**Технология разработки программного обеспечения**

**Контрольные вопросы 2.**

**Управление требованиями к программному продукту.**

1. **Каково назначение этапа управления требованиями к программному продукту в жизненном цикле разработки программного продукта?** Этап управления требованиями направлен на систематическое выявление, документирование, анализ, согласование и контроль изменений требований в процессе разработки. Его основная задача — обеспечить чёткое понимание и выполнение требований заказчика, а также минимизировать риски, связанные с изменениями или недоразумениями.
2. **Какие действия включает в себя работа по управлению требованиями?**- Сбор и формализация требований от заинтересованных сторон.  
   - Анализ и уточнение требований для исключения противоречий и неоднозначностей.  
   - Документирование требований в соответствующей форме.  
   - Согласование требований с заказчиком и командой разработчиков.  
   - Отслеживание изменений требований и управление их версионированием.  
   - Контроль соответствия разработки текущим требованиям.  
   - Поддержка обратной связи и разрешение конфликтов, связанных с требованиями.
3. **Какие цели преследует работа по управлению требованиями?**- Обеспечение точного и единого понимания требований всеми участниками проекта.  
   - Предотвращение разночтений и ошибок на последующих этапах разработки.  
   - Снижение рисков, связанных с изменениями и неполнотами требований.  
   - Повышение качества конечного программного продукта.  
   - Своевременное выявление и разрешение несоответствий и конфликтов требований.  
   - Обеспечение прозрачности и контроля процесса разработки в части требований.
4. **На какие категории можно разделить все требования при определении их приоритетов?**- Обязательные (Must have) — требования, без которых продукт не может быть принят заказчиком или считаться завершённым.  
   - Желательные (Should have) — важные требования, существенно повышающие качество продукта, но не критичные для его работы.  
   - Опциональные (Could have) — дополнительные функции или улучшения, которые могут быть реализованы при наличии ресурсов и времени.  
   - Отложенные (Won't have) — требования, которые не будут реализовываться в данном релизе, но могут быть рассмотрены в будущем.
5. **В какой очередности следует реализовывать требования к программному продукту?**- В первую очередь реализуются обязательные требования (Must have), обеспечивающие базовую работоспособность системы и соответствие критериям приёмки.  
   - Затем переходят к выполнению желательных требований (Should have), повышающих удобство и эффективность использования.  
   - После этого реализуются опциональные требования (Could have), если позволяют ресурсы и сроки.  
   - Отложенные требования (Won't have) выделяются в резерв и могут учитываться в будущих версиях или обновлениях.
6. **Какие требования относятся к функциональным и нефункциональным?**- Функциональные требования — описывают конкретные действия, функции и поведение системы, которые должен выполнять продукт (например, обработка заказов, авторизация пользователей, формирование отчётов).  
   - Нефункциональные требования — определяют качества и характеристики системы, связанные с её работой, но не с конкретными действиями (например, производительность, безопасность, удобство интерфейса, масштабируемость, надёжность).
7. **Какие действия включает в себя цикл формирования требований к программному продукту?**- Сбор требований — сбор информации от заказчиков и пользователей.  
   - Анализ требований — проверка полноты, корректности, достижимости.  
   - Спецификация требований — формальное документирование в понятной форме.  
   - Валидация требований — проверка соответствия ожиданиям заказчика и пользователям.  
   - Управление требованиями — отслеживание изменений и обновлений требований в процессе разработки.
8. **Какие цели преследует анализ первичных требований заказчика?**- Определение точных и измеримых потребностей заказчика.  
   - Выяснение приоритетов и ограничений (временных, технических, бюджетных).  
   - Поиск неоднозначностей и противоречий в требованиях.  
   - Формирование понимания целей и задач проекта.  
   - Подготовка к последующему структурированию и уточнению требований.
9. **Какие действия выполняются при структурировании требований заказчика?**- Классификация требований по категориям (функциональные, нефункциональные, бизнес-требования и т. д.).  
   - Иерархизация — построение дерева или модели, отражающей взаимосвязи между требованиями.  
   - Разделение сложных требований на более мелкие и детальные.  
   - Выделение ключевых требований и зависимостей между ними.  
   - Подготовка документации для удобства анализа и реализации в проекте.
10. **Какова роль прототипа при формировании требований к программному продукту?** Прототип служит инструментом визуализации и проверки требований на ранних этапах разработки. Он помогает заказчику и разработчикам понять, как будет выглядеть и работать будущая система, выявить недочёты и неточности в требованиях, улучшить коммуникацию и снизить риски неправильного понимания задач.
11. **Какие преимущества обеспечивает прототип?**- Раннее выявление ошибок и недочётов в требованиях.  
    - Улучшение взаимодействия между заказчиком и командой разработки.  
    - Более точное представление о функциональности и интерфейсе системы.  
    - Сокращение времени на доработки в последующих этапах.  
    - Возможность быстро вносить изменения и проверять их влияние.
12. **Почему требования в спецификации требований должны быть однозначными?** Однозначность требований обеспечивает:  
    - Единое понимание задач всеми участниками проекта.  
    - Минимизацию ошибок, связанных с неправильной интерпретацией.  
    - Повышение качества программного продукта за счёт точного выполнения заданных функций.  
    - Упрощение процесса тестирования и валидации.  
    - Эффективное управление изменениями и документооборотом.
13. **Какие метрики собирают на этапе управления требованиями?**- Количество требований – общее число задокументированных требований.  
    - Степень покрытия требований – доля требований, которые реализованы и протестированы.  
    - Процент изменённых требований – количество требований, изменённых за определённый период.  
    - Статус требований – количество требований в различных состояниях (например, новые, утверждённые, реализованные, отклонённые).  
    - Время на обработку требований – среднее время от предложения до утверждения или реализации требования.  
    - Количество конфликтов и дублирующихся требований – показывает качество формулировки и согласования.
14. **Какие методы, стандарты и шаблоны используют на этапе управления требованиями?** Методы:  
    - Трассируемость требований – отслеживание взаимосвязей между требованиями, дизайном, тестами.  
    - Приоритизация требований – методы оценки важности, например MoSCoW (Must, Should, Could, Won't).  
    - Управление изменениями – процессы и процедуры для внесения и утверждения изменений.

Стандарты:  
- IEEE 830/29148 – стандарты описания и управления требованиями.  
- ISO/IEC/IEEE 12207 – процессы жизненного цикла ПО, включая управление требованиями.  
 Шаблоны:  
- Форматы спецификаций требований с четкой структурой (описание, критерии приёма, приоритет).  
- Шаблоны для ведения реестра требований (Requirement Traceability Matrix).  
- Формы для заявок на изменение требований и их анализа.

**Источник:** Рудаков А. В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования / А. В. Рудаков. – 11-е изд., стер. – М.: Издательский центр “Академия”, 2017, cтр. 100-106.