**Вопрос 1: Основные понятия и определения ИС. Жизненный цикл информационных систем.**

**Информационная система** (ИС) – это система, предназначенная для сбора, обработки, хранения и передачи информации с целью поддержки принятия управленческих решений или выполнения определенных задач.

Жизненный цикл информационной системы включает в себя несколько этапов:

— **Определение потребностей бизнеса** – этот этап заключается в анализе потребностей и целей бизнеса, которые нужно решить с помощью информационной системы.

— **Проектирование** – на этом этапе определяется архитектура и функциональные требования к информационной системе.

— **Разработка** – разрабатывается и реализуется программное обеспечение, осуществляется интеграция всех компонентов системы.

— **Тестирование и отладка** – проводятся тесты на соответствие требованиям, выявляются и устраняются ошибки.

— **Внедрение и внедрение в эксплуатацию** – информационная системазапускается и начинает работу в реальных условиях.

— **Эксплуатация и поддержка** – система поддерживается, обновляется, и в случае необходимости модернизируется.

— **Утилизация** – информационная система устаревает и выходит из употребления, либо заменяется новой системой.

**Вопрос 2: Основные модели построения информационных систем, их структура, особенности и области применения.**

**1. Модель "Водопад"**

Структура: последовательное выполнение этапов разработки информационной системы, включая сбор требований, проектирование, реализацию, тестирование и внедрение.

Особенности: подходит для проектов с четко определенными требованиями и жесткими сроками, обеспечивает строгий контроль над процессом разработки.

**2. Модель "Итеративно-инкрементный"**

Структура: разработка информационной системы происходит поэтапно, с возможностью внесения изменений и корректировок на каждом этапе.

Особенности: позволяет быстрее реагировать на изменения в требованиях к системе, улучшить качество в процессе разработки.

**3. Модель "Спираль"**

Структура: комбинация последовательного и итеративного подходов, включая этапы определения целей, анализа рисков, разработки и реализации.

Особенности: акцент на управлении рисками и качеством продукта, позволяет достичь высокой степени гибкости и адаптируемости процесса разработки.

**Вопрос 3: Сервисно - ориентированные архитектуры**.

**Сервис-ориентированная архитектура (СОА)** — **модульный подход к разработке программного обеспечения, который обеспечивает удалённое использование по стандартизированным протоколам распределённых, слабо связанных, легко заменяемых компонентов (сервисов) со стандартизированными интерфейсами.**

**Вопрос 4: Case-средства для моделирования деловых процессов (бизнес-процессов)**.

**Case-средства** — это инструменты для моделирования, анализа и оптимизации бизнес-процессов. Они позволяют организациям визуализировать текущие процессы, определить узкие места и разработать стратегии для повышения эффективности работы.

Некоторые популярные Case-средства для моделирования деловых процессов включают в себя:

— ARIS (Architecture of Integrated Information Systems): инструмент для управления процессами, моделирования и анализа бизнес-процессов.

— Bizagi: платформа для создания моделей бизнес-процессов, автоматизации бизнес-операций и управления изменениями.

— Signavio: решение для моделирования процессов, управления проектами и анализа данных.

— Microsoft Visio: инструмент для создания графических моделей бизнес-процессов, архитектуры информационных систем и других диаграмм.

— IBM Blueworks Live: облачный сервис для моделирования и оптимизации бизнес-процессов, совместной работы и управления проектами.

**Вопрос 5: Инструментальная среда – структура, интерфейс, элементы управления**.

**Инструментальная среда** — это комплекс программных и аппаратных средств, предназначенных для разработки, тестирования и поддержки прикладных программных систем. Структура, интерфейс и элементы управления инструментальной средой определяют её функциональность и удобство использования. Вот основные аспекты инструментальных сред:

1. Структура:

— Модульность: Инструментальная среда часто состоит из отдельных модулей или компонентов, каждый из которых выполняет определенные функции, такие как текстовый редактор, компилятор, отладчик, дизайнер форм и т.д.

— Расширяемость: Хорошая инструментальная среда позволяет легко добавлять новые инструменты или расширения, чтобы удовлетворить специфические потребности разработчиков.

2. Интерфейс:

— Графический пользовательский интерфейс (GUI): Это основной способ взаимодействия пользователя с инструментальной средой. GUI должен быть интуитивно понятным и удобным в использовании.

— Консольный интерфейс: Некоторые инструменты могут быть доступны только через командную строку, что полезно для автоматизации процессов и для опытных пользователей.

3. Элементы управления:

— Меню и панели инструментов: Эти элементы предоставляют быстрый доступ к основным функциям и инструментам.

— Конфигурация и настройки: Инструментальная среда должна позволять пользователям настраивать параметры, такие как язык программирования, цвета, шрифты и т.д.

— Отладка и тестирование: Интегрированные инструменты для отладки и тестирования являются важными элементами управления, которые помогают разработчикам обнаруживать и исправлять ошибки.

— Интеграция с системами управления версиями: Это позволяет разработчикам управлять изменениями в коде и работать над проектами в команде.

**Вопрос 6: Принципы построения модели IDEF0: контекстная диаграмма, субъект моделирования, цель и точка зрения.**

1. Контекстная диаграмма:

— Это первый шаг в построении модели IDEF0. Контекстная диаграмма представляет собой высший уровень абстракции, на котором изображается общая функция организации или системы и её взаимодействие с внешней средой. Она показывает входы (inputs), выходы (outputs), управления (controls) и механизмы (mechanisms) для основной функции.

2. Субъект моделирования:

— Субъект моделирования — это объект или процесс, который анализируется с помощью модели IDEF0. Это может быть отдельная функция, процесс или система в целом. Субъект моделирования определяется на начальном этапе, чтобы сосредоточить анализ на конкретном аспекте деятельности.

3. Цель:

— Целью построения модели IDEF0 является детальное описание функций субъекта моделирования, их взаимосвязей и последовательности выполнения. Цель может включать в себя улучшение процессов, оптимизацию работы, создание руководств по процедурам или поддержку принятия решений.

4. Точка зрения:

— Точка зрения определяет, с какой перспективы будет анализироваться субъект моделирования. Это может быть точка зрения руководства, сотрудников, клиентов или любого другого заинтересованного лица. Точка зрения влияет на то, какие аспекты процесса будут подробно рассмотрены, и какие данные будут включены в модель.

**Вопрос 7: Стратегия развития бизнес-процессов. Критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов.**

**Стратегия развития бизнес-процессов** — это комплекс мер и подходов, направленных на улучшение эффективности и результативности деятельности организации через оптимизацию и модернизацию её бизнес-процессов. Разработка такой стратегии включает в себя оценку текущего состояния предметной области, определение целей и выбор соответствующих методов развития.

**Критерии оценивания предметной области:**

1) Эффективность - оценка того, насколько процессы соответствуют поставленным целям и позволяют достигать результатов;

2) Качество - оценка качества процессов и результатов их выполнения.

3) Удовлетворенность клиентов: Оценка того, насколько клиенты удовлетворены результатами работы бизнес-процессов.

4) Соответствие стратегическим целям: Проверка того, насколько бизнес-процессы соответствуют долгосрочным целям и стратегии организации.

5) Возможности для инноваций: Определение потенциала для внедрения новых технологий и методов работы.

6) Риски: Анализ возможных рисков, связанных с бизнес-процессами, и оценка их влияния на деятельность организации.

**Методы определения стратегии развития бизнес-процессов:**

1) Анализ SWOT: Оценка сильных и слабых сторон, возможностей и угроз в рамках бизнес-процессов.

2) Моделирование и оптимизация процессов: Использование методов моделирования (например, IDEF0, BPMN) для визуализации и анализа процессов, а также применение оптимизационных техник для улучшения их эффективности.

3) Корпоративное управление: Включение принципов корпоративного управления для обеспечения прозрачности, ответственности и контроля за бизнес-процессами.

4) Интеграция информационных систем: Внедрение или модернизация информационных систем (ERP, CRM, BPM) для автоматизации и интеграции бизнес-процессов.

5) Разработка стандартов и процедур: Создание стандартов, руководств и процедур, которые обеспечат последовательность и качество выполнения бизнес-процессов.

6) Тренинги и развитие персонала: Обучение сотрудников новым навыкам и методам работы, чтобы подготовить их к изменениям в бизнес-процессах.

7) Мониторинг и анализ: Регулярный мониторинг ключевых показателей эффективности (KPI) и анализ результатов для корректировки стратегии развития.

**Вопрос 8: Предпроектная стадия разработки. Техническое задание на разработку: основные разделы.**

Предпроектная стадия разработки — это первый этап создания проекта, на котором происходит анализ и планирование будущего продукта. Одним из важнейших документов на этой стадии является техническое задание (ТЗ) на разработку.

Техническое задание определяет цели, требования и ограничения проекта. Оно служит основой для всех последующих этапов разработки и должно быть максимально чётким и подробным. Основные разделы технического задания на разработку обычно включают:

Основные разделы технического задания на разработку:

1. Введение:

— Назначение документа: Описание цели создания ТЗ и кто его будет использовать.

— Объект разработки: Краткое описание того, что будет разрабатываться (система, продукт, услуга и т.д.).

— Предполагаемые результаты: Четко сформулированные ожидаемые результаты и выгоды от реализации проекта.

2. Основная часть:

— Назначение и область применения: Описание предназначения разрабатываемого объекта и сферы его использования.

— Требования к функциональности: Детальное описание функций и возможностей, которыми должен обладать разрабатываемый объект.

— Требования к характеристикам: Описание технических характеристик, таких как производительность, надежность, масштабируемость и т.д.

— Требования к интерфейсам: Описание взаимодействия с другими системами, пользователями и внешней средой.

— Требования к информационной и программной совместимости: Указание на совместимость с существующими информационными и программными системами.

— Требования к безопасности: Описание мер безопасности, необходимых для защиты данных и обеспечения конфиденциальности.

— Требования к условиям эксплуатации: Описание условий, в которых будет эксплуатироваться разрабатываемый объект.

— Требования к документированию: Указание на необходимость создания документации для пользователей и обслуживающего персонала.

3. Этапы и сроки разработки:

— План разработки: График работ, включая этапы разработки, сроки и ключевые мероприятия.

— Сроки выполнения: Установление временных рамок для каждого этапа и для всего проекта в целом.

4. Ресурсы и затраты:

— Финансовые ресурсы: Оценка бюджета проекта, включая стоимость материалов, оборудования, оплату труда и т.д.

— Материальные ресурсы: Перечень необходимых материалов, оборудования и инфраструктуры.

— Кадровые ресурсы: Описание требуемых квалификаций и количества персонала.

5. Оценка рисков:

— Анализ рисков: Идентификация возможных рисков и меры по их минимизации.

— Планирование управления рисками: Описание процесса мониторинга и управления рисками на протяжении всего проекта.

6. Приложения:

— Технические и нормативные документы: Ссылки на стандарты, нормы и другие документы, которые необходимо учитывать при разработке.

— Дополнительные материалы: Рисунки, схемы, таблицы и другие вспомогательные материалы, которые помогают в понимании проекта.

**Вопрос 9: Основные понятия и определения информационных систем:**

1) Информационная система (ИС):

Это система, которая используется для сбора, хранения, обработки и распространения информации. Она включает в себя аппаратное и программное обеспечение, данные, персонал и процедуры.

2) Информация:

Это данные, которые имеют смысл и значение в определенной контексте. Информация используется для принятия решений, анализа ситуаций и управления процессами.

3) Данные:

Это факты, числа, символы или другие элементы, которые не имеют смысла сами по себе, но могут быть обработаны для получения информации.

4) Система:

Это совокупность взаимодействующих элементов, организованных для достижения определенной цели. Система имеет входы, выходы, процессы и управляющие элементы.

5) Аппаратное обеспечение:

Это физические компоненты информационной системы, такие как компьютеры, серверы, устройства ввода/вывода, сети и другие технические устройства.

6) Программное обеспечение:

Это совокупность программ, которые запускаются на аппаратном обеспечении для выполнения определенных задач. Оно включает в себя операционные системы, прикладные программы, базы данных и т.д.

7) База данных:

Это организованная коллекция данных, которая хранится и управляется с помощью систем управления базами данных (СУБД). База данных предназначена для эффективного хранения, извлечения и обновления данных.

8) Интеграция:

Это процесс объединения различных компонентов информационной системы для обеспечения единой и согласованной работы. Интеграция может включать в себя объединение данных, приложений и систем.

9) Автоматизация:

Это процесс использования технологий для выполнения задач без прямого участия человека. Автоматизация повышает эффективность и точность работы информационных систем.

10) Безопасность:

Это комплекс мер, направленных на защиту информационных систем от несанкционированного доступа, использования, раскрытия, разрушения, изменения или непреднамеренного разрушения.

11) Управление данными:

Это процесс организации и поддержания данных на всех этапах их жизненного цикла, включая сбор, хранение, обработку, распространение и архивирование.

12) Информационная технология (ИТ):

Это использование компьютеров для хранения, извлечения, передачи и обработки информации. Информационные системы являются частью более широкой области информационных технологий.

**Вопрос 10: Анализ предметной области различными методами.**

**Анализ предметной области** — это процесс изучения и понимания специфики деятельности, проблем и требований, связанных с определенной сферой или бизнесом. Это важный этап при разработке информационных систем, внедрении новых технологий или оптимизации бизнес-процессов.

Основные методы анализа предметной области:

1) Интервьюирование:

Этот метод предполагает прямое общение с заинтересованными сторонами (руководством, сотрудниками, клиентами) для сбора информации о процессах, задачах и проблемах. Интервью могут быть структурированными (с заранее подготовленными вопросами) или неструктурированными (более свободный диалог).

2) Наблюдение:

Анализ предметной области может включать в себя наблюдение за тем, как работают сотрудники и как функционируют процессы. Это позволяет получить глубокое понимание того, как на самом деле выполняются задачи, и обнаружить проблемы, которые могут быть не очевидны в теории.

3) Мозговой штурм:

Это метод коллективной генерации идей, который используется для выявления возможных решений проблем и новых подходов к работе. Мозговой штурм может быть полезен для определения требований к системе или для поиска инновационных идей.

4)Анализ документов и отчетов:

Изучение существующей документации, отчетов, руководств и других материалов может предоставить ценную информацию о предметной области. Этот метод помогает понять исторические аспекты, текущие проблемы и стратегические цели.

5) Прототипирование:

Создание прототипов систем или процессов может быть использовано для тестирования и уточнения требований. Прототипирование позволяет заинтересованным сторонам увидеть, как будет работать система, и внести изменения на ранних этапах разработки.

6) Моделирование процессов:

Использование методов моделирования, таких как BPMN (Business Process Model and Notation) или IDEF0, для визуализации и анализа бизнес-процессов. Моделирование помогает выявить узкие места, повторяющиеся задачи и возможные улучшения.

7) SWOT-анализ:

Анализ сильных и слабых сторон, возможностей и угроз в рамках предметной области. SWOT-анализ помогает определить стратегические направления развития и меры по улучшению эффективности.

8) Анализ конкурентов:

Изучение подходов и решений, используемых конкурентами, может дать представление о лучших практиках и новых тенденциях в отрасли. Этот анализ помогает определить, как можно улучшить свои процессы и продукты.

9) Анализ рисков:

Оценка возможных рисков, связанных с предметной областью, и разработка стратегий их минимизации. Анализ рисков важен для обеспечения устойчивости и надежности бизнес-процессов.

10) Анализ требований:

Сбор и систематизация требований заинтересованных сторон к будущей системе или процессу. Анализ требований включает в себя определение функциональных и нефункциональных требований, а также согласование их с заказчиком и пользователями.

**Вопрос 11: Постановка задачи обработки информации.**

**Постановка задачи обработки информации — это процесс определения чётких и конкретных целей, которые должны быть достигнуты при обработке данных с помощью информационных систем или технологий. Этот этап является важным перед разработкой системы или внедрением новых процессов, поскольку он определяет, какие действия необходимо выполнить и какие результаты ожидаются.**

**Вопрос 12: Основные модели построения информационных систем:**

**1) Модель централизованной системы:**

**В этой модели все данные и процессы управления сосредоточены на одном центральном сервере или в одной базе данных. Пользователи взаимодействуют с системой через терминалы или клиентские приложения, которые передают запросы на сервер и получают ответы. Примеры: традиционные системы обработки транзакций (TPS), системы управления ресурсами предприятия (ERP).**

**2) Модель распределенной системы:**

**В распределенных системах данные и процессы распределены между несколькими узлами или серверами, которые могут находиться на удаленных физических местах. Система обеспечивает согласованность данных и координацию работы узлов. Примеры: клиент-серверные системы, системы с выделенными серверами для определенных функций (например, файловый сервер, сервер баз данных).**

**3) Модель мультиагентных систем:**

**В этой модели система состоит из множества автономных агентов, каждый из которых способен выполнять определенные задачи и взаимодействовать с другими агентами для достижения общей цели. Агенты могут быть программными модулями или физическими устройствами. Примеры: системы управления интеллектуальными домашними устройствами, системы автоматизированного управления производственными процессами.**

**4) Модель облачных вычислений:**

**В модели облачных вычислений ресурсы (серверы, базы данных, приложения) предоставляются пользователям через интернет как услуга. Пользователи могут получать доступ к ресурсам в зависимости от своих потребностей, не заботясь о физическом размещении и техническом обслуживании. Примеры: SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service).**

**5) Модель гибридных систем:**

**Гибридные системы объединяют в себе элементы нескольких моделей. Например, часть системы может быть развернута в облаке, а часть — на локальных серверах. Или система может использовать централизованное управление данными, но распределенные процессы обработки.**

**6) Модель микросервисов:**

**В этой модели система разбивается на множество маленьких, автономных сервисов, каждый из которых реализует отдельную функцию или бизнес-процесс. Сервисы взаимодействуют друг с другом через API и могут быть легко перепрофилированы или заменены. Это позволяет ускорить разработку и обновление системы, а также повысить ее масштабируемость и надежность.**

**7) Модель сервис-ориентированной архитектуры (SOA):**

**SOA представляет собой набор сервисов, каждый из которых выполняет определенную функцию и может быть использован другими сервисами или приложениями через сетевое взаимодействие. Сервисы обмениваются данными через стандартизированные интерфейсы, что позволяет легко интегрировать различные системы и приложения.**

**Вопрос 13: Сервисно - ориентированные архитектуры:**

**Сервисно-ориентированная архитектура (SOA) — это методология проектирования информационных систем, основанная на использовании независимых компонентов (сервисов), которые взаимодействуют между собой через стандартизированные интерфейсы. Ключевые принципы SOA включают модульность, повторное использование кода, гибкость и масштабируемость системы.**

**Преимущества сервисно-ориентированных архитектур включают:**

— **Гибкость и масштабируемость**

— **Повышенная производительность**

— **Легкость интеграции**

— **Гибкий доступ к данным**

— **Легкость обновления и поддержки**

**Вопрос 14: Методы и средства проектирования информационных систем.**

**1) Методологии проектирования:**

— **Процессно-ориентированное проектирование: Определение бизнес-процессов и их автоматизация.**

— **Объектно-ориентированное проектирование: Использование объектной модели для представления реальных объектов и их взаимодействий.**

— **Моделирование данных: Использование методов ER-моделирования, UML и других для визуализации структуры данных.**

— **Сервис-ориентированное проектирование (SOA): Разработка системы как набора взаимодействующих сервисов.**

— **Микросервисная архитектура: Разбиение системы на мелкие, автономные сервисы.**

**2) Средства проектирования:**

— **CASE-инструменты (Computer-Aided Software Engineering): Программные инструменты, которые помогают визуализировать и документировать архитектуру и компоненты системы (например, Rational Rose, Visio, Enterprise Architect).**

— **Инструменты для моделирования данных: Инструменты, которые помогают визуализировать и проектировать базы данных (например, ER/Studio, Oracle SQL Developer Data Modeler).**

— **Инструменты для проектирования интерфейсов: Программные средства для создания макетов пользовательского интерфейса (например, Adobe XD, Sketch, Figma).**

— **Инструменты для проектирования архитектуры: Инструменты, которые помогают визуализировать и документировать архитектуру системы (например, Archimate, UML-инструменты).**

**3) Методы анализа и проектирования:**

— **Методы анализа требований: Интервьюирование, наблюдение, мозговой штурм, прототипирование, анкетирование.**

— **Методы проектирования баз данных: Нормализация, проектирование схем баз данных, создание индексов, оптимизация запросов.**

— **Методы проектирования алгоритмов и процессов: Структурное программирование, объектно-ориентированное программирование, паттерны проектирования, алгоритмические методы.**

**4) Проектирование безопасности:**

— **Методы обеспечения безопасности: Аутентификация, авторизация, шифрование, контроль доступа, мониторинг и аудит.**

— **Инструменты для проектирования безопасности: Средства для моделирования угроз, оценки рисков, проектирования политик безопасности.**

**5) Проектирование пользовательского опыта (UX):**

— **Методы UX-дизайна: Анализ пользовательских сценариев, создание прототипов, тестирование интерфейсов, анкетирование и интервьюирование пользователей.**

— **Инструменты UX-дизайна: Программные средства для создания макетов, прототипов и тестирования пользовательского опыта.**

**6) Проектирование облачных систем:**

— **Методы проектирования облачных систем: Выбор модели облачных услуг (SaaS, PaaS, IaaS), проектирование облачных приложений, управление облачными ресурсами.**

— **Инструменты для проектирования облачных систем: Средства для развертывания и мониторинга облачных приложений, инструменты для автоматизации развертывания (например, Terraform, Ansible).**

**Вопрос 15: Разработка модели архитектуры информационной системы.**

**Разработка модели архитектуры информационной системы — это процесс создания структуры, компонентов и взаимодействия между ними, которые составляют информационную систему. Важной частью этого процесса является определение основных аспектов системы, таких как типы компонентов, их функции, интерфейсы и взаимосвязи.**

**Шаги, которые обычно включаются в разработку модели архитектуры информационной системы:**

— **Определение требований**

— **Создание функциональной модели**

— **Определение структуры и компонентов**

— **Определение архитектурного стиля**

— **Документирование и анализ**

**Вопрос 16: Обоснование выбора средств проектирования информационной системы.**

**Обоснование выбора средств проектирования информационной системы — это важный процесс, который помогает убедиться в том, что выбранные инструменты и методы соответствуют требованиям проекта и обеспечивают эффективное и успешное проектирование системы. Вот несколько ключевых факторов, которые могут быть использованы для обоснования выбора средств проектирования информационной системы:**

— **Соответствие требованиям проекта (анализ требования к проекту и убедиться, что выбранные средства и методы способны эффективно реализовать необходимые функции и возможности информационной системы.**

— **Опыт и компетенция команды (выбор средств, с которыми разработчики уже знакомы, может повысить производительность и качество работы)**

— **Стоимость и доступность (стоимость лицензий или использования выбранных средств, а также бюджет проекта и доступность средств для команды разработки)**

— **Интеграция и совместимость (выбранные средства проектирования могут интегрироваться с другими системами, с которыми ваша информационная система должна взаимодействовать, и обеспечивать необходимую совместимость)**

— **Поддержка и обновления (поддержка и обновления выбранных средств. Важно, чтобы инструменты использовались активно и поддерживались разработчиками для обеспечения безопасности и актуальности информационной системы)**

**Вопрос 17: Описание бизнес-процессов заданной предметной области.**

**Описание бизнес-процессов заданной предметной области — это процесс документирования и визуализации последовательности действий, которые выполняются в рамках определенного бизнеса или организации для достижения поставленных целей. Для описания бизнес-процессов обычно используются моделирование процессов, диаграммы, таблицы и текстовые описания.**

**Этапы описания бизнес-процессов:**

**1) Анализ предметной области:**

**Изучение текущей деятельности организации, выявление ключевых процессов, участников и ресурсов.**

**2) Определение границ процессов:**

**Определение, какие действия и задачи следует включить в бизнес-процесс, а какие оставить за его пределами.**

**3) Моделирование процессов:**

**Создание графических моделей процессов с использованием стандартов, таких как BPMN (Business Process Model and Notation), UML (Unified Modeling Language) или IDEF0.**

**4) Описание шагов процесса:**

**Текстовое описание каждого шага процесса, включая входные данные, выполняемые действия, используемые ресурсы и выходные результаты.**

**5) Определение участников процесса:**

**Указание сотрудников или подразделений, ответственных за выполнение каждого шага процесса.**

**6) Определение ключевых показателей эффективности (KPI):**

**Установление метрик, которые будут использоваться для оценки эффективности бизнес-процесса.**

**7) Анализ и оптимизация процессов:**

**Анализ существующих процессов на предмет возможных улучшений, выявление узких мест и разработка рекомендаций по оптимизации.**

**8) Внедрение и мониторинг:**

**Внедрение изменений в бизнес-процессы и постоянный мониторинг их работы для обеспечения соответствия установленным KPI.**

**Вопрос 18: Case-средства и технологии.**

**CASE-средства (Computer-Aided Software Engineering) — это программные инструменты, которые помогают автоматизировать различные этапы жизненного цикла разработки программного обеспечения, включая проектирование, разработку, тестирование и документирование. CASE-технологии направлены на повышение эффективности и качества разработки информационных систем.**

**Основные типы CASE-средств:**

**1) Инструменты для анализа и проектирования:**

— **Моделирование данных: Инструменты для создания ER-диаграмм, UML-диаграмм и других моделей данных (например, ER/Studio, Oracle SQL Developer Data Modeler).**

— **Моделирование процессов: Инструменты для создания диаграмм бизнес-процессов (например, Bizagi, Visio, ARIS).**

— **Проектирование интерфейсов: Инструменты для создания макетов пользовательского интерфейса (например, Adobe XD, Sketch, Figma).**

**2) Инструменты для генерации кода:**

— **Инструменты, которые автоматически создают исходный код на основе разработанных моделей (например, Rational Rose, Enterprise Architect).**

**3) Инструменты для тестирования:**

— **Инструменты для автоматизации тестирования, включая модульное тестирование, интеграционное тестирование и системное тестирование (например, JUnit, Selenium, LoadRunner).**

**4) Инструменты для управления проектами:**

— **Инструменты для планирования, отслеживания и управления проектами разработки (например, Jira, Trello, Asana).**

**5) Инструменты для документирования:**

— **Инструменты для автоматического создания документации на основе разработанных моделей и кода (например, Doxygen, Javadoc).**

**6) Инструменты для конфигурационного управления:**

— **Инструменты для управления версиями и изменениями в коде и других ресурсах проекта (например, Git, SVN, Mercurial).**

**7) Инструменты для анализа и оптимизации:**

— **Инструменты для анализа производительности, профилирования кода и оптимизации (например, Valgrind, VisualVM).**

**Преимущества использования CASE-средств:**

— **Повышение эффективности: Автоматизация повторяющихся задач уменьшает время разработки и улучшает использование ресурсов.**

— **Повышение качества: CASE-средства помогают создавать более надежные и качественные системы за счет стандартизации и контроля процессов.**

— **Улучшение документирования: Автоматическое создание документации упрощает понимание системы и ее поддержку.**

— **Повышение гибкости: CASE-средства позволяют легко вносить изменения в проект и быстро адаптировать систему к новым требованиям.**

**Недостатки использования CASE-средств:**

— **Стоимость: CASE-средства могут быть дорогими, особенно для малых и средних предприятий.**

— **Требует обучения: Работа с CASE-инструментами требует специальных знаний и навыков, что может потребовать дополнительного обучения.**

— **Ограниченная универсальность: Некоторые CASE-инструменты могут быть специализированы под определенные платформы или технологии, что ограничивает их применимость.**

**Вопрос 19: Принципы построения модели IDEF0.**

**IDEF0 (Integration Definition for Function Modeling) — это методология моделирования функций, которая используется для документирования и анализа бизнес-процессов и систем. IDEF0 помогает визуализировать и структурировать процессы таким образом, чтобы их можно было легко понять и анализировать. Модели IDEF0 строятся на основе определенных принципов и правил.**

**Основные принципы построения модели IDEF0:**

**1) Актуальность:**

**Модель должна отражать текущее состояние процесса или системы.**

**2) Полнота:**

**Модель должна включать все важные аспекты процесса или системы.**

**3) Консистентность:**

**Модель должна быть последовательной и непротиворечивой.**

**4) Простота:**

**Модель должна быть ясной и легко читаемой, избегая ненужной сложности.**

**5) Декомпозиция:**

**Модель строится путем последовательного разбиения процесса на более мелкие подпроцессы (функции).**

**6) Структурированность:**

**Модель должна иметь четкую иерархическую структуру, где каждая функция может быть разбита на подфункции.**

**7) Обозначение:**

**Используется стандартное обозначение, где функции изображаются в виде прямоугольников, а взаимодействие с внешней средой и между функциями обозначается стрелками.**

**Вопрос 20: Диаграммы ER.**

**Диаграммы ER (Entity-Relationship) — это графические инструменты, используемые для моделирования и визуализации структуры данных в базах данных. Они помогают определить сущности (объекты или понятия), атрибуты этих сущностей и отношения между ними. Диаграммы ER являются важным этапом проектирования базы данных, так как они предоставляют наглядное представление о том, как данные будут организованы и взаимодействовать.**

**Основные компоненты диаграмм ER:**

**1) Сущности (Entities):**

**Сущности представляют собой объекты или понятия, которые имеют значение для предметной области. Например, в базе данных университета сущности могут включать "Студент", "Преподаватель", "Курс".**

**2) Атрибуты (Attributes):**

**Атрибуты — это свойства или характеристики сущностей. Например, для сущности "Студент" атрибутами могут быть "Имя", "Фамилия", "Номер студенческого билета".**

**3) Отношения (Relationships):**

**Отношения представляют связи между сущностями. Например, между сущностями "Студент" и "Курс" может быть установлено отношение "Занимается", которое указывает, какие студенты посещают какие курсы.**

**Вопрос 21: Диаграмма потоков данных DFD.**

**Диаграмма потоков данных (DFD, Data Flow Diagram) — это графический инструмент, используемый для визуализации системных процессов и потоков данных в рамках информационной системы или бизнес-процесса. DFD помогает понять, как данные вводятся, обрабатываются и передаются между различными компонентами системы.**

**Основные компоненты диаграммы DFD:**

**1) Процессы (Processes):**

**Процессы представляют собой действия или преобразования, которые выполняются над данными. Они изображаются в виде кружков или прямоугольников с закругленными углами.**

**2) Данные (Data Stores):**

**Данные — это хранилища, где данные временно или постоянно сохраняются. Они изображаются в виде параллелограммов или линий с двумя вертикальными линиями на концах.**

**3) Внешние сущности (External Entities):**

**Внешние сущности представляют собой источники или приемники данных, которые находятся за пределами изучаемой системы. Они изображаются в виде прямоугольников.**

**4) Потоки данных (Data Flows):**

**Потоки данных представляют собой направленные потоки информации между процессами, данными и внешними сущностями. Они изображаются в виде стрелок, указывающих направление потока.**

**Структура диаграммы DFD:**

— **Уровень 0 (Контекстная диаграмма):**

**На этом уровне изображается общая схема системы, где один процесс представляет всю систему, а внешние сущности взаимодействуют с ней.**

— **Уровень 1 и далее:**

**На более детальных уровнях DFD разбивается на подпроцессы, которые более подробно описывают функции системы. Каждый подпроцесс может быть развернут в отдельную диаграмму более низкого уровня.**

**Вопрос 22: Диаграмма IDEF3.**

**Диаграмма IDEF3 (Integration Definition for Process Description) — это методология, используемая для документирования и анализа процессов в рамках организационного или системного контекста. IDEF3 позволяет создавать детальные описания процессов, включая последовательности действий, взаимосвязи между ними и условия, которые могут влиять на ход процесса.**

**Основные компоненты диаграммы IDEF3:**

**1) Узлы (Nodes):**

**Узлы представляют собой основные элементы процесса, такие как действия, события или решения. Они изображаются в виде прямоугольников или других фигур, в зависимости от типа узла.**

**2) Дуги (Arcs):**

**Дуги представляют собой потоки между узлами, указывая направление процесса. Они изображаются в виде стрелок, соединяющих узлы.**

**3) Спецификации узлов (Node Specifications):**

**Спецификации узлов предоставляют детальные описания каждого узла, включая информацию о действиях, ресурсах, временных ограничениях и других аспектах.**

**4) Контекстные узлы (Context Nodes):**

**Контекстные узлы используются для определения границ процесса и его взаимосвязи с другими процессами или системами.**

**5) Условия (Conditions):**

**Условия представляют собой логические элементы, которые могут влиять на последовательность процесса. Они изображаются в виде ромбов и могут принимать различные пути в зависимости от выполнения определенных условий.**

**Правила построения диаграммы IDEF3:**

**1) Последовательность действий:**

**Узлы должны быть организованы в логическую последовательность, отражающую ход процесса.**

**2) Обозначение узлов:**

**Каждый узел должен быть четко обозначен и описан, чтобы исключить двусмысленность.**

**3) Направление потока:**

**Дуги должны указывать направление процесса, и каждый узел должен иметь только один входной и один выходной поток.**

**4) Условия и альтернативные пути:**

**Условия должны быть использованы для отображения альтернативных путей в процессе, которые зависят от определенных условий.**

**5) Спецификации:**

**Каждый узел должен быть дополнен спецификациями, которые предоставляют дополнительную информацию о его содержании и роли в процессе.**

**Вопрос 23: Основные понятия качества информационной системы.**

**Качество информационной системы — это комплекс характеристик, которые определяют ее пригодность для решения поставленных задач и удовлетворения потребностей пользователей. Основные понятия качества информационной системы включают:**

**1) Функциональная полнота (Functional Completeness):**

**Система должна предоставлять все необходимые функции и возможности, которые определены в требованиях.**

**2) Надежность (Reliability):**

**Система должна быть стабильной и корректно работать в течение длительного времени без сбоев и отказов.**

**3) Производительность (Performance):**

**Система должна обеспечивать требуемую скорость обработки данных, время отклика и пропускную способность.**

**4) Удобство использования (Usability):**

**Система должна быть легкой в освоении и использовании, иметь интуитивно понятный интерфейс и документированные функции.**

**5) Безопасность (Security):**

**Система должна защищать данные от несанкционированного доступа, повреждения и утечки, обеспечивая аутентификацию, авторизацию и конфиденциальность.**

**6) Масштабируемость (Scalability):**

**Система должна быть способной эффективно работать и обслуживать увеличивающееся количество пользователей и объем данных.**

**7) Поддерживаемость (Maintainability):**

**Система должна быть легко модифицируемой и обслуживаемой, с четким и понятным кодом, а также хорошо документированными процессами.**

**8) Совместимость (Interoperability):**

**Система должна быть способной взаимодействовать с другими системами и технологиями, используя стандартные протоколы и форматы данных.**

**9) Адаптивность (Adaptability):**

**Система должна быть гибкой и легко адаптироваться к изменениям в требованиях, окружающей среде и технологиях.**

**10) Надёжность данных (Data Integrity):**

**Система должна обеспечивать точность, полноту и согласованность данных.**

**11) Отказоустойчивость (Fault Tolerance):**

**Система должна быть способной продолжать работу даже в случае сбоев или ошибок, обеспечивая резервирование и восстановление данных.**

**12) Конфигурируемость (Configurability):**

**Система должна позволять настраивать ее параметры и поведение в соответствии с конкретными требованиями пользователей.**

**13) Прозрачность (Transparency):**

**Система должна обеспечивать прозрачный доступ к данным и функциям для пользователей, не требуя глубокого понимания ее внутреннего устройства.**

**14) Экономичность (Cost-Effectiveness):**

**Система должна быть разработана и поддерживаться с учетом экономической эффективности, обеспечивая оптимальное соотношение между затратами и получаемыми преимуществами.**

**Вопрос 24: Стандарты группы ISO.**

**Стандарты группы ISO (International Organization for Standardization) — это набор международных стандартов, которые устанавливают требования, рекомендации и руководства по различным аспектам деятельности, включая производство, технологии, управление, качество и безопасность. ISO разрабатывает стандарты для улучшения качества продукции и услуг, повышения эффективности процессов и упрощения международной торговли.**

**Основные группы стандартов ISO:**

**1) Стандарты качества:**

— **ISO 9000 — серия стандартов, касающихся основных принципов и языка управления качеством.**

— **ISO 9001 — стандарт, устанавливающий требования к системе менеджмента качества организации.**

**2) Стандарты экологического менеджмента:**

— **ISO 14000 — серия стандартов, касающихся управления экологической деятельностью организации.**

— **ISO 14001 — стандарт, устанавливающий требования к системе экологического менеджмента.**

**3) Стандарты информационной безопасности:**

— **ISO/IEC 27000 — серия стандартов, касающихся управления информационной безопасностью.**

— **ISO/IEC 27001 — стандарт, устанавливающий требования к системе менеджмента информационной безопасности.**

**4)Стандарты безопасности труда:**

— **ISO 45001 — стандарт, устанавливающий требования к системе менеджмента охраны труда и здоровья.**

**Вопрос 25: Методы контроля качества в информационных системах**

**Контроль качества в информационных системах — это комплекс мер, направленных на обеспечение соответствия системы установленным требованиям и стандартам. Методы контроля качества могут быть различными и включают в себя следующие подходы:**

**1) Анализ требований (Requirements Analysis):**

**Проверка того, что требования к системе четко определены, непротиворечивы и полны.**

**2) Тестирование (Testing):**

**Функциональное тестирование — проверка соответствия функций системы описанным в требованиях.**

**Производительность тестирование — оценка скорости, нагрузочной способности и устойчивости системы.**

**Интеграционное тестирование — проверка взаимодействия между компонентами системы.**

**Системное тестирование — проверка системы в целом на соответствие требованиям.**

**Приемочное тестирование — проверка системы заказчиком или пользователями перед принятием.**

**Регрессионное тестирование — проверка системы после внесения изменений, чтобы убедиться, что новые изменения не нарушили работу существующих функций.**

**3) Code Review (Code Inspection):**

**Проверка исходного кода на наличие ошибок, соответствие стилю и стандартам, а также на качество кодирования.**

**4) Модульное тестирование (Unit Testing):**

**Проверка отдельных модулей или компонентов системы на соответствие их функциональности.**

**5) Безопасность тестирование (Security Testing):**

**Проверка системы на уязвимости, аутентификацию, авторизацию и конфиденциальность данных.**

**6) Мониторинг производительности (Performance Monitoring):**

**Непрерывное наблюдение за производительностью системы в реальном времени.**

**7) Аудит системы (System Audit):**

**Инспектирование системы с целью оценки ее соответствия установленным стандартам и процедурам.**

**8) Инструменты автоматизации (Automation Tools):**

**Использование программных инструментов для автоматизации тестирования, мониторинга и анализа системы.**

**9) Статический анализ кода (Static Code Analysis):**

**Использование инструментов для анализа исходного кода без его выполнения, чтобы обнаружить потенциальные ошибки и уязвимости.**

**10) Динамический анализ кода (Dynamic Code Analysis):**

**Анализ исходного кода во время выполнения, чтобы обнаружить проблемы, которые могут не проявляться в статическом анализе.**

**11) Инструменты контроля версий (Version Control Tools):**

**Использование систем контроля версий для отслеживания изменений в коде и обеспечения возврата к предыдущим версиям при необходимости.**

**12) Процессы непрерывной интеграции и непрерывного развертывания (CI/CD):**

**Использование процессов непрерывной интеграции и непрерывного развертывания для автоматизации сборки, тестирования и развертывания системы.**

**13) Оценка пользовательского опыта (User Experience Evaluation):**

**Анализ того, как пользователи взаимодействуют с системой, и оценка удобства использования.**

**14) Обучение и аттестация персонала (Training and Certification):**

**Обеспечение того, чтобы персонал, работающий с системой, был подготовлен и аттестован в соответствии с требованиями.**

**15) Процессы управления изменениями (Change Management Processes):**

**Обеспечение контроля и документирования изменений в системе, чтобы минимизировать риски и обеспечить согласованность.**

**Вопрос 26: Автоматизация системы управления качеством разработки**

**Автоматизация системы управления качеством разработки — это процесс внедрения инструментов и технологий, которые помогают автоматизировать и упростить задачи, связанные с контролем и улучшением качества разработки программного обеспечения. Автоматизация позволяет повысить эффективность, сократить время на выполнение задач и уменьшить количество ошибок.**

**Основные аспекты автоматизации системы управления качеством разработки:**

**1) Инструменты контроля версий (Version Control Systems):**

**Системы контроля версий, такие как Git, Mercurial или Subversion, позволяют отслеживать изменения в коде, объединять вклады от разных разработчиков и легко возвращаться к предыдущим версиям.**

**2) Инструменты непрерывной интеграции (Continuous Integration Tools):**

**Системы непрерывной интеграции, такие как Jenkins, Travis CI или CircleCI, автоматизируют процесс сборки и тестирования кода каждый раз, когда разработчики вносят изменения.**

**3) Инструменты непрерывного развертывания (Continuous Deployment Tools):**

**Системы непрерывного развертывания автоматизируют процесс развертывания новых версий приложения в производственную среду после успешного прохождения тестирования.**

**4) Инструменты автоматизированного тестирования (Automated Testing Tools):**

**Инструменты для автоматизации функционального тестирования (например, Selenium), производительности тестирования (например, JMeter) и тестирования безопасности (например, OWASP ZAP) позволяют выполнять тесты без участия человека.**

**5) Инструменты статического анализа кода (Static Code Analysis Tools):**

**Инструменты, такие как SonarQube, PMD или ESLint, анализируют исходный код на наличие потенциальных ошибок, нарушений стиля кодирования и уязвимостей.**

**6) Инструменты динамического анализа кода (Dynamic Code Analysis Tools):**

**Инструменты, такие как Valgrind или Dynatrace, анализируют поведение программы во время выполнения, чтобы обнаружить проблемы, которые не видны в статическом анализе.**

**7) Инструменты мониторинга производительности (Performance Monitoring Tools):**

**Инструменты, такие как New Relic или AppDynamics, позволяют отслеживать производительность приложений в реальном времени и быстро реагировать на возникающие проблемы.**

**Вопрос 27: Обеспечение безопасности функционирования информационных систем**

**Обеспечение безопасности функционирования информационных систем — это комплекс мер, направленных на защиту системы от несанкционированного доступа, использования, раскрытия, нарушения, изменения или разрушения информации, хранящейся или обрабатываемой системой.**

**Вот основные подходы к обеспечению безопасности информационных систем:**

**1) Политика безопасности (Security Policy):**

**Разработка и внедрение политики безопасности, которая определяет правила и процедуры, необходимые для защиты информации и системы в целом.**

**2) Аутентификация и авторизация (Authentication and Authorization):**

**Использование механизмов аутентификации (например, пароли, двухфакторная аутентификация) для подтверждения личности пользователей и авторизации для ограничения доступа к ресурсам в соответствии с их ролями и правами.**

**3) Конфиденциальность данных (Data Confidentiality):**

**Использование шифрования для защиты конфиденциальности данных, как во время передачи (например, SSL/TLS), так и при хранении (например, шифрование на уровне файлов или баз данных).**

**4) Целостность данных (Data Integrity):**

**Применение механизмов контроля целостности данных, таких как контрольные суммы, хэши и цифровые подписи, для обнаружения несанкционированных изменений в данных.**

**5) Доступность данных (Data Availability):**

**Реализация стратегий резервного копирования, аварийного восстановления и непрерывности бизнеса для обеспечения доступности критически важных данных и систем.**

**6) Безопасность сети (Network Security):**

**Использование брандмауэров, систем обнаружения вторжений (IDS), систем предотвращения вторжений (IPS) и виртуальных частных сетей (VPN) для защиты сети от нежелательного трафика и атак.**

**7) Безопасность приложений (Application Security):**

**Внедрение мер безопасности на уровне приложений, включая проверку ввода данных, защиту от SQL-инъекций, XSS и других уязвимостей.**

**8) Мониторинг и аудит (Monitoring and Auditing):**

**Непрерывное наблюдение за системой с помощью систем мониторинга безопасности (например, SIEM) и проведение регулярных аудитов безопасности для обнаружения и анализа потенциальных угроз и нарушений.**

**9) Обучение и просвещение пользователей (User Awareness and Training):**

**Проведение обучения и просвещения пользователей о методах безопасности, потенциальных угрозах и их роли в поддержании безопасности системы.**

**10) Управление уязвимостями (Vulnerability Management):**

**Регулярная оценка уязвимостей системы с помощью сканирования и тестирования на проникновение, а также применение патчей и обновлений для устранения обнаруженных уязвимостей.**

**11) Реагирование на инциденты безопасности (Incident Response):**

**Разработка и внедрение плана реагирования на инциденты безопасности, включая процедуры обнаружения, анализа, устранения и восстановления после инцидентов.**

**12) Соответствие требованиям и стандартам (Compliance):**

**Обеспечение соответствия системы требованиям законодательства и стандартов безопасности, таких как GDPR, HIPAA, PCI DSS и других.**

**13) Мультифакторная аутентификация (Multi-Factor Authentication, MFA):**

**Использование дополнительных уровней аутентификации, кроме пароля, например, SMS-коды, приложения для генерации кодов и биометрические данные.**

**14) Шифрование дисков (Disk Encryption):**

**Шифрование данных на жестких дисках и съемных носителях для защиты информации в случае кражи или потери устройства.**

**15) Ограничение прав доступа (Least Privilege):**

**Применение принципа "минимальных привилегий", который заключается в предоставлении пользователям и системам минимально необходимого уровня доступа для выполнения их функций.**

**Вопрос 28: Стратегия развития бизнес-процессов**

**Стратегия развития бизнес-процессов — это комплексный подход к улучшению и оптимизации деятельности организации, направленный на повышение эффективности, качества услуг и конкурентоспособности на рынке.**

**Вопрос 29: Реинжиниринг бизнес-процессов**

**Реинжиниринг бизнес-процессов (Business Process Reengineering, BPR) — это стратегический подход к изменению и оптимизации существующих бизнес-процессов с целью достижения значительного улучшения в таких областях, как стоимость, качество, обслуживание и скорость. Реинжиниринг предполагает радикальное переосмысление и перепроектирование процессов, а не их постепенное улучшение.**

**Вопрос 30: Разработка требований безопасности информационной системы**

**Разработка требований безопасности информационной системы — это процесс определения и документирования минимальных стандартов, которым должна соответствовать система для защиты информации от несанкционированного доступа, использования, раскрытия, нарушения, изменения или разрушения.**

**Вопрос 31: Тестирование информационных систем**

**Тестирование информационных систем — это процесс оценки и обеспечения качества системы путем выявления ошибок и дефектов в ее функциональности, производительности, безопасности и других аспектах. Тестирование является важным этапом жизненного цикла разработки информационных систем и включает в себя следующие виды:**

**1) Функциональное тестирование (Functional Testing):**

— **Проверка соответствия функций системы описанным в требованиях и спецификациях. Это включает в себя:**

**Системное тестирование — проверка системы в целом на соответствие требованиям.**

**Интеграционное тестирование — проверка взаимодействия между компонентами системы.**

**Модульное тестирование — проверка отдельных модулей или компонентов системы.**

**Приемочное тестирование — проверка системы заказчиком или пользователями перед принятием.**

**2) Производительность тестирование (Performance Testing):**

— **Оценка скорости, нагрузочной способности и устойчивости системы под различными нагрузками. Это включает в себя:**

**Нагрузочное тестирование — проверка системы на способность выдерживать предполагаемые рабочие нагрузки.**

**Стресс-тестирование — проверка системы на способность функционировать при превышении обычных нагрузок.**

**Тестирование устойчивости — проверка системы на способность поддерживать производительность в течение длительного времени.**

**3) Безопасность тестирование (Security Testing):**

— **Проверка системы на уязвимости, аутентификацию, авторизацию и конфиденциальность данных. Это включает в себя:**

**Тестирование на проникновение — попытка найти и воспользоваться уязвимостями в системе.**

**Тестирование конфиденциальности — проверка защиты данных от несанкционированного доступа.**

**Тестирование целостности — проверка защиты данных от несанкционированного изменения.**

**4) Удобство использования тестирование (Usability Testing):**

— **Оценка удобства использования системы для конечных пользователей.**

**5) Регрессионное тестирование (Regression Testing):**

— **Проверка системы после внесения изменений, чтобы убедиться, что новые изменения не нарушили работу существующих функций.**

**6) Тестирование совместимости (Compatibility Testing):**

— **Проверка системы на совместимость с различными операционными системами, браузерами, устройствами и другими компонентами.**

**7) Тестирование установки (Installation Testing):**

— **Проверка процесса установки системы на различных платформах и конфигурациях.**

**8) Тестирование восстановления (Recovery Testing):**

— **Проверка способности системы восстанавливаться после сбоев или других нештатных ситуаций.**

**9) Тестирование надежности (Reliability Testing):**

— **Проверка способности системы работать без сбоев в течение определенного периода времени.**

**Вопрос 32: Факторы качества ПО:**

— Функциональность. ПО должно выполнять все необходимые функции и операции, предусмотренные требованиями заказчика.

— Надежность. ПО должно работать стабильно и без сбоев в течение длительного времени.

— Эффективность. ПО должно обеспечивать быструю и эффективную работу в рамках выделенных ресурсов.

— Удобство использования. ПО должно быть удобным для пользователя, иметь интуитивно понятный интерфейс и простую навигацию.

— Портабельность. ПО должно корректно работать на различных платформах и операционных системах.

— Сопровождаемость. ПО должно быть легко сопровождаемым, иметь документацию и возможности для обновлений и исправлений.

— Безопасность. ПО должно обеспечивать защиту данных, конфиденциальность и целостность информации.

— Совместимость. ПО должно быть совместимо с другими программами и устройствами, чтобы обеспечить безпроблемную интеграцию.

— Тестируемость. ПО должно быть легко тестируемым, чтобы обеспечить высокую степень надёжности и безопасности.

— Поддержка. ПО должно иметь возможность получения технической поддержки и регулярных обновлений от разработчика.

**Вопрос 33: Виды тестирования.**

**1) По уровню тестирования:**

— **Модульное тестирование (Unit Testing):**

**Проверка отдельных модулей или компонентов ПО на соответствие требованиям.**

— **Интеграционное тестирование (Integration Testing):**

**Проверка взаимодействия между интегрированными модулями или компонентами.**

— **Системное тестирование (System Testing):**

**Проверка системы в целом на соответствие требованиям и функциональности.**

**2) По типу тестирования:**

— **Функциональное тестирование (Functional Testing):**

**Проверка функций ПО на соответствие описанным в требованиях.**

— **Нефункциональное тестирование (Non-Functional Testing):**

**Проверка нефункциональных аспектов, таких как производительность, безопасность, удобство использования и т.д.**

— **Регрессионное тестирование (Regression Testing):**

**Проверка системы после внесения изменений, чтобы убедиться, что новые изменения не нарушили работу существующих функций.**

— **Структурное тестирование (Structural Testing):**

**Тестирование, основанное на анализе структуры кода и потоков управления.**

**3) По автоматизации:**

— **Автоматизированное тестирование (Automated Testing):**

**Тестирование с использованием программных инструментов для выполнения тестовых сценариев.**

— **Ручное тестирование (Manual Testing):**

**Тестирование, выполняемое человеком без использования автоматизированных инструментов.**

**Вопрос 34: Различия и сходства модульного и интеграционного тестирования:**

**Модульное тестирование (Unit Testing) и интеграционное тестирование (Integration Testing) являются двумя важными этапами в процессе тестирования программного обеспечения. Они преследуют разные цели и имеют свои особенности, но также имеют общие черты.**

— **Сходства между модульным и интеграционным тестированием:**

**1) Цель:**

**Обе формы тестирования направлены на выявление дефектов и обеспечение качества ПО.**

**2) Планирование и дизайн:**

**И модульное, и интеграционное тестирование требуют тщательного планирования и разработки тестовых кейсов и сценариев.**

**3) Автоматизация:**

**Оба вида тестирования могут быть автоматизированы с использованием соответствующих инструментов и фреймворков.**

**4) Отчетность:**

**Результаты обоих видов тестирования должны быть документированы и отчеты должны быть доступны для анализа и принятия решений.**

**5) Зависимость от требований:**

**Тестирование, как модульное, так и интеграционное, основывается на требованиях и спецификациях, которые определяют функциональность и взаимодействие компонентов.**

— **Различия между модульным и интеграционным тестированием:**

**1) Уровень тестирования:**

**Модульное тестирование фокусируется на отдельных модулях или компонентах ПО (например, функциях, классах, процедурах), проверяя их на соответствие требованиям в изолированном виде.**

**Интеграционное тестирование проверяет взаимодействие между интегрированными модулями или компонентами, чтобы убедиться, что они работают корректно вместе.**

**2) Объект тестирования:**

**В модульном тестировании объектом тестирования является сам модуль, и часто тестирование проводится разработчиками.**

**В интеграционном тестировании объектом тестирования является взаимодействие между модулями, и тестирование может проводиться как разработчиками, так и тестировщиками.**

**3) Сложность:**

**Модульное тестирование обычно проще, так как оно фокусируется на отдельных частях кода.**

**Интеграционное тестирование более сложное, так как оно требует учета взаимодействия между различными частями системы.**

**5) Время проведения:**

**Модульное тестирование обычно проводится на ранних этапах разработки, когда модули еще не интегрированы в систему.**

**Интеграционное тестирование проводится после завершения модульного тестирования, когда модули уже объединены в более крупные компоненты или систему в целом.**

**6) Результаты:**

**В модульном тестировании результаты обычно касаются отдельных функций или классов.**

**В интеграционном тестировании результаты касаются взаимодействия между модулями и возможных проблем в их интеграции.**

**Вопрос 35: Методы сборки модулей:**

**— Монолитный (Сборка снизу вверх):**

**Этот метод начинается с интеграции самых низкоуровневых модулей или компонентов, которые не зависят от других модулей. Затем постепенно добавляются более высокоуровневые модули, пока не будет достигнута полная интеграция системы.**

**— Инкрементальный (Сборка сверху вниз):**

**В этом методе начинают с интеграции главного модуля или компонента, который управляет другими модулями. Затем постепенно добавляются зависимости, начиная с самых высокоуровневых и заканчивая самыми низкоуровневыми.**

**Вопрос 36: Классификация по объекту тестирования:**

— Unit-тестирование (также модульное тестирование).

— Интеграционное тестирование.

— Системное тестирование (End-to-end тестирование).

— Приемочное тестирование.

**Вопрос 37: Характеристики хорошего теста:**

— Тест полностью автоматизирован

— Тест повторяем: тест не ломается, если приложение не поменялось

— Тест заканчивается валидацией

— Тест достаточно стабилен, чтобы его использовать в CI/CD

— Тест очень легко читать

— Тест требует минимальной поддержки

— Тест работает параллельно с другими тестами и не ломается

**Вопрос 38: Виды тестов**

**1) Базовый тест** — **smoke test**

**Это базовый вид тестирования, цель которого заключается в быстрой проверке основных функций программного обеспечения на предмет явных ошибок и сбоев.**

**2) Инвентаризация**

**Этот вид тестирования помогает определить и проверить все составные части системы, включая ее архитектуру, модули, библиотеки и зависимости.**

**3) Комбинированные тесты**

**Это вид тестирования, который проверяет взаимодействие между различными компонентами или модулями системы.**

**4) Граничные оценки**

**Это вид тестирования, который фокусируется на проверке поведения системы на границах входных и выходных диапазонов данных.**

**5) Ошибочные данные**

**Это данные, которые не соответствуют ожидаемым или допустимым значениям, форматам или структуре, как это определено в требованиях или спецификациях системы. Тестирование с использованием ошибочных данных является важным аспектом тестирования, поскольку оно помогает определить, насколько хорошо система может справиться с непредвиденными или неправильными входными данными и как она реагирует на такие ситуации.**

**6) Нагрузочные тесты**

**Это вид тестирования, который предназначен для оценки производительности системы под воздействием определенной нагрузки.**

**Вопрос 39: Структурное тестирование**

**Структурное тестирование (также известное как "тестирование покрытия кода" или "черное тестирование") — это подход в тестировании программного обеспечения, который фокусируется на проверке структуры и логики программы, а также на обеспечении того, чтобы все элементы кода были протестированы. Цель структурного тестирования — убедиться, что каждая часть кода выполняется по крайней мере один раз во время тестирования, что позволяет выявить потенциальные дефекты.**

**Вопрос 40: Функциональное тестирование**

**Функциональное тестирование — это вид тестирования программного обеспечения, который фокусируется на проверке того, как система реагирует на определенные входные данные и выполняет требуемые функции в соответствии с ее спецификациями и требованиями. Цель функционального тестирования — убедиться, что каждая функция системы работает так, как ожидается, и соответствует описанным в документации требованиям.**

**Вопрос 41: Автоматизированное тестирование программного обеспечения**

**Автоматизированное тестирование программного обеспечения — это процесс использования специализированного программного обеспечения для управления выполнением тестов и сравнения фактических результатов с ожидаемыми. Цель автоматизированного тестирования — повысить эффективность и эффективность тестирования, уменьшить время и усилия, необходимые для проведения тестов, и обеспечить возможность повторного использования тестовых сценариев.**

**Вопрос 42: Тестирование веб-приложений**

Тестирование веб-приложений — это процесс проверки функциональности, производительности, безопасности, совместимости и других аспектов веб-приложений. Цель тестирования веб-приложений заключается в обеспечении их корректной работы и соответствия заданным требованиям.

**Вопрос 43: Оформление баг-репорта:**

— Заголовок баг-репорта

— Описание ошибки

— Шаги для воспроизведения ошибки

**Вопрос 44: Чек-лист. Баг-репорт. Тест-кейс.**

Чек-лист – это список задач, пунктов или критериев, которые нужно выполнить или проверить в процессе выполнения определенной задачи или процедуры.

Баг-репорт – это документ, в котором описывается найденная ошибка или недочет в программном обеспечении или веб-приложении.

Тест-кейс – это документ, который содержит последовательность шагов или действий, необходимых для выполнения конкретного тестового сценария или проверки определенной функциональности программного продукта.

**Вопрос 45: Smoke testing**

Smoke testing – это форма быстрого и поверхностного тестирования, которое выполняется для проверки базовой работоспособности основной функциональности программного продукта после его обновления, исправлений или интеграции нового кода.

**Вопрос 46: Восходящие и нисходящие тестирование**

Восходящее тестирование – проверка основных функций и поведения верхних уровней программного продукта, а затем постепенное углубление в детали и проверка нижних уровней.

Нисходящее тестирование – проверка отдельных компонентов или модулей нижних уровней программного продукта, а затем постепеенный учёт вышестоящих компонентов.

**Вопрос 47: Структура отчёта тестирования:**

— Введение;

— Объект тестирования;

— **Цель тестирования;**

— План тестирования;

— Используемые тестовые сценарии;

— Результаты тестирования;

— Общие выводы и рекомендации;

— Приложения.

**Вопрос 48: Пользовательское тестирование**

**Пользовательское тестирование (User Acceptance Testing, UAT) — это этап тестирования программного обеспечения, на котором конечные пользователи или заказчики проверяют, соответствует ли система их бизнес-требованиям и может ли она быть принята в эксплуатацию. Цель UAT — убедиться, что разработанное ПО работает так, как ожидают пользователи, и может решать их бизнес-задачи.**

**Вопрос 49: Позитивное и негативное тестирование**

— **Позитивное тестирование (также известное как "белое тестирование" или "тестирование по методу "верно"") — это подход в тестировании программного обеспечения, при котором тестировщик сосредотачивается на проверке того, что система корректно обрабатывает валидные, ожидаемые и стандартные входные данные и сценарии использования. Цель позитивного тестирования — убедиться, что функциональные требования и ожидания пользователей реализованы и работают как предполагается.**

— **Негативное тестирование (также известное как "черное тестирование" или "тестирование по методу "неверно"") — это подход в тестировании программного обеспечения, при котором тестировщик сосредотачивается на проверке того, как система обрабатывает неправильные, непредвиденные и нестандартные входные данные и сценарии использования.**

**Вопрос 50: Системное тестирование**

**Системное тестирование — это этап тестирования программного обеспечения, на котором проверяется работа системы в целом, включая все ее компоненты и интеграцию с другими системами. Цель системного тестирования — убедиться, что система соответствует определенным требованиям и может функционировать корректно в рамках всей системы или предприятия.**