Контрольные вопросы № 1

1. **Требования к информационной системе.**  
   Требования к программному обеспечению – это совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Требования могут выражаться в виде текстовых утверждений и графических моделей. Требования функционального характера определяют требуемое поведение программной системы.
2. **Методы анализа и спецификации требований.**  
   Выделяют следующие виды требований функционального характера:
   1. Бизнес-требования – определяют назначение ПО, описываются в документе о видении и границах программного проекта.
   2. Пользовательские требования – определяют набор пользовательских задач, которые должен решать программный продукт, а также способы их решения. Пользовательские требования могут выражаться в виде фраз утверждений, сценариев использования, сценариев взаимодействия, пользовательских историй.
   3. Функциональные требования – охватывают предполагаемое поведение системы, определяя действия, которые система способна выполнять. Процесс разработки требований включает в себя выполнение следующих этапов:
      1. выявление требований (сбор, понимание, рассмотрение и выяснение потребностей заинтересованных лиц);
      2. анализ (проверка целостности и законченности);
      3. спецификация (документирование требований);
      4. проверка правильности.
3. **Анализ предметной области.**

**В общем и целом:**

Анализ предметной области - это процесс изучения и понимания конкретной сферы знаний или деятельности. Он включает в себя изучение основных понятий, принципов, процессов, требований и проблем, связанных с этой областью.

Целью анализа предметной области является создание полного и точного описания того, что составляет данную область, а также выявление её особенностей, слабых мест и потенциальных возможностей для улучшения.

Этот процесс часто применяется в области разработки программного обеспечения, системного проектирования, управления проектами и других областях, где необходимо четко определить требования и цели перед началом работы.

**Про наш проект в частности:**

* Типы долговых отношений: Первоначально необходимо определить различные виды долговых отношений, которые будут учитываться в приложении. Например, это могут быть кредиты, займы, долги по кредитным картам, долги среди друзей или членов семьи.
* Данные о долгах: Важно определить информацию, которая будет храниться для каждого долга. Это может включать в себя сумму долга, дату выдачи, сроки возврата, проценты (если применимо), информацию о должнике и т. д.
* Безопасность: Поскольку в приложении будут храниться чувствительные финансовые данные, необходимо обеспечить высокий уровень безопасности, включая защиту данных и обеспечение конфиденциальности информации.
* Законодательство и регулирование: Важно учитывать законодательные аспекты, связанные с учетом долговых отношений и обработкой финансовых данных, чтобы приложение соответствовало действующим нормативам и требованиям безопасности данных.

1. **Разработка технического задания на создание информационной системы.**  
   Разработка ТЗ включает в себя подготовку специального документа с аналогичным названием. В Техническом задании обязательно должны быть описаны:
   1. ограничения, риски, критические факторы, влияющие на успешность проекта, например время реакции системы на запрос является заданным ограничением, а не желательным фактором;
   2. совокупность условий, при которых предполагается эксплуатировать будущую систему: архитектура системы, аппаратные и программные ресурсы, предоставляемые системе, внешние условия её функционирования, состав людей и работ, которые обеспечивают бесперебойное функционирование системы;
   3. сроки завершения отдельных этапов, форма сдачи работ, ресурсы, привлекаемые в процессе разработки проекта, меры по защите информации;
   4. описание выполняемых системой функций;
   5. будущие требования к системе в случае её развития, например возможность работы пользователя с системой с помощью Интернета и т.п.;
   6. сущности, необходимые для выполнения функций системы;
   7. интерфейсы и распределение функций между человеком и системой;
   8. требования к программным и информационным компонентам ПО, требования к СУБД. Если проект предполагается реализовывать для нескольких СУБД, то требования к каждой из них, или общие требования к абстрактной (например, распределённой) СУБД и список рекомендуемых для данного проекта СУБД, которые удовлетворяют заданным условиям;
   9. что не будет реализовано в рамках проекта.

Разработка ТЗ ведётся в соответствии со стандартами:

1. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. **Концептуальные требования.**

Концептуальные требования - это высокоуровневое описание функциональности и характеристик системы или продукта, которое обычно представлено в виде абстрактных концепций, идей и общих принципов. Эти требования фокусируются на том, что должна делать система, а не на том, как это должно быть сделано.

Основные характеристики концептуальных требований:

* Абстрактность: Концептуальные требования описывают функциональность системы на высоком уровне абстракции, не вдаваясь в детали реализации.
* Основные функции: Они определяют основные функции и возможности системы, обычно без конкретных деталей или спецификаций.
* Направление разработки: Концептуальные требования обеспечивают направление для дальнейшей разработки системы, помогая определить ее общие цели и ценности.
* Понимание целей заказчика: Эти требования помогают понять потребности и цели заказчика или пользователей системы.
* Гибкость: Поскольку концептуальные требования не описывают детали реализации, они могут быть относительно гибкими и способствовать творческому процессу при разработке.
* Валидация концепции: Они позволяют проверить и уточнить концепцию системы перед тем, как начать более детальную разработку.

Примеры концептуальных требований для системы учета долгов могут включать в себя:

* Система должна позволять пользователям удобно управлять своими долгами и кредитами.
* Система должна предоставлять уведомления о предстоящих платежах и сроках возврата долгов.
* Система должна обеспечивать безопасное хранение финансовых данных пользователей.
* Система должна быть доступна на мобильных устройствах для удобства пользования пользователями в любое время и в любом месте.

1. **Функциональные требования.**

Функциональные требования - это спецификация того, что должна делать система, то есть они определяют функции, сервисы или операции, которые система должна выполнять. Они описывают входные данные, поведение системы и ожидаемые результаты. Функциональные требования формулируются на основе потребностей пользователей и бизнес-задач, которые система должна решать.

Основные характеристики функциональных требований:

* Определение функций системы: Они определяют, какие функции или операции должны быть доступны в системе.
* Взаимодействие с пользователем: Функциональные требования описывают, как система должна взаимодействовать с пользователями и как пользователи будут взаимодействовать с системой.
* Описание входных и выходных данных: Они определяют формат и структуру входных данных, которые система будет принимать, а также ожидаемые выходные данные или результаты операций.
* Условия и ограничения: Функциональные требования могут также включать в себя условия и ограничения, в которых должны выполняться функции системы.
* Поведение системы: Они описывают ожидаемое поведение системы в различных сценариях использования.

Примеры функциональных требований для системы учета долгов могут включать в себя:

* Пользователь должен иметь возможность добавлять новые долговые отношения в систему, указывая информацию о сумме долга, сроках возврата и другие данные.
* Система должна отправлять уведомления пользователю о предстоящих сроках погашения долгов.
* Пользователь должен иметь возможность просматривать список своих долгов и их статусы (например, оплачен, неоплачен, просрочен).
* Система должна позволять пользователю отмечать долги как оплаченные и вносить информацию об оплате.

1. **Технические требования.**

Технические требования - это спецификация того, как система должна быть реализована, а также какие технологии, стандарты и архитектурные решения должны быть использованы для достижения заданных функциональных и нефункциональных характеристик системы.

Основные характеристики технических требований:

* Технологические решения: Они определяют используемые технологии, программные средства и инструменты, необходимые для разработки и реализации системы.
* Архитектурные принципы: Технические требования могут включать в себя описание архитектурных принципов и паттернов, которые должны быть использованы при разработке системы.
* Интеграция: Они описывают требования к интеграции системы с другими системами или компонентами.
* Производительность: Технические требования могут включать в себя требования к производительности системы, такие как время отклика, пропускная способность и другие характеристики производительности.
* Безопасность: Они определяют требования к безопасности системы, включая защиту данных, аутентификацию и авторизацию, шифрование и другие аспекты безопасности.
* Надежность и отказоустойчивость: Они описывают требования к надежности и отказоустойчивости системы, включая обработку ошибок, резервное копирование данных и восстановление после сбоев.
* Масштабируемость и расширяемость: Они определяют требования к масштабируемости и расширяемости системы, позволяющие ей эффективно масштабироваться при увеличении объема данных или нагрузки, а также легко расширяться за счет добавления новых функций или компонентов.

Примеры технических требований для системы учета долгов могут включать в себя:

* Система должна быть разработана с использованием языка программирования Java и фреймворка Spring.Для хранения данных должна использоваться реляционная база данных MySQL.

1. **Технологии и методологии управления требованиями.**

Технологии и методологии управления требованиями - это набор подходов, инструментов, методов и практик, используемых для эффективного управления жизненным циклом требований от их сбора и анализа до их реализации и управления изменениями. Эти технологии и методологии помогают организациям разрабатывать и поддерживать продукты и системы, которые соответствуют потребностям пользователей и бизнес-задачам.

Вот несколько основных технологий и методологий управления требованиями:

* Моделирование требований: Использование графических моделей, таких как диаграммы Use Case, диаграммы классов или диаграммы последовательностей, для визуализации и документирования требований.
* Системы управления требованиями (RM): Использование специализированных инструментов и платформ для сбора, анализа, документирования и управления требованиями в течение всего их жизненного цикла.
* Методология Agile: Agile-методологии, такие как Scrum или Kanban, предлагают гибкий и итеративный подход к управлению требованиями, позволяющий быстро реагировать на изменения и обеспечивать более гибкую разработку продуктов.
* Методология Waterfall: Водопадная методология предполагает последовательное выполнение этапов разработки, включая сбор и анализ требований, разработку, тестирование и внедрение. Это подходит для проектов с четко определенными и неизменными требованиями.
* Прототипирование и проблемно-ориентированное проектирование (Problem-Driven Design, PDD): Создание прототипов и применение методологии PDD помогает более эффективно понять потребности пользователей и выявить новые требования в процессе разработки.
* Стандарты и нормативы: Существуют стандарты и рекомендации, такие как IREB (Международный совет по инженерии требований), BABOK (Руководство к анализу бизнес-требований), которые предлагают наборы лучших практик и рекомендаций по управлению требованиями.
* Управление изменениями: Процессы и инструменты управления изменениями требований, такие как системы отслеживания изменений и ревизий требований, помогают эффективно управлять изменениями в требованиях в течение жизненного цикла проекта.
* Тестирование требований: Процессы тестирования требований, такие как верификация и валидация требований, позволяют проверить соответствие реализации требованиям и обеспечить качество конечного продукта.

Эти методы и инструменты часто комбинируются и адаптируются в зависимости от конкретных потребностей и контекста проекта.

Контрольные вопросы № 2

1. **Моделирование информационных систем.**   
   Разработка информационной системы невозможна без ее тщательного проектирования: слишком велико влияние этого шага на 17 последующие этапы жизненного цикла информационной системы, в основе которой лежит создаваемая база данных.
2. **Виды моделей.**  
   Для целей проектирования информационной системы могут быть использованы следующие виды моделей:
   * 1. методология функционального моделирования работ SADT (Structured Analysis and Design Technique);
     2. диаграммы потоков данных DFD (Data Flow Diagrams);
     3. методология объектного проектирования на языке UML (UMLдиаграммы).
3. **Структурные модели ИС.**
4. **Объектно-ориентированный анализ и проектирование.**
5. **Технологии, языки и средства моделирования.**
   * 1. Методология SADT (Structured Analisys and Design Technique - технология структурного анализа и проектирования) разработана Дугласом Т. Россом и является одной из самых известных и широко используемых методик проектирования. Новое название методики, принятое в качестве стандарта, -IDEF0 (Icam DEFinition) является частью программы ICAM (Integrated Computer -Aided Manufacturing - интегрированная компьютеризация производства).
     2. В IDEF0 система представляется как совокупность взаимодействующих работ (или функций). Связи между работами определяют технологический процесс или структуру взаимосвязи внутри организации. Модель SADT представляет собой серию диаграмм, разбивающих сложный объект на составные части.
6. **Язык унифицированного моделирования UML.**   
   Унифицированный язык моделирования UML (Universal Modeling Language) – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех компонентов, создаваемых при разработке программных систем. Язык UML является объектно-ориентированным языком.
7. **Диаграммы языка UML: структурные диаграммы, диаграммы поведения, диаграммы взаимодействия.**   
   Диаграмма (Diagram) — это графическое представление множества элементов. Чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями).
8. **Инструментальные средства моделирования ИС.**
9. **Применение UML при проектировании ИС.**  
   UML был разработан компанией Rational Software с целью создания наиболее оптимального и универсального языка для описания как предметной области, так и конкретной задачи в программировании. Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее обшей и абстрактной концептуальной модели системы к логической, а затем и к физической модели соответствующей системы. Любая задача, таким образом, моделируется при помощи некоторого набора иерархических диаграмм, каждая из которых представляет собой некоторую проекцию системы.