**ВОПРОСЫ К ЭКЗАМЕНУ**

**по дисциплине Системная и программная инженерия**

**Знать:**

1. Дайте определение понятия «Системная инженерия»
2. Дайте определение понятия «Программная инженерия»

Программная инженерия — приложение систематического, дисциплинированного, измеримого подхода к разработке, функционированию и сопровождению программного обеспечения, а также исследованию этих подходов; то есть, приложение дисциплины инженерии к программному обеспечению

1. Дайте определение понятия «Система». Строение системы: элемент, компонент, связь

Элемент — предел членения системы с точки зрения аспекта рассмотрения, решения конкретной задачи, поставленной цели. Компонент, подсистема — относительно независимая часть системы, обладающая свойствами системы, и в частности, имеющая подцель. Связь, отношение — ограничение степени свободы элементов: элементы, вступая в связь друг с другом, утрачивают часть свойств или степеней свободы, которыми они потенциально обладали;

1. Основные положения системной инженерию по А. Холлу.
2. Перечислите основные свойства систем

Основные свойства систем: эмерджентность, целостность, организованность, функциональность, наличие поведения.

1. Определение понятия «жизненный цикл системы».

Жизненный цикл — стадии процесса развития системы, начиная с момента возникновения необходимости в такой системе и заканчивая её исчезновением.

1. Какие существуют виды систем в программной инженерии?

Виды систем в программной инженерии:

По типу:

1. Операционные системы (ОС):- Управляют ресурсами компьютера, предоставляют интерфейс для пользователей и приложений.- Примеры: Windows, macOS, Linux, Android, iOS.

2. Прикладные системы:- Решают конкретные задачи пользователей, работают над определенными данными.- Примеры: офисные приложения (Word, Excel), браузеры, игры, системы управления базами данных (СУБД), системы управления версиями (Git).

3. Встроенные системы:- Специализированные системы, интегрированные в устройства - Примеры: системы управления автомобилем, системы управления бытовой техникой, системы контроля доступа.

4. Распределенные системы:- Состоят из нескольких узлов, работающих совместно.- Примеры: интернет-магазины, системы онлайн-банкинга, сети социальных медиа.

5. Веб-приложения:- Доступные через веб-браузеры, обычно работают на серверах.- Примеры: сайты, интернет-магазины, онлайн-игры, облачные сервисы.

6. Мобильные приложения:- Разработаны для мобильных устройств.- Примеры: приложения для социальных сетей, приложения для заказа еды, игры, приложения для здоровья.

7. Системы искусственного интеллекта (ИИ):

- Системы, способные имитировать человеческий интеллект.- Примеры: системы распознавания речи, системы машинного обучения, чат-боты.

8. Системы машинного обучения (МО):- Системы, которые обучаются на данных и могут делать прогнозы.- Примеры: системы прогнозирования продаж, системы обнаружения мошенничества, системы распознавания образов.

По размеру:1. Малые системы:- Относительно простые, с небольшим количеством компонентов.

2. Средние системы:- Более сложные, чем малые, но все еще управляемы.

3. Большие системы:- Крупные, сложные системы с большим количеством компонентов.

4. Гигантские системы:- Наиболее сложные и масштабные системы, часто включающие в себя множество подсистем

1. Что понимается под тестированием программного обеспечения.

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом

1. Дайте определение процесса тестирования программного обеспечения. Жизненный цикл дефекта.

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом

1. Дайте определение процесса тестирования. Функциональное тестирование.

Тестирование — это одна из техник контроля качества, включающая в себя активности по :

планированию работ (Test Management),

проектированию тестов (Test Design),

выполнению тестирования (Test Execution)

анализу полученных результатов (Test Analysis)

**Функциональное тестирование** рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом.

**Преимущества функционального тестирования**:

* имитирует фактическое использование системы.

**Недостатки функционального тестирования**:

* возможность упущения логических ошибок в программном обеспечении;
* вероятность избыточного тестирования.

1. Дайте определение процесса тестирования. Нагрузочное тестирование.

**Нагрузочное тестирование** — это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем (разделяемом ими) ресурсе.

1. Дайте определение процесса тестирования. Тестирование на отказоустойчивость.

Тестирование на отказоустойчивость (Fault Tolerance Testing) — это процесс проверки надежности и стабильности программного обеспечения при возникновении сбоев, ошибок или непредвиденных ситуаций. Цель данного вида тестирования — определить способность системы продолжать работу и восстанавливаться после возникновения проблем.

1. Дайте определение процесса тестирования. Повторное и регрессионное тестирование.

**Регрессионное тестирование** — это вид тестирования направленный на проверку изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб сервер или сервер приложения), для подтверждения того факта, что существующая ранее функциональность работает как и прежде.

**Регрессия багов** (**Bug regression**) - попытка доказать, что исправленная ошибка на самом деле не исправлена.

**Регрессия старых багов** (**Old bugs regression**) - попытка доказать, что недавнее изменение кода или данных сломало исправление старых ошибок, т.е. старые баги стали снова воспроизводиться.

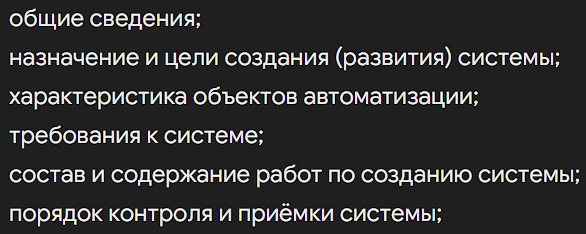
**Регрессия побочного эффекта** (**Side effect regression**) - попытка доказать, что недавнее изменение кода или данных сломало другие части разрабатываемого приложения.

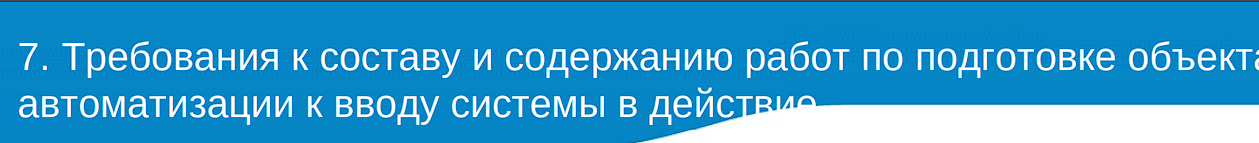
**Повторное тестирование** — тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок

1. Что означает компонентное тестирование.

Компонентное (модульное) тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т. д.)

1. Назовите группы основных разделов Технического задания по ГОСТ 34.602-20.



1. Какие требования включаются в раздел технических требований Технического задания.
2. Назовите основные характеристики требований.

Выделяют 9 критериев качества требований: 1) корректность; 2) недвусмысленность; 3) полнота; 4) непротиворечивость; 5) упорядоченность по важности и стабильности; 6) проверяемость; 7) модифицируемость; 8) трассируемость;

1. Что такое нефункциональные требования.

Нефункциональные - условия выполнения ПО и доступа к данным.

1. Что такое функциональные требования.

Функциональные требования определяют назначение и функции системы

1. Назовите основные методы структурного анализа и проектирования.

DFD (Data Flow Diagrams) — диаграммы потоков данных

* STD (State Transition Diagrams) — диаграммы перехода состояний,
* ERD (Entity-Relationship Diagrams) — диаграммы «сущность-связь» в нотациях Чена и Баркера;
* SADT (Structured Analysis and Design Technique) — технология структурного анализа и проектирования;
* семейство IDEF (Integration Definition for Function Modeling):

1. Что такое CASE-технологии.

CASE – это набор автоматизированных инструментов, которые применяются на всех этапах жизненного цикла разработки программного обеспечения. CASE-технологии помогают повысить эффективность, качество и производительность разработки.

Основные характеристики CASE-технологий:

• Автоматизация: CASE-инструменты автоматизируют рутинные задачи, освобождая разработчиков для творческой работы.

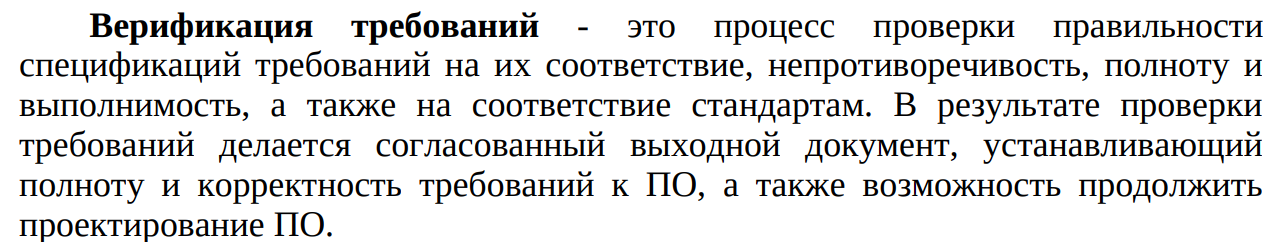
• Интеграция: CASE-инструменты интегрируются друг с другом, обеспечивая сквозной процесс разработки.

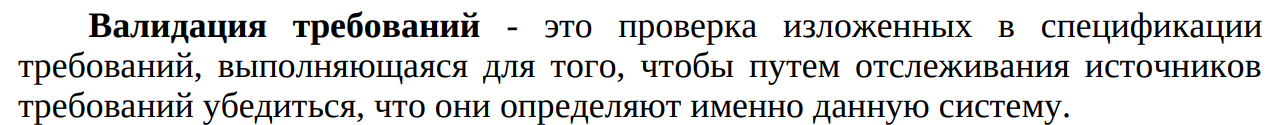
• Стандартизация: CASE-технологии способствуют стандартизации процессов разработки, повышая единообразие и согласованность.

• Визуализация: CASE-инструменты предоставляют визуальные средства для представления моделей, диаграмм и других артефактов разработки.

• Документирование: CASE-инструменты автоматически генерируют документацию по проекту, облегчая его сопровождение и развитие.

1. Модели «AS IS» и «TO BE». Валидация и верификация требований.





1. Что такое проектный риск

**Риск проекта** — это кумулятивный эффект вероятностей наступления неопределенных событий, способных оказать отрицательное или положительное влияние на цели проекта.

1. Что включает процесс идентификации рисков

**Идентификация рисков**—процесс определения рисков, способных повлиять на проект, и документирование их характеристик.

1. Какими методами проводится анализ рисков.

Матрица вериятностей и последствий, классификация рисков

1. Риски в информационных технологиях. Основные источники рисков.

Риски информационных технологий условно можно подразделить на ***риски информационной безопасности*** и ***риски информационных систем***.

1.

* риск остановки/сбоя информационных систем в результате атаки злоумышленников;
* риск утраты/искажения информации в результате заражения рабочих станций  вредоносным программным обеспечением ;
* риск утечки информации в результате использования шпионского оборудования;

2.

* риск сбоя/остановки информационных систем. Причиной могут послужить разные факторы, например, нехватка мощностей;
* риск искажения данных в результате некорректной интеграции между информационными системами;
* риск утраты данных в результате отсутствия процесса резервного копирования;

1. Управление рисками. Этапы управления рисками.

Этапы управления рисками:

Планирование

Идентификация факторов риска

Качественная оценка

Количественная оценка

Подготовка плана реакции

Мониторинг и контроль

1. Методы управления рисками: снижение риска, отказ от риска, передача риска, принятие риска.

снижение — принятие мер для уменьшения вероятности возникновения рисков или их воздействия;

Избегание — отказ от определенных действий или решений, которые могут создать опасность

Передача - переложение ответственности за управление нежелательными событиями на другие стороны, например, через страхование или контрактные договоры;

самоутверждение — осознанное принятие некоторых рисков, когда потенциальные выгоды превышают возможные убытки;

1. Институты по стандартизации: Росстандарт, ИСО, МЭК, МСЭ, IEEE. Область работы. Основные направления работы.
2. Стандартизация в области программной инженерии. Задачи органов по стандартизации. Состав технического комитета.
3. Мониторинг программных систем. Задачи мониторинга.

Мониторинг программных систем — это непрерывный процесс наблюдения за их состоянием, собирая и анализируя данные о работе системы для выявления отклонений, проблем и оптимизации производительности.

1. Понятия SLA, SLO и SLI. Соглашение об уровне сервиса, критерии и показатели уровня сервиса.

SLA (Service Level Agreement) — Соглашение об уровне сервиса — это договор между поставщиком услуг и клиентом, который определяет уровень качества предоставляемых услуг.

SLO (Service Level Objective) — Цель уровня сервиса — это конкретные, измеримые показатели, определяющие желаемый уровень качества предоставляемых услуг.

SLI (Service Level Indicator) — Показатель уровня сервиса — это метрика, которая используется для измерения фактического уровня сервиса.

Типы критериев уровня сервиса:

• Доступность: • Производительность• Безопасность

• Качество: • Надежность:

Типы показателей уровня сервиса:

• Среднее время отклика• Пропускная способность: • Время простоя: • Количество активных пользователей.

1. Виды документации на программную систему. Техническая документация.
2. Виды документации на программную систему. Проектная документация.
3. Виды документации на программную систему. Пользовательская документация

Основные виды документации:

1. Техническая документация:

- Спецификация требований: Описывает функциональность, требования к производительности, безопасности, совместимости, удобству использования и другим аспектам системы.

- Архитектурная документация: Описывает общую структуру системы, ее компоненты, взаимодействие между ними, а также используемые технологии.

- Документация по коду: Комментарии в коде, документация по API, схемы баз данных, диаграммы UML (Unified Modeling Language).

- Документация по тестированию: Планы тестирования, сценарии, отчеты о результатах тестирования.

- Документация по развертыванию: Руководство по установке, конфигурации и запуску системы.

- Документация по эксплуатации: Руководство по обслуживанию системы, инструкции по мониторингу, диагностике и устранению неисправностей.

- Документация по безопасности: Политика безопасности, описание угроз, меры по защите системы от атак.

2. Пользовательская документация:

- Руководство пользователя: Описывает, как использовать систему, содержит инструкции по работе с интерфейсом, функциональными возможностями и настройками.

- Справочник: Содержит подробную информацию о функциях системы, описания команд и параметров.

- Обучающие материалы: Курсы, видеоуроки, презентации, которые помогают пользователям освоить систему.

3. Проектная документация:

- Техническое задание: Описывает цели, задачи, требования к системе, сроки и бюджет.

- План проекта: Описывает этапы проекта, ответственных исполнителей, ресурсы, графики и контрольные точки.

- Отчеты по проекту: Документы о прогрессе проекта, выявленных проблемах, принятых решениях, расходах и других важных аспектах.

1. Определение инцидента. Управление инцидентами. Отчет об инциденте.

Инцидент — это любое событие, которое нарушает нормальную работу системы, сервиса или процесса. Это может быть сбой в работе приложения, отказ оборудования

Управление инцидентами — это процесс выявления, анализа, реагирования и решения инцидентов, чтобы минимизировать их влияние на бизнес и пользователей.

1. Какие специалисты принимают участие в создании автоматизированной системы.

Аналитики: Проектировщики: Разработчики: Специалисты по безопасности: Специалисты по документации: Специалисты по обучению: Менеджмент проекта:

1. Организация процесса разработки: назначение систем управления версиями. Примеры систем контроля версий

Системы управления версиями (VCS - Version Control Systems) - это набор инструментов, которые помогают разработчикам отслеживать изменения в исходном коде и других файлах проекта.

Сохранение истории изменений: Сотрудничество разработчиков: Управление ветками: Откат к предыдущим версиям: Просмотр изменений: GIT SVN

1. Назовите основные стадии жизненного цикла автоматизированной

системы.

1. Инициация (Планирование):

2. Анализ:

3. Проектирование:

4. Разработка:

5. Тестирование:

6. Развертывание:

7. Эксплуатация:

8. Вывод из эксплуатации:

1. Назовите стадии работ по созданию автоматизированной системы.

. Анализ требований: . Проектирование: . Разработка: Развертывание: . Тестирование: Эксплуатация: Вывод из эксплуатации:

1. Назовите основные процессы, входящие в жизненный цикл автоматизированной системы.
2. Назовите этапы и стадии создания АС по ГОСТ 34.601-90.

Этапы:

* 1. Предпроектный этап: 2. Проектный этап: . Этап разработки: 4. Этап внедрения: 5. Этап эксплуатации: 6. Этап вывода из эксплуатации:

В рамках каждого этапа, кроме этапа внедрения, выделяются следующие стадии:

• Техническое задание: • Технический проект: • Рабочая документация:

1. Математическое обеспечение АС. Понятия, относящиеся к данному виду обеспечения.

Математическое обеспечение (МО) автоматизированной системы (АС) - это совокупность алгоритмов, программ, моделей и методик, которые обеспечивают выполнение поставленных задач системы.

1. Программное обеспечение АС. Понятия, относящиеся к данному виду обеспечения.

Программное обеспечение (ПО) автоматизированной системы (АС) - это набор инструкций, программ и данных, которые управляют работой аппаратных средств и обеспечивают выполнение заданных функций системы.

• Прикладное ПО: Программные продукты, предназначенные для решения конкретных задач в определенной предметной области.

• Системное ПО: Программное обеспечение, обеспечивающее функционирование ОС, управление ресурсами и взаимодействие различных программных модулей.

• Инструментальное ПО: Программные инструменты, которые используются для разработки, тестирования и отладки программного обеспечения.

44. Информационное обеспечение АС. Понятия, относящиеся к данному виду обеспечения.

Информационное обеспечение (ИО) автоматизированной системы (АС) - это совокупность данных, информации и знаний, необходимых для функционирования системы и достижения ее целей.

1. Лингвистическое обеспечение АС. Понятия, относящиеся к данному виду обеспечения.

Лингвистическое обеспечение (ЛО) автоматизированной системы (АС) - это совокупность языковых средств, правил и алгоритмов, которые обеспечивают взаимодействие между пользователем и системой, а также обработку текстовой информации.

1. Техническое обеспечение АС. Понятия, относящиеся к данному виду обеспечения.

Техническое обеспечение АСУ представляет собой целый комплекс технических средств и инструментов, которые используются для обеспечения и поддержания работы автоматизированной системы управления.

**Уметь:**

1. Составьте матрицу требований для системы интернет-магазина.
2. Составьте матрицу рисков для системы интернет-магазина.
3. Составьте матрицу требований для системы социальной сети.
4. Составьте матрицу рисков для системы социальной сети.
5. Составьте матрицу требований для системы вызова такси через мобильное приложение.
6. Составьте матрицу рисков для системы вызова такси через мобильное приложение.
7. Изобразите информационную диаграмму для системы интернет- магазина.
8. Изобразите информационную диаграмму для системы социальной сети.
9. Изобразите информационную диаграмму вызова такси через мобильное приложение.
10. Изобразите структурную диаграмму системы интернет-магазина.
11. Изобразите структурную диаграмму системы социальной сети.
12. Изобразите структурную диаграмму системы вызова такси через мобильное приложение.
13. Определите модель жизненного цикла для разработки системы

«мессенджер». Распишите необходимые роли для реализации проекта по созданию данной системы.

1. Определите перечень программного обеспечения, необходимого для поддержки разных этапов жизненного цикла системы «мессенджер».
2. Опишите тестовый случай для проверки функционала «отправка сообщения» в системе «мессенджер».
3. Сформулируйте цель создания системы «мессенджер».
4. Перечислите задачи создания системы «мессенджер».
5. Определите модель жизненного цикла для разработки системы «сайт дистанционного обучения». Распишите необходимые роли для реализации проекта по созданию данной системы.
6. Определите перечень программного обеспечения, необходимого для поддержки разных этапов жизненного цикла системы «сайт дистанционного обучения».
7. Опишите тестовый случай для проверки функционала «загрузка работы на проверку» в системе «сайт дистанционного обучения».
8. Сформулируйте цель создания системы «сайт дистанционного обучения».
9. Перечислите задачи создания системы «сайт дистанционного обучения».
10. Определите модель жизненного цикла для разработки системы

«интернет-магазин». Распишите необходимые роли для реализации проекта по созданию данной системы.

1. Определите перечень программного обеспечения, необходимого для поддержки разных этапов жизненного цикла системы «интернет-

магазин».

1. Опишите тестовый случай для проверки функционала «оплата картой» в системе «интернет-магазин».
2. Сформулируйте цель создания системы «интернет-магазин».
3. Перечислите задачи создания системы «интернет-магазин».

**Владеть:**

1. Обоснуйте различия между большими и малыми системами.

1. Масштаб и сложность:

2. Функциональность

3. Ресурсы:

4. Риски:

5. Влияние на бизнес

1. Обоснуйте выбор типа системы. В чем различия между автоматическими, автоматизированными, информационными системами.

1. Автоматические системы:

• Полная автоматизация процессов: Работают без участия человека, выполняя все операции самостоятельно.

• Высокая точность и скорость: Минимизируют ошибки, характерные для человеческого фактора, и повышают скорость выполнения задач.

2. Автоматизированные системы:

• Частичная автоматизация: Сочетают автоматические и ручные операции.

• Помощь человеку: Снижают нагрузку на человека, автоматизируют рутинные задачи, предоставляют информацию для принятия решений.

3. Информационные системы:

• Сбор, обработка и хранение информации: Предоставляют информацию для анализа, планирования и принятия решений.

• Не автоматизируют процессы: Не заменяют человека, а предоставляют ему инструменты для работы с информацией.

1. Обоснуйте различия между сложной и большой системой. Приведите примеры систем такого типа.

1. Размер:

• Большая система: Характеризуется большим количеством компонентов, объемом данных, количеством пользователей и географическим охватом. Она охватывает широкий спектр функций и процессов, как правило, включает множество взаимодействующих подсистем.

• Сложная система: Может быть небольшой по размеру, но ее внутренняя структура и взаимодействие элементов могут быть очень сложными.

2. Сложность взаимодействия:

• Большая система: Сложность связана с управлением большим количеством компонентов, координацией действий разных команд, обеспечением интеграции между подсистемами, а также с учетом многочисленных внешних факторов и изменений.

• Сложная система: Сложность обусловлена нелинейными взаимодействиями между элементами, возникновением непредсказуемых эффектов и сложностью анализа ее поведения.

3. Пример:

• Большая система: Система управления воздушным движением (ATC) - охватывает большую территорию, имеет множество компонентов, работает с огромным объемом данных и управляет действиями многих пользователей.

• Сложная система: Биологическая клетка - состоит из ограниченного количества элементов, но их взаимодействие нелинейно и очень сложно для анализа.

1. Определите цель создания системы интернет-магазина. Поставьте задачи по созданию данной системы.

Цель: Создание системы интернет-магазина направлено на увеличение продаж и расширение рынка сбыта путем предоставления покупателям удобного и доступного онлайн-сервиса для покупки товаров.

## Задачи по созданию системы интернет-магазина:

1. Функциональные задачи:

• Разработка веб-сайта: Создание привлекательного и удобного сайта с интуитивно понятной навигацией, информативным описанием товаров, качественными изображениями, возможностью поиска и фильтрации.

• Реализация корзины покупок: Обеспечение удобного добавления товаров в корзину, просмотра и редактирования списка покупок, расчета стоимости заказа.

• Система оплаты: Интеграция с платежными системами для безопасной оплаты заказов.

• Система доставки: Выбор способа доставки, отслеживание статуса заказа, возможность самовывоза.

• Управление товарами: Добавление, редактирование, удаление товаров, управление запасами, ценами, скидками.

• Управление заказами: Прием, обработка, отслеживание заказов, формирование отчетности.

• Управление пользователями: Регистрация, авторизация, создание профилей, управление личной информацией.

• Интеграция с CRM-системой: Создание единой базы данных о клиентах, создание маркетинговых кампаний, аналитика продаж.

2. Нефункциональные задачи

• Обеспечение безопасности данных: Защита от несанкционированного доступа, защита от мошенничества, соответствие требованиям законодательства.

• Надежность работы системы: Гарантия доступности сайта, скорость работы, устойчивость к ошибкам.

• Масштабируемость: Возможность расширения системы в соответствии с ростом объема продаж и количества пользователей.

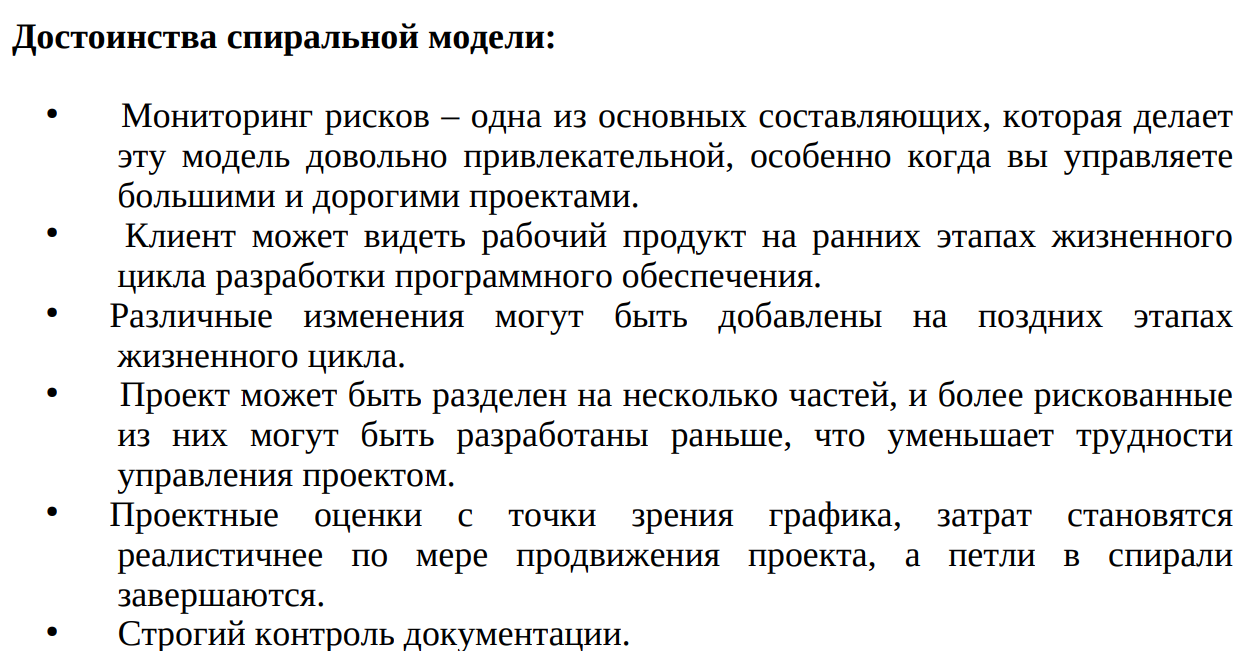
• Удобство использования: Создание интуитивно понятного и удобного интерфейса для покупателей и администраторов системы.

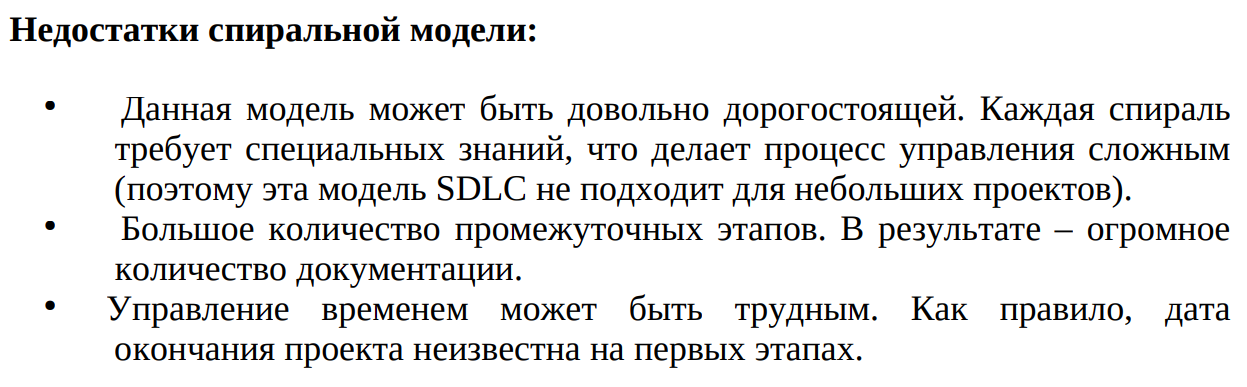
• Соответствие требованиям законодательства: Соблюдение законодательных норм в сфере электронной коммерции, защиты данных и конфиденциальности.

• Мобильная версия: Создание мобильной версии сайта для удобного доступа с мобильных устройств.

1. Определите цель создания системы социальной сети. Поставьте задачи по созданию данной системы.
2. Определите цель создания системы вызова такси через мобильное приложение. Поставьте задачи по созданию данной системы.
3. Обоснуйте, почему процесс проектирования считается последовательной реализацией требований, предъявляемых к системе?
4. Опишите основные риски на этапах создания системы.
5. Опишите различия автоматизированных систем управления от других автоматизированных систем.
6. Опишите содержание плана тестирования.
7. Опишите роль и задачи следующего специалиста, принимающего участие в разработке вычислительных систем: Разработчик программного обеспечения.
8. Опишите роль и задачи следующего специалиста, принимающего участие в разработке вычислительных систем: Проектный менеджер.
9. Опишите роль и задачи следующего специалиста, принимающего участие в разработке вычислительных систем: Владелец продукта (только в гибких методологиях).
10. Опишите роль и задачи следующего специалиста, принимающего участие в разработке вычислительных систем: Инженер по качеству (QA – quality assurance).
11. Опишите роль и задачи следующего специалиста, принимающего участие в разработке вычислительных систем: Специалист по тестированию (QС – quality сontrol).
12. Опишите основные причины создания неэффективных автоматизированных систем.
13. Опишите стадию концептуальной разработки системы. Приведите примеры основных работ, выполняемых на данной стадии.
14. Опишите стадию технической разработки системы. Приведите примеры основных работ, выполняемых на данной стадии.
15. Опишите стадию пост-разработки системы. Приведите примеры основных работ, выполняемых на данной стадии.
16. Опишите стадию технического проектирования системы. Приведите примеры основных работ, выполняемых на данной стадии.
17. Каскадная модель жизненного цикла ПО. Опишите преимущества и недостатки каскадной модели жизненного цикла ПО.

22.V-модель жизненного цикла ПО. Опишите преимущества и недостатки V модели жизненного цикла ПО.

1. Спиральная модель жизненного цикла ПО. Опишите преимущества и недостатки спиральной модели жизненного цикла ПО. 



1. Опишите Преимущества и недостатки гибких методологий разработки программного обеспечения.
2. Опишите процесс сопровождения программных систем: назначение систем управления конфигурациями. Примеры систем контроля конфигураций.
3. Опишите процесс технической поддержки. Уровни технической поддержки. Таблица эскалации.
4. Опишите процесс сбора требований к системе. Определите источники требований к системе.