**R001. Интерфейс**

**R001-1. Приложение должно быть максимально похожим на notepad.**

- Проблема двусмысленности требования (Что значит “быть максимально похожим”? На какой именно notepad какой версии?) Необходимо указать конкретное приложение “notepad”.

**R001-2. Цвета фона и текста должны задаваться в меню.**

- Проблема завершённости требования. Следует описать меню подробнее.

**R001-2. Шрифты должны выбираться пользователями.**

- Проблема завершённости требования. Необходимо определить количество и набор шрифтов.

**R001-3. Кнопки должны утопать при наведении на них мышью.**

- Проблема завершённости требования. Что значит утопать? Визуально становиться ниже или становиться другого цвета? И то, и другое? Проблема корректности требования: это требование к дизайну, а не к функциональным особенностям ПП.

**R002. Функциональность**

**R002-1. Приложение должно уметь открывать, редактировать и сохранять файлы.**

- Лучше сформулировать требование так: “Приложение должно позволять пользователю …”. Проблема завершённости требования. Необходимо определить формат файла. Проблема атомарности: требование можно разбить на несколько и каждое из них описать детальнее.

**R002-2. Опционально должна быть возможность выбора кодировки символов.**

- Проблема завершённости требования. Необходимо определить конкретный список кодировок.

**R002-3. Приложение должно автоматически сохранять файл через заданный интервал времени, если файл не был сохранён.**

- Требование нарушает завершённость. Необходимо определить конкретный интервал времени.

**R003. Общие требования**

**R003-1. Приложение должно быть написано на C#.**

- Проблема завершённости требования. Необходимо определить платформу и версию языка.

**R003-2. Приложение должно быть удобным в использовании.**

- Проблема проверяемости требования. Невозможно написать тест, проверяющий требование. Проблема противоречивости требований: приложение, написанное на C# (требование R003-1) может потребовать установку дополнительных .NET библиотек, что может сказаться на удобстве использования.

Также в наборе требований присутствуют следующие нарушения:

- Набор требований не является прослеживаемым. У каждого требования должен быть уникальный идентификатор, по которому на это требования можно сослаться. Номер требования R001-2 повторяется два раза.

- Требования не проранжированны по важности. Для каждого требования должен быть указан уровень его важности

- Набор требований должен быть модифицируемым. Данные требования представлены в неудобной для обработки форме (не использованы инструменты управления требованиями, и в итоге команде, вероятно, в будущем придётся работать с огромными текстовыми файлами). Также присутствует некоторая избыточность, например слово “автоматически” в требовании R002-3 можно опустить, оно понятно из контекста.

**Ответы на контрольные вопросы:**

**1) Дайте определение требований:**

- Требования – это подробное изложение функционального наполнения системы.

**2) Опишите три уровня требований – в чём они выражаются и что описывают?**

- 1 уровень – Бизнес требования (Общее видение и обзорная документация): Зачем вообще нужен ПП и что с его помощью будет делаться.

- 2 уровень – Пользовательские требования (Use Cases): Задачи, которые пользователь может выполнять с помощью ПП, и реакцию ПП.

- 3 уровень – Функциональные и нефункциональные требования (Требования к ПО): Функциональные требования описывают поведение системы. Нефункциональные требования описывают специфические свойства системы, такие как: требования к удобству использования, безопасности, надёжности, расширяемости и т.д.

**3) Перечислите как можно больше свойств хороших требований.**

- Требование должно быть: завершённым, непротиворечивым, корректным, недвусмысленным, проверяемым.

**4) Какими свойствами должны обладать наборы требований?**

- Наборы требований должны быть: модифицируемыми, прослеживаемыми, проранжированными по важности.

**5) Какие проблемы чаще всего возникают при работе с требованиями? В чём их суть?**

- По завершённости. Упущены нефункциональные составляющие требования.

- По непротиворечивости. Требования противоречат явным образом друг другу.

- По корректности. Опечатки. Технически невыполнимые требования.

- По недвусмысленности. Требование должно однозначно трактоваться.

- По проверяемости. Должно быть возможно написать тест, проверяющий требование.

- По ранжированности. Выделены критические требования.

**6) Что такое тестовый план?**

- базовый документ, описывающий перечень тестируемых компонент, критические факторы и риски тестирования, ресурсы и график тестирования, тестовую стратегию и виды тестирования.

**7) Каковы основные свойства хорошего тестового плана?**

- полнота, корректность и недвусмысленность

- соответствие принятым стандартам

- тестовый подход должен быть осуществимым

- цели тестирования должны быть определены

- тестовое окружение должно быть описано

- описаны критерии прекращения тестирования

- критерии качества реалистичны

- определены продукты, подлежащие доставке

- стратегия и виды тестирования должны быть определены

- ресурсы выделены и описаны

- составлен реалистичный график тестирования

**8) Что такое тестовый сценарий (test case)?**

- описывает сценарии приемочного, критического и расширенного тестов, ожидаемые результаты и результат выполнения каждого сценария с возможными комментариями.

**9) Каковы характеристики хорошего тестового сценария?**

- все требования проверены и охвачены тестами

- все сценарии имеют уникальные идентификаторы и сгруппированы логически

- сценарии очевидны, корректны, ясны и сопоставимы

- набор тестов не должен быть избыточным, слишком простым или сложным

**10) Перечислите и опишите этапы жизненного цикла дефекта.**

- Зафиксирован – тестировщик записывает дефект в систему.

- Назначен – ведущий разработчик просматривает отчет об ошибке и назначает разработчика для устранения.

- Исправлен – разработчик сделал необходимые изменения, чтобы устранить ошибку.

- Проверен – тестировщик проверяет устранена ли действительно ошибка.

- Повторно обнаружен

- Отклонён

- Отложен

**11) Что такое баг?**

- это невыполнение требования, связанного с предполагаемым или установленным использованием.