнерами и сторонними сервисами.

Недостатки:

Особенно подходит для ипотечных кредитов и может быть менее подходящим для других видов кредитов.

**Experian PowerCurve:**

Особенности:

Анализ кредитоспособности и оптимизация процессов.

Мониторинг кредитных рисков.

Недостатки:

Необходимость в качественных данных для точного анализа.

**CreditTrack by Equifax:**

Особенности:

Мониторинг кредитных данных и рисков.

Интеграция с кредитными бюро и сторонними источниками данных.

Недостатки:

Могут возникнуть конфиденциальность и безопасность данных.

1. **Проектирование:**

Разработать архитектуру ПП.

Создать дизайн интерфейса пользователя.

Определить технологический стек и инструменты разработки.

**Backend:**

Язык программирования: Java для надежности и масштабируемости.

Фреймворк: Spring (для Java) для облегчения разработки.

СУБД: Microsoft Access или Oracle для хранения и управления данными о клиентах и кредитах.

**Frontend:**

Язык: JavaScript с использованием фреймворка React или Angular для построения интерфейса.

Визуальный дизайн: Использование библиотеки компонентов, например, Material-UI.

**API и взаимодействие:**

RESTful API: Для обмена данными между клиентами и сервером.

Swagger/OpenAPI: Документация API для упрощения интеграции.

**Обеспечение безопасности:**

HTTPS: Защита передачи данных между клиентом и сервером.

**Тестирование:**

JUnit/TestNG: Для юнит-тестирования бэкенда.

Jest/Mocha: Для тестирования фронтенда.

Appium или XCTest: Для тестирования мобильных приложений.

Инструменты управления проектом:

Git: Для контроля версий кода.

JIRA или Trello: Для управления задачами и проектом.

**Контейнеризация и развертывание:**

Docker: Для контейнеризации приложения.

Kubernetes: Для оркестрации контейнеров и обеспечения масштабируемости.

**Мониторинг и логирование:**

ELK Stack (Elasticsearch, Logstash, Kibana): Для централизованного логирования.

Prometheus/Grafana: Для мониторинга системы.

1. **Разработка:**

Написать код основной функциональности.

Провести тестирование промежуточных версий.

1. **Тестирование:**

Провести функциональное тестирование для проверки соответствия требованиям.

Выявить и устранить ошибки.

1. **Доработка и оптимизация:**

Улучшить производительность и стабильность программы.

Внедрить корректировки и дополнения, если необходимо.

1. **Документирование:**

Подготовить техническую документацию по ПП.

Создать руководство пользователя.

1. **Тестирование завершенной версии:**

Провести финальное тестирование перед выпуском.

1. **Релиз и внедрение:**

Подготовить программу к выпуску.

Разработать стратегию маркетинга и внедрения.

1. **Поддержка и обновление:**

Обеспечить техническую поддержку пользователям.

Регулярно выпускать обновления для улучшения и дополнения функционала.

Контрольные задания:

1. **Перечислите модели ЖЦПО**

Модель жизненного цикла программного обеспечения (ЖЦПО) представляет собой процесс разработки и управления программным продуктом от начальной концепции до завершения. Существует несколько моделей ЖЦПО, каждая из которых описывает этот процесс по-своему. Вот некоторые из наиболее распространенных моделей ЖЦПО:

**Водопадная модель (Waterfall Model):** Линейная модель, где каждая фаза разработки следует за предыдущей, начиная с анализа и проходя через проектирование, реализацию, тестирование и поддержку.

**Итеративная модель (Iterative Model):** Разработка программы выполняется в циклах, называемых итерациями. Каждая итерация добавляет новый функционал и проходит через все этапы разработки.

**Инкрементальная модель (Incremental Model):** Программа разрабатывается как набор небольших инкрементов или модулей, которые добавляются по мере развития проекта.

**Спиральная модель (Spiral Model):** Модель, объединяющая элементы водопадной модели и итеративной разработки. Процесс построен на циклах, с каждым циклом добавляются новые функции.

**Модель быстрой разработки приложений (RAD - Rapid Application Development):** Уделяет большое внимание быстрому прототипированию и вовлечению заказчика в процесс разработки.

**Модель спецификаций на языке формальных методов (Formal Methods Specification Model):** Основывается на математических методах для формальной верификации и спецификации программ.

**DevOps и CI/CD (Continuous Integration/Continuous Deployment):** Эти модели сосредотачиваются на автоматизации процесса разработки, тестирования и развертывания ПО, обеспечивая непрерывное улучшение и доставку.

Каждая из этих моделей имеет свои преимущества и недостатки, и выбор модели зависит от конкретных требований и характеристик проекта.нтрольные вопросы:

1. **Дайте сравнительную характеристику каждому виду.**

Стравнительная характеристика различных видов моделей жизненного цикла программного обеспечения (ЖЦПО):

**Водопадная модель (Waterfall Model):**

**Особенности:** Линейный подход, последовательные фазы.

**Преимущества:** Простота управления, хорошо подходит для небольших проектов с четкими требованиями.

**Недостатки:** Не подходит для проектов с изменяющимися требованиями, риски высоки при отсутствии реальных результатов до завершения.

**Итеративная модель (Iterative Model):**

**Особенности:** Разработка в итерациях, повторное проектирование и улучшение на каждой итерации.

**Преимущества**: Возможность быстрого реагирования на изменения требований, улучшение качества на каждой итерации.

**Недостатки**: Может потребовать дополнительного времени и ресурсов.

**Инкрементальная модель (Incremental Model):**

**Особенности**: Разработка по частям, поочередное добавление инкрементов.

**Преимущества**: Возможность быстрого выпуска базового функционала, удобно для расширения.

**Недостатки**: Требует внимательного планирования и управления версиями.

**Спиральная модель (Spiral Model):**

**Особенности**: Комбинирует элементы водопада и итераций с учетом рисков.

**Преимущества**: Учитывает риски, позволяет быстро реагировать на изменения.

**Недостатки**: Сложность управления, может потребовать больше времени.

**Модель быстрой разработки приложений (RAD - Rapid Application Development):**

**Особенности**: Основана на быстром прототипировании и вовлечении заказчика.

**Преимущества**: Сокращение времени разработки, высокая степень участия заказчика.

**Недостатки**: Не всегда подходит для сложных проектов, требует активного участия заказчика.

**Модель спецификаций на языке формальных методов (Formal Methods Specification Model):**

**Особенности**: Использует математические методы для спецификации и верификации.

**Преимущества**: Высокая точность и надежность.

**Недостатки**: Сложность в реализации, не всегда подходит для всех типов ПО.

**DevOps и CI/CD (Continuous Integration/Continuous Deployment):**

**Особенности**: Автоматизированный процесс интеграции, тестирования и развертывания.

**Преимущества**: Быстрая поставка, непрерывное улучшение.

**Недостатки**: Требует инфраструктуры и автоматизации.

Техническое задание (ТЗ) для разработки программного обеспечения (ПО) является важным документом, который описывает требования и спецификации проекта. Вот общий шаблон технического задания для разработки ПО:

**Техническое задание на разработку нашего программного обеспечения**

**1. Введение**

1.1 Цель проекта

**Автоматизация и оптимизация процесса выдачи кредитов:** Целью может быть сокращение времени, необходимого для рассмотрения заявок и выдачи кредитов, что повышает удовлетворенность клиентов и уменьшает операционные издержки.

**Улучшение качества принятия решений:** С использованием аналитики и машинного обучения можно стремиться к улучшению точности оценки кредитоспособности клиентов, снижая риски невозврата кредитов.

**Расширение клиентской базы:** Проект может нацелиться на привлечение новых клиентов через более удобные и привлекательные условия кредитования.

**Соблюдение законодательных требований**: Гарантировать соблюдение всех законов и нормативов, связанных с выдачей кредитов, может быть одной из основных целей.

**Увеличение доходности банка**: Целью может быть увеличение прибыли банка за счет увеличения объема выданных кредитов и снижения невозвратов.

**Анализ и оптимизация производительности портфеля кредитов:** Мониторинг и анализ производительности кредитного портфеля с целью оптимизации его структуры и снижения рисков.

**Внедрение новых технологий:** Использование современных технологий и инновационных подходов в выдаче кредитов может быть целью с целью улучшения конкурентоспособности банка.

1.2 Контекст проекта

**Банк:** Сбербанк, полное наименование которого - ПАО Сбербанк России, является крупнейшим банком в России и одним из крупнейших финансовых институтов в мире. Ниже предоставлена основная информация о Сбербанке:

**Название**: ПАО Сбербанк России (Сбербанк)

**Описание**: Сбербанк является универсальным банком, предоставляющим широкий спектр банковских услуг для частных лиц, предприятий и организаций. Банк был основан в 1841 году и имеет богатую историю в области финансовых услуг.

**Размер**: Сбербанк является крупнейшим банком в России и одним из крупнейших банков в мире по общей активной базе, рыночной капитализации и другим финансовым показателям.

**Региональное и мировое присутствие**: Сбербанк имеет широкую сеть филиалов и офисов в России и также расширил свою деятельность за границу. Он активно развивает свои операции в различных странах и регионах мира, включая страны СНГ и внешние рынки.

**Виды банковских услуг**: Сбербанк предоставляет множество видов банковских услуг, включая:

**Розничное банковское обслуживание**: Вклады, сберегательные счета, кредитные карты, ипотечные кредиты и другие продукты для частных клиентов.

**Корпоративное банковское обслуживание**: Финансовые услуги для предприятий и организаций, включая кредитование, управление ликвидностью, трейдинг и другие услуги.

**Инвестиционные услуги**: Управление активами, инвестиционное консультирование, брокерские услуги и другие финансовые услуги для инвесторов.

**Интернет- и мобильное банковское обслуживание**: Сбербанк предоставляет современные технологические решения для онлайн-банкинга и мобильных приложений для удобства клиентов.

**Цели и задачи проекта:**

**Автоматизация процессов**: ПП может помочь банку автоматизировать весь процесс выдачи кредитов, что сократит ручной труд и ускорит принятие решений.

**Улучшение точности принятия решений**: Благодаря анализу больших объемов данных, ПП может помочь банку более точно оценивать кредитоспособность заявителей и риски.

**Увеличение скорости обработки заявок**: Эффективное ПП позволит банку обрабатывать больше заявок в короткие сроки, что может привести к увеличению выдачи кредитов и увеличению прибыли.

**Снижение рисков невозврата**: Путем более детального анализа данных и оценки рисков, ПП помогает банку уменьшить вероятность невозврата кредитов.

**Улучшение клиентского опыта**: Если ПП позволяет быстро и удобно оформить кредит, это может привести к повышению удовлетворенности клиентов.

**Соблюдение регуляторных требований**: ПП может помочь банку соблюдать законодательные и регуляторные требования в области выдачи кредитов.

**Сокращение операционных затрат**: Автоматизация процессов с помощью ПП может снизить операционные расходы банка.

**Целевая аудитория**:

**Физические лица:**

Заемщики, нуждающиеся в кредите для покупки недвижимости, автомобиля или других долгосрочных потребностей.

Люди, ищущие краткосрочное финансирование, такое как кредиты на потребительские нужды.

Клиенты, имеющие хорошую кредитную историю и способных обеспечить погашение кредита.

**Малый и средний бизнес:**

Предприниматели и владельцы бизнесов, которые нуждаются в финансировании для развития своего бизнеса, закупки оборудования и т. д.

Собственники малых компаний, ищущие рабочий капитал или кредиты для расширения.

**Сегменты рынка:**

Определенные сегменты, такие как студенты, молодые семьи, пенсионеры, которые могут иметь специфические потребности и условия для кредитования.

**Клиенты с разными целями:**

Заемщики, ищущие ипотечные кредиты.

Люди, нуждающиеся в кредитах с низкой процентной ставкой для образования.

Люди, нуждающиеся в кредитах для медицинских расходов.

**Текущие проблемы**:

**Недостаточная точность алгоритмов:** Алгоритмы оценки кредитоспособности могут быть недостаточно точными, что может привести к неверным решениям о выдаче кредита.

**Недостаточная защита от мошенничества:** ПП может быть уязвимым перед различными видами мошенничества, например, подделкой личных данных или финансовых документов.

**Неполные или некорректные данные**: Если входные данные для ПП неполные, устаревшие или некорректные, это может исказить результаты оценки кредитоспособности.

**Отсутствие человеческого фактора**: ПП не всегда способен учитывать контекст и человеческий опыт, что может привести к неправильным решениям в нестандартных ситуациях.

**Проблемы с безопасностью данных**: Хранение и обработка чувствительных данных о клиентах может привести к угрозам безопасности, если ПП не обладает должным уровнем защиты.

**Недостаточная прозрачность алгоритмов**: Некоторые ПП работают на основе сложных алгоритмов машинного обучения, которые могут быть непрозрачными и труднопонимаемыми, что затрудняет объяснение решений клиентам и регуляторам.

**Технологическая среда**:

Технологическая среда: Автоматизированные банковские системы (Core Banking Systems): Они обеспечивают основные банковские функции, такие как управление счетами, операции с кредитами и вкладами, обработка транзакций и т.д.

Системы управления клиентскими отношениями (CRM): Они используются для управления взаимодействием с клиентами, отслеживания контактов и анализа данных для улучшения обслуживания клиентов.

Системы анализа данных и отчетности: Они помогают в анализе финансовых данных, прогноз

**Бюджет и ресурсы**: Бюджет проекта: ???

Ресурсы:

Проектная команда:

Проектный менеджер

Аналитик данных

Разработчик программного обеспечения

Эксперт по кредитованию

Оборудование: Серверы для хранения данных и тестирования новой системы.

Программное обеспечение: Специализированные программы для оценки кредитоспособности и автоматизации процессов.

Дополнительные ресурсы по необходимости.

**Сроки и график**:

Фаза 1: Анализ и проектирование (3 месяца)

Фаза 2: Разработка и тестирование (6 месяцев)

Фаза 3: Внедрение и обучение персонала (2 месяца)

Фаза 4: Мониторинг и оптимизация (постоянно)

2. Требования к функциональности

2.1 Основные функции

**Заполнение заявки на кредит**: Позволяет клиентам банка подать онлайн-заявку на кредит, включая необходимую информацию о себе и запрашиваемой сумме.

**Анализ кредитоспособности**: Осуществляет оценку кредитоспособности заявителей на основе предоставленных данных, кредитной истории, доходов и других факторов.

**Принятие решения по выдаче кредита**: Автоматически или с участием сотрудников банка принимает решение о выдаче кредита и определяет условия (сумма, срок, процентная ставка).

**Документооборот:** Управляет документами, связанными с процессом выдачи кредитов, включая подписание соглашений и согласование условий.

**Управление кредитным портфелем**: Позволяет банку следить за состоянием кредитных счетов клиентов, мониторить платежи и выявлять просрочки.

**Генерация документов**: Создает необходимые документы для клиентов и банка, такие как договоры, расписки и платежные графики.

**Уведомления и согласование**: Отправляет уведомления клиентам о решении по кредиту и согласовании условий.

**Мониторинг и анализ**: Предоставляет аналитические отчеты и дашборды для оценки эффективности процесса выдачи кредитов и рисков.

**Интеграция с внешними системами**: Взаимодействует с внешними системами, такими как кредитные бюро и системы безопасности для проверки информации о клиентах.

**Архивирование данных**: Хранит и обеспечивает доступ к архивным данным о кредитах и клиентах для внутренних и регуляторных нужд.

3. Требования к интерфейсу

**Понятность и удобство**: Интерфейс должен быть интуитивно понятным для пользователя, даже без специальных навыков. Используйте понятные и легко узнаваемые иконки и метки.

**Адаптивность**: Интерфейс должен адаптироваться к разным типам устройств (компьютеры, планшеты, мобильные телефоны) и разрешениям экранов.

**Безопасность**: Обеспечьте высокий уровень безопасности, включая защиту данных клиентов и возможность двухфакторной аутентификации.

**Доступность**: Сделайте интерфейс доступным для всех категорий пользователей, включая людей с ограниченными возможностями (например, поддержка скринридеров).

**Эффективность**: Обеспечьте возможность быстрого заполнения и отправки заявки на кредит, минимизируя лишние шаги.

**Контактная информация**: Включите контактные данные банка, чтобы пользователи могли задать вопросы или получить поддержку.

**Обратная связь**: Предусмотрите возможность для пользователей отправлять обратную связь и жалобы на сервис.

**Документация**: Предоставьте пользователям подробную документацию по использованию ПП.

**Согласие на обработку данных**: Включите механизм получения согласия пользователя на обработку его личных данных в соответствии с GDPR и другими применимыми законами.

5. Требования к безопасности

**Аутентификация и авторизация**: Обеспечьте механизмы аутентификации пользователей и контроля доступа на основе ролей, чтобы предотвратить несанкционированный доступ.

**Защита данных**: Храните и передавайте данные пользователей в зашифрованном виде (SSL/TLS), чтобы защитить их от утечки.

**Защита от SQL-инъекций**: Проверяйте ввод данных пользователей и используйте параметризованные запросы, чтобы предотвратить SQL-инъекции.

**Защита от межсайтовых атак (CSRF):** Реализуйте механизмы проверки CSRF-токенов для защиты от атак, связанных с выполнением действий от имени аутентифицированных пользователей без их ведома.

**Защита от межсайтового скриптинга (XSS):** Валидируйте и фильтруйте ввод данных пользователя для предотвращения внедрения вредоносных скриптов.

**Системные обновления**: Регулярно обновляйте операционную систему, базы данных и все зависимости, чтобы устранить уязвимости.

**Мониторинг и аудит**: Включите системы мониторинга, аудита и журналирования, чтобы отслеживать необычную активность и анализировать инциденты безопасности.

**Обработка ошибок:** Управляйте ошибками без раскрытия конфиденциальной информации и предоставляйте пользователю информацию о действиях для восстановления.

**Бэкапы и восстановление**: Регулярно создавайте резервные копии данных и имейте процедуры восстановления в случае сбоев или атак.

**Соответствие стандартам**: Убедитесь, что ПП соответствует стандартам безопасности, таким как OWASP Top Ten и другим отраслевым рекомендациям.

**Обновление паролей**: Заставляйте пользователей регулярно менять пароли и требуйте сложные пароли.

**Блокировка аккаунтов**: Реализуйте механизмы блокировки аккаунтов после нескольких неудачных попыток входа.

**Обучение сотрудников**: Обучите сотрудников компании мерам безопасности и реакции на инциденты.

**Тестирование на проникновение**: Проводите регулярные тесты на проникновение для выявления уязвимостей.

6. Требования к поддержке и обслуживанию

**Техническая поддержка**: Предоставьте механизмы для получения технической поддержки, включая контактные данные, чаты, электронную почту или телефон.

**Документация**: Разработайте подробную документацию по установке, настройке и использованию ПП, доступную для пользователей и администраторов.

**Обновления и патчи**: Регулярно выпускайте обновления ПП, включая исправления уязвимостей и новые функции.

**Совместимость**: Убедитесь, что ПП совместим с разными операционными системами и браузерами, и предоставьте рекомендации по совместимости.

**Мониторинг состояния системы**: Разработайте механизмы мониторинга для отслеживания состояния ПП и быстрого реагирования на проблемы.

**Резервное копирование и восстановление**: Предоставьте инструкции и инструменты для создания резервных копий данных и восстановления системы после сбоев.

**Управление запросами и инцидентами**: Используйте систему для учета и управления запросами и инцидентами поддержки, чтобы обеспечить структурированное взаимодействие с клиентами.

**Обратная связь**: Собирайте обратную связь от пользователей о качестве поддержки и используйте ее для улучшения сервиса.

**Интеграция с мониторингом и аналитикой**: Интегрируйте ПП с системами мониторинга и аналитики, чтобы получать данные о использовании и производительности.

**Секретные ключи и доступ к серверам**: Управляйте секретными ключами и доступом к серверам ПП, чтобы предотвратить несанкционированный доступ.

**Создание тестовых сре**д: Разработайте среды для тестирования обновлений и патчей перед их внедрением в продуктивное окружение.

**Обновляемая документация**: Постоянно обновляйте документацию, отражая изменения и новые функции.

7. Требования к документации

**Техническая документация:**

Техническое задание (ТЗ) на разработку программного обеспечения или системы, включая описание функциональности и требований к системе.

Диаграммы баз данных, схемы архитектуры системы и другие технические спецификации.

Код программного обеспечения, если он разрабатывается в рамках проекта.

**Руководства пользователя:**

Руководство пользователя по использованию системы выдачи кредитов.

Инструкции по работе с веб-интерфейсом для сотрудников банка, обслуживающих кредитные операции.

Руководство для администраторов системы (если необходимо).

**Прочие документы:**

Бизнес-процессы и инструкции по выдаче и управлению кредитами в банке.

План тестирования, отчеты о тестировании и результаты испытаний.

Документация по безопасности информации и политика обработки данных клиентов.

Любые другие релевантные документы, связанные с процессом выдачи кредитов и его автоматизацией.

8. Заключение

**Утверждение и согласование технического задания (ТЗ) в проекте по выдаче кредитов в банке обычно проходит через следующие этапы и ответственные лица:**

**Инициатор проекта**: Этим может быть высший руководитель банка или отдела, ответственного за кредитование. Он инициирует проект и определяет цели и общие требования.

**Заказчик проекта**: Заказчиком обычно выступает тот же высший руководитель или управляющий проектом. Он отвечает за утверждение и согласование ТЗ, а также за бюджет проекта.

**Отдел, занимающийся кредитованием**: Представители отдела кредитования в банке должны участвовать в согласовании ТЗ, так как они обладают экспертизой в данной области.

**Отдел информационных технологий (ИТ):** Специалисты ИТ обязаны принимать участие в согласовании ТЗ, так как они будут разрабатывать и поддерживать техническую часть системы.

**Бизнес-аналитики**: Они помогают перевести требования бизнеса в технические требования и участвуют в согласовании ТЗ.

**Юридический отдел**: Он может быть вовлечен в утверждение ТЗ, чтобы гарантировать, что проект соответствует законодательству и правилам банка.

**Внутренние и внешние эксперты**: При необходимости могут быть привлечены эксперты или консультанты, имеющие опыт в области кредитования.

11. Подписи

Подписать документ и указать дату утверждения.

**Техническое задание служит основой для разработки ПО и является ключевым документом, определяющим требования и ожидания заказчика.**

**Контрольные задания:**

1. **Назовите возможные цели и задачи внедрения ПО.**

Внедрение программного обеспечения (ПО) может иметь разнообразные цели и задачи в зависимости от конкретного контекста и потребностей организации или пользователя. Ниже приведены некоторые возможные цели и задачи внедрения ПО:

**Цели:**

Автоматизация бизнес-процессов: Улучшение эффективности и производительности организации через автоматизацию рутинных задач и процессов.

Улучшение качества продукта или услуги: Повышение качества и надежности продукции или услуги с помощью ПО для контроля и управления производством или предоставлением услуг.

Улучшение взаимодействия с клиентами: Создание приложений для улучшения взаимодействия с клиентами, предоставления онлайн-сервисов и усовершенствования клиентского опыта.

Улучшение безопасности: Защита систем и данных организации от киберугроз и угроз безопасности через внедрение специализированных ПО и мер безопасности.

Сокращение затрат: Оптимизация бизнес-процессов и снижение операционных расходов с использованием эффективного ПО.

**Задачи:**

Выбор ПО: Идентификация подходящего ПО с учетом потребностей организации, бюджета и функциональных требований.

Установка и настройка: Установка программного обеспечения на компьютеры и серверы, а также его настройка в соответствии с требованиями.

Обучение персонала: Подготовка сотрудников к использованию нового ПО путем обучения и обеспечения необходимых ресурсов и материалов.

Тестирование: Проведение тестирования для обеспечения стабильной работы ПО и выявления возможных проблем.

Миграция данных: Перенос данных из старых систем в новое ПО, если необходимо.

Сопровождение и обновление: Поддержка и обновление ПО для обеспечения его надежной и актуальной работы.

Мониторинг и анализ: Оценка производительности и эффективности внедренного ПО, а также анализ его влияния на бизнес-процессы.

Цели и задачи внедрения ПО могут сильно различаться в зависимости от конкретной ситуации, и они должны быть четко определены перед началом проекта внедрения.

1. **Назовите основные рабочие группы в команде разработчиков.**

**команде разработчиков обычно выделяют следующие основные рабочие группы:**

**Front-end разработчики:** Отвечают за создание пользовательского интерфейса, который видит и взаимодействует с ними конечные пользователи.

**Back-end разработчики:** Занимаются разработкой серверной части приложения, управлением базами данных, и обеспечивают взаимодействие между фронт- и бэк-эндом.

**QA (Quality Assurance) или тестировщики**: Отвечают за тестирование приложения на ошибки и баги, а также за обеспечение его качества и соответствия требованиям.

**Дизайнеры**: Разрабатывают дизайн интерфейса, графику и визуальное оформление продукта.

**Продуктовые менеджеры:** Занимаются формированием требований к продукту, определением его функциональности и управлением процессом разработки.

**DevOps-инженеры:** Занимаются настройкой и поддержкой инфраструктуры, автоматизацией процессов разработки и развертывания приложения.

**Аналитики данных:** Если приложение связано с анализом данных, то аналитики занимаются сбором, анализом и визуализацией данных.

Системные администраторы: Занимаются управлением серверами и инфраструктурой, обеспечивая их надежную работу.

Эти группы работают совместно для успешной разработки и поддержки программных продуктов.

1. **Дайте функциональную характеристику каждой группе**

**функциональная характеристика каждой из основных рабочих групп в команде разработчиков:**

**Front-end разработчики:**

Отвечают за создание пользовательского интерфейса (UI) приложения.

Разрабатывают клиентскую часть, с которой взаимодействуют пользователи.

Занимаются версткой, стилизацией и созданием интерактивных элементов на веб-сайтах или в мобильных приложениях.

**Back-end разработчики:**

Разрабатывают серверную часть приложения, обеспечивая его функциональность и безопасность.

Управляют базами данных и обеспечивают взаимодействие фронт- и бэк-энда через API.

Решают задачи, связанные с хранением и обработкой данных, а также управлением серверами.

**QA (Quality Assurance) или тестировщики:**

Проводят тестирование приложения на ошибки, баги и несоответствие требованиям.

Отвечают за обеспечение качества разрабатываемого продукта.

Создают тестовые сценарии и отчеты о найденных проблемах.

**Дизайнеры:**

Разрабатывают дизайн интерфейса, определяя его визуальное оформление и пользовательскую эргономику.

Создают макеты, иллюстрации и графические элементы приложения.

Сотрудничают с разработчиками для внедрения дизайна в приложение.

**Продуктовые менеджеры:**

Формулируют требования к продукту, определяют его стратегию развития и функциональность.

Управляют процессом разработки, учитывая потребности пользователей и бизнес-цели.

Принимают решения о приоритетах и выпуске новых версий продукта.

**DevOps-инженеры:**

Обеспечивают автоматизацию процессов разработки, тестирования и развертывания приложения.

Управляют инфраструктурой и серверами, обеспечивая их надежную работу.

Разрабатывают и внедряют системы мониторинга и управления конфигурациями.

**Аналитики данных:**

Занимаются сбором, анализом и визуализацией данных, если приложение связано с аналитикой.

Помогают принимать бизнес-решения на основе данных и метрик.

Разрабатывают отчеты и дашборды для анализа производительности приложения.

**Управление внедрением**

Задания:

1. **В ранее созданных вами группах распределите функции управления.**

Руководитель проекта – Петченко Никита

Лидер разработки – Правдин Евгений

1. Ответственным – Петченко Никита

Исполнительным – Евгений Правдин

1. **Продумайте и опишите все процессы управления.**

Процессы управления могут различаться в зависимости от конкретного проекта или бизнес-процесса. Однако, я могу предложить общие этапы, которые обычно включаются в процессы управления:

**Планирование (Planning):** Этот этап включает в себя определение целей и задач, выбор стратегии и создание плана действий. Важно установить четкие критерии успеха.

**Организация (Organization):** здесь определяются роли и ответственности, а также распределение ресурсов и создание команды, если необходимо.

**Исполнение (Execution):** на этом этапе фактически выполняются задачи согласно установленным планам. Ответственные и исполнители проводят необходимые действия.

**Мониторинг (Monitoring):** Этот этап включает в себя постоянное наблюдение за ходом выполнения, оценку процессов и их соответствия плану. Исправление отклонений, если они возникают.

**Контроль (Control):** В данном случае контроль означает корректировку процессов, если необходимо, чтобы убедиться, что цели достигаются. Это может включать в себя изменение планов или ресурсов.

**Оценка (Evaluation):** По завершении процесса или проекта проводится анализ результатов и эффективности. Это позволяет извлечь уроки для будущих управленческих решений.

**Улучшение (Improvement):** на основе оценки и анализа результатов, процессы могут быть улучшены и оптимизированы для более эффективного выполнения в будущем.

Это общая структура процессов управления, которая может быть адаптирована и дополнена в зависимости от конкретной области применения. Для конкретных задач и проектов важно разработать более подробные процессы управления, учитывая их особенности и требования.

**Контрольные вопросы:**

1. **Что такое методологии внедрения?**

Методологии внедрения (или методологии реализации) - это систематические и структурированные подходы к внедрению новых процессов, систем, технологий или изменений в организации. Они представляют собой набор методов, инструментов, практик и рекомендаций, которые помогают управлять и координировать процесс внедрения, чтобы достичь желаемых результатов.

Методологии внедрения могут применяться в различных областях, включая информационные технологии, управление проектами, управление качеством, бизнес-процессы и другие. Они помогают организациям минимизировать риски, повышать эффективность и ускорять процессы внедрения.

**Некоторые из известных методологий внедрения включают в себя:**

Методология управления проектами (Project Management Methodologies): Например, Agile, Waterfall, PRINCE2, Scrum - методологии, которые помогают управлять проектами от начала до конца.

Методологии управления изменениями (Change Management Methodologies): Такие как ADKAR, Kotter's 8-Step Model - методологии, предназначенные для эффективного управления процессами изменений в организации.

Методологии внедрения информационных систем (Information Systems Implementation Methodologies): Например, SDLC (Software Development Life Cycle), ITIL (Information Technology Infrastructure Library) - методологии, которые ориентированы на внедрение и управление информационными технологиями.

Бизнес-процесс методологии (Business Process Methodologies): Такие как BPMN (Business Process Model and Notation), Six Sigma - методологии, которые помогают оптимизировать и управлять бизнес-процессами.

Выбор методологии внедрения зависит от конкретной задачи, типа проекта и целей организации. Применение методологии помогает систематизировать и структурировать процесс внедрения, что облегчает достижение желаемых результатов.

1. **Назовите известные вам методологии управления:**

Методология управления проектами (Project Management Methodologies):

Agile (Scrum, Kanban)

Waterfall

Принципы PMBOK (Project Management Body of Knowledge)

PRINCE2 (PRojects IN Controlled Environments)

Lean Project Management

Методологии управления качеством:

Six Sigma

Total Quality Management (TQM)

Lean Six Sigma

Методологии управления процессами:

Business Process Reengineering (BPR)

Business Process Management (BPM)

Lean Manufacturing

Методологии управления рисками:

Risk Management Framework (RMF)

ISO 31000

Методологии управления инновациями:

Design Thinking

Lean Startup

TRIZ (Теория решения изобретательских задач)

Методологии управления IT-проектами:

ITIL (Information Technology Infrastructure Library)

DevOps

Scrum (часть Agile-методологии)

Методологии управления знаниями:

Knowledge Management (KM)

Communities of Practice (CoP)

Методологии управления человеческими ресурсами:

Human Resource Management Systems (HRMS)

Talent Management

1. **Что такое управление сроками проекта? Для чего оно нужно?**

Управление сроками проекта — это процесс планирования, оценки, мониторинга и управления временными аспектами проекта. Основная цель — обеспечить завершение проекта в срок и в соответствии с установленными сроками.

Цели управления сроками проекта включают:

Планирование времени: Определение временных рамок для выполнения каждой задачи и этапа проекта.

Оценка ресурсов: Определение необходимых ресурсов (людей, оборудования, материалов) для завершения проекта в срок.

Определение критического пути: Выделение задач, которые определяют общую длительность проекта, и обеспечение их эффективного выполнения.

Разработка графика проекта: Создание графика, отражающего последовательность и продолжительность задач.

Управление изменениями: Реагирование на изменения в проекте, которые могут повлиять на сроки, и корректировка плана при необходимости.

Мониторинг и контроль: Отслеживание прогресса выполнения задач, выявление задержек и предпринятие мер для их устранения.

Оптимизация ресурсов: Максимизация эффективности использования ресурсов для соблюдения сроков.

Управление сроками проекта не только помогает в достижении целей в установленные сроки, но также способствует оптимизации ресурсов и повышению общей эффективности проекта.

1. **Из каких процессов оно состоит?**

**Управление сроками проекта состоит из нескольких основных процессов**. В рамках стандартного подхода к управлению сроками проекта, можно выделить следующие этапы:

Планирование сроков (Schedule Planning): На этом этапе определяются цели и требования проекта, разрабатывается временной план и создается календарь проекта. Важными компонентами являются определение последовательности задач, оценка продолжительности работ и выделение критического пути.

Установление графика (Schedule Development): На этом этапе создается конкретный график проекта на основе плана сроков. Задачи распределяются по времени, определяются начальные и конечные сроки для каждой задачи, и строится графическое представление графика.

Управление графиком (Schedule Control): Этот процесс включает в себя мониторинг и контроль выполнения задач в соответствии с установленным графиком. Если возникают задержки или изменения, производится анализ и принятие корректирующих мер.

Оптимизация сроков (Schedule Optimization): При необходимости производится оптимизация графика проекта для максимизации эффективности использования ресурсов и соблюдения сроков.

Управление рисками сроков (Schedule Risk Management): Этот процесс включает в себя анализ рисков, которые могут повлиять на сроки проекта, и разработку стратегий для их минимизации.

Управление изменениями сроков (Schedule Change Management): Если в ходе проекта возникают изменения, которые могут повлиять на сроки, необходимо эффективно управлять этими изменениями, оценивая их воздействие и принимая соответствующие решения.

Эти процессы взаимосвязаны и помогают обеспечить выполнение проекта в срок и с учетом всех факторов, влияющих на временные аспекты проекта.

1. **В чем заключается управление стоимостью?**

Управление стоимостью (Cost Management) в проектном управлении заключается в планировании, оценке, мониторинге и контроле расходов, связанных с выполнением проекта. Основная цель управления стоимостью — обеспечить выполнение проекта в рамках установленного бюджета.

1. Из каких процессов оно состоит?

Процесс управления стоимостью включает в себя следующие ключевые шаги:

Планирование стоимости (Cost Planning): На этом этапе разрабатывается бюджет проекта. Определяются стоимость ресурсов, материалов, оборудования и труда, необходимых для завершения проекта. Также устанавливаются критерии оценки и контроля стоимости.

Оценка стоимости (Cost Estimation): Оценка производится на основе данных о затратах на ресурсы и труд, а также учетом рисков и неопределенности. Это позволяет получить предварительное представление о стоимости проекта.

Бюджетирование (Budgeting): На основе оценок стоимости создается бюджет проекта. Этот бюджет является основой для финансового управления проектом.

Мониторинг стоимости (Cost Monitoring): В процессе выполнения проекта ведется непрерывное отслеживание затрат и сравнение фактических расходов с бюджетом. Это позволяет выявлять отклонения и задержки в расходах.

Управление затратами (Cost Control): При выявлении отклонений от бюджета, принимаются корректирующие меры. Могут быть изменены процессы, ресурсы, или планы проекта с целью удержания расходов в рамках бюджета.

Управление изменениями стоимости (Cost Change Management): Если в ходе проекта возникают изменения, которые могут повлиять на стоимость, они должны быть оценены и управлены в соответствии с процедурами изменений.

Управление стоимостью позволяет эффективно управлять финансовыми аспектами проекта, минимизировать риски превышения бюджета и обеспечивать контроль над затратами на каждом этапе выполнения проекта.

1. **В чем заключается управление рисками?**

Управление рисками — это процесс определения, анализа и управления потенциальными угрозами и неопределенностями, которые могут повлиять на достижение целей организации или индивида. Основные шаги управления рисками включают:

Идентификация рисков: Определение всех потенциальных рисков, которые могут возникнуть в конкретной ситуации.

Анализ рисков: Оценке вероятности возникновения рисков и их воздействия на цели. Это включает в себя определение приоритетов по рискам.

Планирование управления рисками: Разработка стратегий и методов управления рисками, включая определение мер по снижению рисков и создание планов действий в случае их реализации.

Внедрение мер по управлению рисками: Реализация планов по снижению рисков и мониторинг эффективности принятых мер.

Мониторинг и анализ: Постоянное отслеживание ситуации и периодический анализ рисков, чтобы адаптировать стратегии управления рисками при необходимости.

Коммуникация и отчетность: Важно информировать заинтересованные стороны о результатах анализа рисков и мерах по их управлению.

Управление изменениями: Следить за изменениями в окружающей среде и бизнес-процессах, которые могут повлиять на риски, и адаптировать стратегии соответственно.

Управление рисками позволяет минимизировать потенциальные негативные последствия и повысить вероятность достижения поставленных целей.

**Вывод по теме "Результаты внедрения":**

Анализ результатов внедрения представляет собой важный этап оценки эффективности и успешности любого проекта или инициативы. Основываясь на имеющихся данных и информации, можно сделать следующие общие выводы:

**Оценка целей и задач**: Перед началом проекта были определены четкие цели и задачи. Оценка результатов позволяет определить, насколько успешно эти цели были достигнуты.

**Качество реализации**: Качество выполнения проекта оказывает влияние на его результаты. Высококачественная реализация часто приводит к лучшим результатам.

**Изменения и улучшения**: Важно оценить, какие изменения и улучшения были внесены в организацию благодаря проекту. Это может включать в себя увеличение эффективности, сокращение издержек или улучшение обслуживания клиентов.

**Обратная связь пользователя**: Мнение и обратная связь пользователей о результатах внедрения также имеют значение. Учесть их мнение может помочь улучшить процесс и продукт в будущем.

Финансовые показатели: Оценка финансовых показателей, таких как ROI (возврат инвестиций), прибыльность и срок окупаемости, является важной частью анализа результатов.

Уроки и рекомендации: Изучение результатов позволяет выявить уроки, которые могут быть использованы в будущих проектах. Также важно составить рекомендации по улучшению подхода к внедрению в будущем.

**Контрольные вопросы**

1)Что такое внедрение?

Внедрение - это процесс внесения чего-либо (например, новой технологии, системы, программы, политики или изменений в организации) в практическое использование или реализацию. Этот термин часто используется в контексте бизнеса и информационных технологий.

1. **Какие эффекты можно ожидать от внедрения ПО?**

Внедрение программного обеспечения (ПО) может иметь различные эффекты и влиять на организацию или процессы в зависимости от конкретных целей и характера ПО. Вот несколько общих эффектов, которые можно ожидать:

Автоматизация задач: ПО может автоматизировать рутинные задачи, что позволяет сотрудникам сосредотачиваться на более важных и сложных заданиях.

Увеличение производительности: Эффективное ПО может увеличить производительность рабочих процессов, уменьшив время, требуемое для выполнения задач.

Улучшение качества и точности: ПО может помочь улучшить качество работы, уменьшить вероятность ошибок и повысить точность данных.

Сокращение затрат: Оптимизация процессов через ПО может привести к снижению операционных затрат.

Улучшение управления данными: ПО для управления данными помогает в организации, хранении и анализе информации.

Более эффективное принятие решений: Аналитическое ПО предоставляет инсайты и данные для более обоснованных стратегических

Улучшение клиентского опыта: Клиентоориентированное ПО может повысить уровень обслуживания и удовлетворенность

Инновации и конкурентоспособность: Внедрение новых технологий или инновационных решений через ПО может усилить конкурентоспособность организации.

Соблюдение стандартов и нормативов: ПО может помочь в соблюдении законодательных требований и стандартов в соответствующей области.

Увеличение масштабируемости: ПО может облегчить расширение бизнеса или управление большим объемом данных и клиентов.

Улучшение безопасности: Использование ПО для обеспечения информационной безопасности может уменьшить угрозы и риски.

Ожидаемые эффекты внедрения ПО могут сильно различаться в зависимости от конкретных задач и контекста организации, поэтому важно тщательно планировать и анализировать результаты при внедрении новых программных решений.

1. **Какие эффекты можно ожидать от внедрения ПО?**

Внедрение программного обеспечения (ПО) может привести к различным результатам, в зависимости от целей и конкретного типа ПО. Вот несколько общих результатов, которые можно ожидать:

Автоматизация процессов: Внедрение ПО может автоматизировать рутинные операции и задачи, что ускоряет выполнение заданий и снижает риски человеческих ошибок.

Увеличение производительности: Эффективное ПО может повысить эффективность рабочих процессов и увеличить производительность сотрудников.

Улучшение качества данных: ПО для управления данными может помочь в поддержании точности и надежности данных, что важно для принятия решений.

Снижение затрат: Оптимизация через ПО может сократить операционные расходы и ресурсозатраты.

Увеличение удовлетворенности клиентов: Внедрение клиентоориентированного ПО может улучшить качество обслуживания и удовлетворенность клиентов.

Более эффективное управление ресурсами: Ресурсное планирование и управление ресурсами через ПО могут помочь оптимизировать использование ресурсов компании.

Улучшение прозрачности и отчетности: ПО для анализа данных и создания отчетов может обеспечить более прозрачную и информативную отчетность.

Соблюдение стандартов и нормативов: ПО может помочь в соблюдении законодательных требований и стандартов в соответствующей области.

Инновации и конкурентоспособность: Внедрение новых технологий и инноваций через ПО может усилить конкурентоспособность организации.

Рост выручки и прибыли: ПО, направленное на увеличение продаж или оптимизацию бизнес-процессов, может способствовать росту выручки и прибыли.

Улучшение безопасности: ПО для обеспечения информационной безопасности может уменьшить угрозы и риски для организации.

Результаты внедрения ПО могут быть различными в зависимости от конкретных задач и контекста организации. Поэтому важно определить ясные цели и ожидания перед началом внедрения ПО, а также следить за его эффективностью и адаптировать стратегии при необходимости.

**Распределение ответственности на заключительных этапах внедрения (по группам)**

Задание

1**) В ранее созданных вами группах распределите ключевые роли**

Руководитель проекта – Петченко Никита

Лидер разработки – Правдин Евгений

2**) Определите их ответственность**

Ответственным – Петченко Никита

Исполнительным – Евгений Правдин

3**) Создайте организационную структуру проекта.**

Проект по теме “Выдача кредитов в банке”

├── Исполнительный комитет

│ ├── Директор проекта – Петченко Никита

│ ├── Главный архитектор - Петченко Никита

│ └── Главный менеджер проекта -Правдин Евгений

│

├── Управление проектом

│ ├── Менеджер проекта

│ ├── Ассистент менеджера проекта

│ └── Аналитик проекта

│

├── Технический отдел

│ ├── Главный инженер

│ ├── Разработчик 1 - Петченко Никита

│ ├── Разработчик 2 – Правдин Евгений

│ └── Тестировщик – Правдин Евгений

│

├── Маркетинг и коммуникации

│ ├── Менеджер по маркетингу - Петченко Никита

│ ├── Коммуникационный специалист

│ └── Графический дизайнер – Правдин Евгений

**Контрольные задания:**

1. **Какие ключевые лица принимают решения по управлению проектом?**

Ключевые роли со стороны исполнителя - руководитель проекта (менеджер проекта) со стороны исполнителя и бизнес-менеджер. (изменено)

1. **Какую ответственность несет спонсор проекта?**

Спонсор проекта - обеспечивает организационную сторону проекта и подтверждает правильность целей проекта. В его ведении находится бюджет проекта. Спонсором проекта может быть отдельный человек или целый комитет, в зависимости от масштабов и сложности проекта.

1. **За что отвечают менеджер проекта со стороны заказчика и менеджер проекта со стороны исполнителя?**

Менеджер со стороны заказчика - предоставление ресурсов заказчиков, разрешение проблем и отслеживание состояния проекта

Менеджер проекта(руководитель проекта) отвечает как за успехи, так и за неудачи проекта. В его задачи входит управление сроками, стоимостью, качеством работ с целью удовлетворения ожиданий заказчика и достижения бизнес-целей исполнителя. (изменено)

1. **Каковы обязанности бизнес-менеджера?**

Бизнес-менеджер - отвечает за успешное выполнение проекта и представляет исполнителя в его договорных отношениях с заказчиком

1. **Какие ключевые лица входят в команду управления проектом?**

Команда управления проектом включает координатора проекта, администратора проекта, менеджера по конфигурации.

1. **Перед кем несет ответственность менеджер по качеству?**

Менеджер по качеству несет ответственность перед несколькими стейкхолдерами в проекте:

Проектным менеджером: Менеджер по качеству отчитывается перед проектным менеджером о том, какие шаги предпринимаются для обеспечения высокого стандарта качества в проекте.

Командой проекта: Он также ответственен перед членами команды за обучение и соблюдение процедур и стандартов качества в процессе выполнения задач.

Заказчиком или заказчиком проекта: Менеджер по качеству предоставляет отчеты и обеспечивает доказательства того, что качество продукта или услуги соответствует ожиданиям заказчика.

Системой управления качеством (если применимо): В случае, если используется система управления качеством, менеджер по качеству ответственен за соблюдение ее положений и обеспечение соответствия процессов стандартам.

Аудиторами качества (если применимо): В случае внутренних или внешних аудитов менеджер по качеству может быть обязан предоставить информацию и доказательства своей работы.

1. **Какие еще комитеты могут быть организованы в крупных проектах?**

В крупных проектах могут быть организованы комитет по управлению, комитет по контролю за изменениями, комитет по анализу спорных вопросов.

1. **Какие существуют форматы документирования распределения ролей и ответственности членов команды проекта?**

Документирование распределения ролей и ответственности членов команды проекта может осуществляться разными форматами. Вот некоторые из наиболее распространенных форматов:

Организационная структура проекта: Это документ, который показывает иерархию ролей в проекте и взаимосвязи между ними. Обычно это представляется в виде диаграммы, например, организационной структуры в форме дерева.

Матрица ответственности (RACI-матрица): RACI - это аббревиатура, которая обозначает разные уровни ответственности для каждой роли в проекте. RACI-матрица определяет, кто ответственен (Responsible), кто участвует (Accountable), кто консультирует (Consulted) и кто информируется (Informed) по каждой конкретной задаче или решению в проекте.

Описания ролей и обязанностей: Каждая роль в команде проекта может иметь свое описание, которое подробно определяет ее функции, обязанности и ожидаемые результаты. Эти описания часто оформляются в виде текстовых документов.

Процессы и процедуры управления проектом: Эти документы могут включать в себя информацию о том, какие роли и ответственности связаны с различными проектными процессами. Например, документ по управлению изменениями может указать, какие роли принимают участие в процессе утверждения и внедрения изменений.

Электронные системы управления проектом (например, Microsoft Project или Jira): Многие проектные инструменты позволяют создавать и отслеживать распределение ролей и задач в рамках проекта. Они предоставляют возможность вносить изменения и автоматически уведомлять о них членов команды.