Лабораторная работа

**Модель качества информационной системы**

**Цель работы**: построить модель качества информационной системы.

1. Сайт страховой медицинской компании МАКС-М
2. Лица, заинтересованные в качестве данной информационной системы
   * Клиенты организации, в том числе потенциальные
   * Сотрудники организации
   * Разработчики
   * Заказчик
3. Функциональные требования

Посетитель сайта, потенциальный клиент.

* Как посетитель сайта, я хочу иметь возможность оставить электронную заявку на оформление полиса ОМС, чтобы сэкономить своё время, избавившись от необходимости посещать организацию до выяснения детальных условий оказания услуги и упростить процесс получения услуги в целом
* Как посетитель сайта, при формировании заявки я хочу указать свои ФИО для того, чтобы сотрудники организации могли меня идентифицировать
* Как посетитель сайта, при формировании заявки, я хочу иметь возможность указать свои контактные данные, чтобы я мог получить обратную связь от организации
* Как посетитель сайта, при формировании заявки я хочу иметь возможность оставить комментарий, чтобы уточнить детали, которые я считаю важными
* Как посетитель сайта, я хочу иметь возможность получить контактные данные организации для поддержания связи в процессе оказания услуги, или для уточнения некоторых деталей
* Как посетитель сайта, я хочу иметь возможность скачать и просмотреть лицензию на осуществление деятельности организации, чтобы убедиться, что она имеет права на предоставления услуги, прежде чем связываться с данной организацией

Сотрудник организации, участвующий в предоставлении услуг клиентам компании.

* Как сотрудник организации, я хочу иметь возможность просматривать оставленные посетителями электронные заявки, чтобы связываться с ними и осуществлять процесс оказания услуги
* Как сотрудник организации, я хочу иметь возможность сортировать заявки, чтобы отличать новые заявки от уже рассмотренных

Организация.

* Как организация, мы хотим размещать на сайте информацию о нашей деятельности для ознакомления клиентов с нашей историей и опытом, формируя у клиентов представление о нас
* Как организация, мы хотим публиковать на сайте электронные версии правоустанавливающих документов (аккредитация, лицензия и т.д.), чтобы формировать доверие у потенциальных клиентов, тем самым способствуя формированию положительной репутации компании

Нефункциональные требования

* Кросс-браузерность. Веб-сайт должен корректно функционировать и идентично отображаться во всех часто-используемых браузерах.
* Текст на сайте не должен выходить за рамки содержащего его контейнера, а также не должен обрезаться.
* Компоненты графического пользовательского интерфейса не должны перекрывать друг друга, ограничивая доступ к просмотру контента сайта.
* Ссылки в навигационном меню должны вести на соответствующие разделы сайта
* Форма отправки заявки на оформление полиса ОМС должна работать корректно

3. Дайте подробное описание каждого атрибута качества в зависимости от специфики ИС.

**Функциональность:**

* **Способность к взаимодействию –** способность взаимодействовать с другими системами и средами. Примеры:
  + возможность обмена данными с электронными медицинскими картами, другими системами
* **Функциональная пригодность –** соответствие функций, реализуемых системой, поставленным задачам. Функционал информационный системы должен решать бизнес-задачи в соответствии с назначением ПО и техническим заданием. Примеры:
  + Функционал записи на приём к врачу
  + Функционал доступа пользователей к своим медицинским данным (результаты анализов и т.п.)
  + Отправка электронной заявки на оформление полиса ОМС
* **Защищенность –** способность предотвращать несанкционированный доступ к данным. Примеры:
  + Аутентификация пользователей для доступа к личным данным
  + Хранение и передача данных пользователей в зашифрованном виде
* **Точность** – способность системы выдавать достоверные данные
  + обеспечение актуальности и точность медицинской информации, предоставляемой на сайте (информация о врачах, услугах и ценах)
  + регулярное обновление контента сайта

**Надежность:**

* **Зрелость -** измеряет время, в течение которого система доступна для использования, среднее время бесперебойной работы. Примеры
  + сайт компании должен быть недоступен для пользователей не более некоторой малой доли времени от периода работы.
* **Устойчивость к отказам -** способность поддерживать работоспособность на заданном уровне при программных ошибках и при неправильных действиях пользователя, т.е. действиях, не соответствующих предусмотренному пользовательскому интерфейсу
* **Способность к восстановлению –** способность программы восстанавливаться после сбоев и неполадок. Восстановление может включать в себя восстановление данных или переключение на резервные ресурсы. Примеры:
  + за счёт регулярного создания резервных копий клиентских данных, в случае сбоя или утечки данных, компания сможет быстро восстановить информацию (учётные данные пользователей для авторизации, личные медицинские данные клиентов, их заявки на консультации и оформление полиса ОМС)

**Производительность:**

* **Эффективность использования ресурсов** - оценивает, уровень использования ресурсов программным обеспечением, таких как оперативная и долговременная память, процессорное время и сетевая пропускная способность.
* **Время отклика** - измеряет, сколько времени требуется программе для реагирования на запросы пользователя. Примеры:
  + малое время, необходимое для формирования http ответа сервером на клиентские запросы за счёт оптимизации бизнес-логики backend части сайта
* **Пропускная способность – определяет,** сколько операций программа может обработать за единицу времени. Высокая пропускная способность важна для приложений с большой нагрузкой. Примеры:
  + большая скорость ответа сервера на клиентские запросы позволяет обслуживать большее количество пользователей в единицу времени, что способствует более эффективному решению бизнес-задач (получить и рассмотреть большее количество заявок на оформление полиса ОМС)

**Безопасность:**

1. **Конфиденциальность**: Этот атрибут оценивает способность программы защищать конфиденциальные данные от несанкционированного доступа. Высокая конфиденциальность важна для систем, которые обрабатывают пользовательские данные.

**Удобство использования:**

* **Понятность -** оценивает, насколько легко пользователи могут понять, как использовать программу. Простой и понятный интерфейс повышает удобство использования. Примеры:
  + простая и логичная навигация по разделам сайта
  + понятная организация структуры (перечень полей ввода с метками) формы регистрации / отправки заявки на оформление полиса ОМС / записи на консультацию
* **Удобство обучения –** оценивает скорость обучения пользователей взаимодействию с ПО. Примеры:
  + логичный и понятный пользовательский интерфейс
  + наличие обучающих материалов, руководств или раздела часто задаваемых вопросов
  + внедрение контекстных подсказок (например, при наведении курсора на элемент)
* **Удобство работы –** количество усилий, прилагаемых пользователем для решения задач с помощью данного ПО. Примеры:
  + структура сайта должна позволять пользователям быстро находить нужную информацию
  + быстрый доступ к функционалу, который используется наиболее часто
* **Привлекательность –** способность ПО быть привлекательным для пользователей
  + цветовая палитра и шрифты должны быть приятными для глаз
  + соответствие дизайна сайта имиджу медицинской компании, создающее доверие у пользователей
  + использование уместной анимации элементов
  + использование качественных изображений

**Удобство сопровождения:**

* **Адаптируемость -** определяет, насколько легко система может быть изменена или модифицирована для удовлетворения новых требований
  + проектирование архитектуры сайта с использованием различных структурных паттернов (например MVC, Сервисный слой) для отделения бизнес-логики сайта от логики пользовательского интерфейса, а также для разделения различного функционала сайта на конкретные модули
* **Удобство проверки -** относится к легкости, с которой можно провести тестирование системы для подтверждения корректности её работы и соответствия требованиям
  + наличие автоматизированных тестов для проверки функциональности сайта
  + наличие четкой и доступной документации по тестовым случаям и сценариям использования
  + внедрение систем логирования, позволяющих отслеживать ошибки и отклонения в работе сайта
* **Анализируемость -** удобство проведения анализа недостатков, дефектов или ошибок, а также удобство определения модулей, требующих изменения и модернизации
* **Стабильность -** показатель, обратно пропорциональный риску возникновения непредвиденных эффектов при изменении или модернизации программы

**Переносимость:**

* Адаптируемость – атрибут, оценивающий усилия, затрачиваемый на адаптацию ПО к различным средам (программным, аппаратным)
* Удобство замены - возможность применения данного ПО вместо других программных систем для решения тех же задач в определенном окружении.
* Способность к сосуществованию - способность ПО сосуществовать с другими программами в общем окружении с общими ресурсами
* Удобство установки - определяет необходимые усилия для установки данного ПО в определённой среде;

Контрольные вопросы

**Какие стандарты качества используются для программных продуктов?**

ISO/IEC 25010 - Стандарт, описывающий характеристики качества программного обеспечения, включая функциональность, надежность, удобство использования, производительность, безопасность и поддерживаемость.

IEEE 829 - Стандарт для документирования тестирования программного обеспечения, который включает в себя различные виды тестовых документов.

Международный стандарт тестирования ПО (ISTQB) — это набор руководящих принципов и глоссарий терминов, который разработан Международным квалификационным советом по тестированию ПО (ISTQB). Этот стандарт описывает принципы, практики и процессы, которые используются при тестировании ПО, и является основой для сертификации тестировщиков на разных уровнях.

**Кто отвечает в проекте за качество разрабатываемого продукта?**

Аналитики - помогают определить требования и критерии качества на этапе сбора требований, что важно для дальнейшего тестирования.

Разработчики - также несут ответственность за качество своего кода, следуя лучшим практикам программирования и проводя юнит-тестирование.

Тестировщики (QA-инженеры) - занимаются тестированием ПО, выявлением дефектов и обеспечением соответствия требованиям.

Архитекторы ПО - обеспечивают качество на уровне архитектуры ИС, принимая решения о том, как организовать систему для достижения высоких стандартов качества.

**Кто кроме пользователей заинтересован в качестве программного продукта и почему?**

Разработчики, так как разработка качественного продукта с ранних этапов упрощает дальнейшие поддержку, масштабируемость и внедрение нового функционала в ПО, снижает затраты времени на исправление ошибок.

Команда поддержки, так как качественный продукт снижает количество обращений от пользователей

**Почему функциональность путают с удобством использования? (приведите примеры)**

Возможно, это происходит потому, что функциональность определяет соответствие реализуемых системой функций поставленным задачам. Наличие и удобство реализации предоставляемых системой функций в значительной степени определяют то, насколько пользователю будет удобно использовать данное ПО для решения своих задач.

**Чем отличается надёжность от безопасности? (приведите примеры)**

Надёжность оценивает устойчивость системы к отказам при возникновении программных ошибок, её способность к восстановлению после сбоев. Пример: восстановление пользовательских данных за счёт резервных копий. А безопасность оценивает способность системы защищать пользовательские данные от несанкционированного доступа. Пример: шифрование пользовательских данных при передаче.

**Каким образом можно оценить надёжность? Есть ли определённые метрики?**

Время безотказной работы - среднее время, в течение которого система работает без сбоев

Время восстановления - среднее время, необходимое для восстановления системы после сбоя

Доступность - процент времени, в течение которого система доступна для использования. Рассчитывается по формуле:

Время безотказной работы / (Время безотказной работы + Время восстановления) \* 100%

**Объясните взаимосвязь переносимости и совместимости (приведите примеры)**

Переносимость и совместимость ПО взаимосвязаны: высокое качество одного из этих атрибутов часто способствует улучшению другого. Переносимое ПО часто разрабатывается с учетом совместимости. Например, если приложение создано с использованием стандартных библиотек и API, оно будет более переносимым и совместимым с различными системами.

Для обеспечения переносимости необходимо тестировать приложение на различных платформах, что также улучшает его совместимость с другими системами