**Анализ спецификаций, верификация и аттестация программного обеспечения**

*Часть 1: теоретические вопросы*

1. Определения:
   1. **Спецификация программного обеспечения:** Это документ, в котором описаны все требования к будущему ПО. Он включает в себя, что должна делать программа, как она должна работать, и какие характеристики она должна иметь. Это как рецепт.
   2. **Верификация:** Это процесс проверки того, что программа соответствует заявленным требованиям.
   3. **Аттестация:** Это процедура, которая подтверждает, что программа соответствует всем стандартам и готова к использованию. Это как гарант качества, говорящий о том, что продукт безопасен и надёжен.
   4. **Задача анализа спецификаций:** Это проверка того, что спецификация полная, не противоречивая, понятная и проверяемая.
   5. **Моделирование в процессе анализа спецификаций:** Это создание моделей или прототипов программы, чтобы проверить, правильно ли мы понимаем требования.
2. **Цели верификации:** Верификация важна, потому что она помогает найти ошибки на ранних этапах разработки. Чем раньше мы найдём ошибку, тем дешевле и проще будет её исправить. Например, верификация может помочь убедиться, что программа правильно обрабатывает ввод данных, или что она не вызывает сбоев при выполнении скрипта.
3. **Роль аттестации:** Аттестация необходима для определённых типов ПО, особенно для тех, что работают в критических областях. Аттестация подтверждает, что программа безопасна и надежна. Например, медицинское ПО должно пройти аттестацию, чтобы гарантировать, что оно не вызовет ошибок, которые могут навредить пациентам. Аттестация может быть обязательной (атомная энергетика) или добровольной (коммерческие цели).

*Часть 2: практические задания*

1. Анализ требований:

Требование 1: «Система должна обеспечивать безопасность пользовательских данных»

* 1. **Анализ:** Требование не очень конкретное. Нужно уточнить, какие именно меры безопасности должны быть реализованы (шифрование данных, контроль доступа).
  2. **Рекомендация:** Уточнить требование, например: «Система должна шифровать все пользовательские данные при передаче и хранении, а также контролировать доступ к данным на основе ролей пользователей»

Требование 2: «Приложение должно загружаться быстро»

1. **Анализ:** Требование не однозначное. Нужно указать конкретное время загрузки.
2. **Рекомендация:** Уточнить требование, например: «Приложение должно загружаться не дольше 3-х секунд на устройствах с процессором не ниже Intel Core i5»

Требование 3: «Пользователь должен иметь возможность восстановить доступ к учетной записи через электронную почту»

1. **Анализ:** Требование достаточно конкретное для понимания, но можно уточнить детали. Например, что должно произойти после отправки письма с инструкциями по восстановлению?
2. **Рекомендация:** Уточнить все требования, например: «Пользователь должен иметь возможность восстановить доступ к учетной записи через электронную почту. После отправки письма с инструкциями, пользователь должен получить уведомление о том, что письмо отправлено.»
3. Процедура верификации функции «восстановление пароля» в веб-приложении:
   1. Проверка функциональности:
      1. **Этап:** Отправка письма с инструкциями
      2. **Ошибки:** Письмо не отправляется, инструкции некорректны
      3. **Как избежать:** Проверить, что письмо отправляется и содержит правильные инструкции
   2. Проверка доступности:
      1. **Этап:** Переход по ссылке в письме
      2. **Ошибки:** Ссылка не работает, страница недоступна или отсутствует
      3. **Как избежать:** Убедиться, что ссылка активна и ведет на правильную страницу
   3. Проверка безопасности:
      1. **Этап:** Ввод нового пароля
      2. **Ошибки:** Пароль не соответствует требованиям безопасности, как следствие, данные не шифруются
      3. **Как избежать:** Проверить, что пароль соответствует требованиям и данные шифруются
   4. Проверка логики:
      1. **Этап:** Восстановление доступа
      2. **Ошибки:** Доступ не восстанавливается, ошибки в логике работы функции
      3. **Как избежать:** Протестировать функцию на разных сценариях использования
4. Оценка требований к аттестации медицинского приложения:
   1. Основные требования к безопасности и надежности:
      1. **Конфиденциальность данных:** Данные пациентов должны быть защищены от несанкционированного доступа
      2. **Надежность:** Приложение не должно терять данные и должно работать без сбоев
      3. **Соответствие стандартам:** Приложение должно соответствовать медицинским стандартам и требованиям регулирующих органов
   2. Документация и данные для аттестации:
      1. **Техническая документация:** Описание архитектуры приложения, исходный код и документация по безопасности
      2. **Результаты тестирования:** Отчеты о тестировании функциональности, безопасности и надежности
      3. **Политика безопасности:** Документы, описывающие меры безопасности, применяемые в приложении

*Часть 3: аналитика*

1. **Сравнение анализа спецификаций и верификации:** Анализ спецификаций проводится на этапе проектирования, чтобы убедиться, что требования корректны и понятны. Верификация проводится на этапе разработки, дабы убедиться, что программа соответствует этим требованиям. Важно разделять эти процессы, потому что анализ спецификаций помогает избежать ошибок на начальных этапах, а верификация помогает найти ошибки в уже разработанном коде.
2. **Риски недостаточной аттестации:** Недостаточная аттестация может привести к проблемам в работе ПО. Например, в медицине некачественная аттестация может привести к потере данных о пациентах или к ошибкам в диагностике. В авиации это может привести к сбоям в работе бортовых систем, что крайне опасно для жизни людей.
3. **Влияние на затраты:** Регулярная верификация и аттестация могут увеличить затраты на разработку, но они помогают избежать более крупных затрат на исправление ошибок на поздних этапах. Например, если ошибка в ПО для управления банковскими операциями не будет обнаружена, это может привести к потере денег как клиентов, так и компании, за чем следует потеря репутации.
4. **Анализ времени на тестирование:** Анализ спецификаций и грамотное распределение времени на верификацию и аттестацию могут сократить общий срок разработки ПО. Если заранее убедится, что требования корректны, проводятся регулярные проверки выполнения, то это поможет избежать многих ошибок, которые могут задержать выпуск продукта.
5. **Роль внешних аудитов:** Внешние аудиты и независимая аттестация повышают доверие к продукту, ибо они подтверждают, что продукт соответствует стандартам.
   1. **Плюсы внешних аудитов:** объективность и независимость
   2. **Минусы:** дополнительные затраты и возможные задержки

*Заключение*

Анализ спецификаций, верификация и аттестация - важные этапы в создании качественного, безопасного и соответствующего требованиям программного обеспечения. Они помогают убедиться, что программа соответствует требованиям, не содержит ошибок, да и в целом готова к использованию. Без этих процессов можно столкнуться с серьезными проблемами, которые могут привести к потере денег и репутации.