ПРАКТИЧЕСКАЯ РАБОТА №1

Инструкционно-технологическая карта

Тема: «Тестирование требований к ПО»

Цель работы: Знать основные уровни требований, уметь формулировать требования, научиться составлять и тестировать требования.

Время выполнения: 2 часа

1. Теоретические сведения для выполнения работы

Разработка требований

Разработка требований — это процесс, включающий мероприятия, необходимые для созда­ния и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований:

1. анализ технической осуществимости создания системы,
2. формирование и анализ требований,
3. специфицирование требований и создание соответствующей документации,
4. аттестация этих требований.

На рисунке 1 показаны взаимосвязи между этими этапами и результаты, сопровождающие каждый этап процесса разработки системных требований.

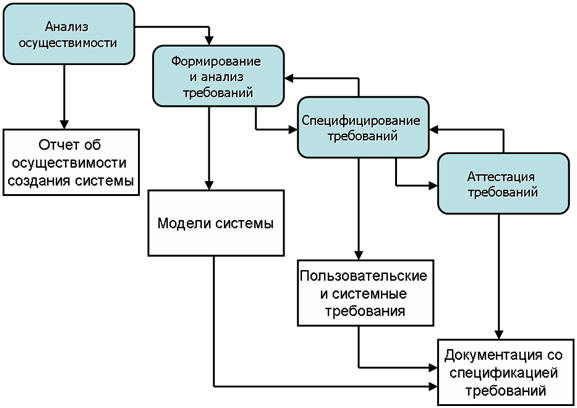
Но поскольку в процессе разработки системы в силу разнообразных причин требова­ния могут меняться, управление требованиями, т.е. процесс управления изменениями системных требований, является необходимой составной частью деятельности по их раз­работке.

Рисунок 1 - Процесс разработки требований

Спецификация требований

Рассмотрим составляющие спецификации требований, рекомендованные стандартом IEEE 830 (структура SRS – Software requirements specification):

**Введение:**

* Цели
* Соглашения о терминах
* Предполагаемая аудитория и последовательность восприятия
* Масштаб проекта
* Ссылки на источники

**Общее описание:**

* Видение продукта
* Функциональность продукта
* Классы и характеристики пользователей
* Среда функционирования продукта (операционная среда)
* Рамки, ограничения, правила и стандарты
* Документация для пользователей
* Допущения и зависимости

**Функциональность системы:**

* Функциональный блок X (таких блоков может быть несколько)
* Описание и приоритет
* Причинно-следственные связи, алгоритмы (движение процессов, workflows)
* Функциональные требования

**Требования к внешним интерфейсам**

* Интерфейсы пользователя (UX)
* Программные интерфейсы
* Интерфейсы оборудования
* Интерфейсы связи и коммуникации

**Нефункциональные требования**

* Требования к производительности
* Требования к сохранности (данных)
* Критерии качества программного обеспечения
* Требования к безопасности системы

**Прочие требования**

* Приложение А: Глоссарий
* Приложение Б: Модели процессов и предметной области и другие диаграммы
* Приложение В: Список ключевых задач

Несмотря на такое большое количество пунктов, которые входят в нашу спецификацию, все же «ядро» требований к приложению находится в функциональных требованиях. Зачастую, их представляют в виде пользовательских требований, выполненных в виде пользовательских сценариев (Use cases).

 Пользовательские требования

*Пользовательские требования* должны описывать внешнее поведение системы, основные функции и сервисы предоставляемые системой, её нефункциональные свойства. Необходимо выделить опорные точки зрения и сгруппировать требования в соответствии с ними. Пользовательские требования можно оформить как простым перечислением, так и используя нотацию вариантов использования.

**Use case описывает то, как действующее лицо пытается достичь некой цели, используя систему.**

Сценарий или спецификация ВИ (use case scenario or specification) – тестовое формальное описание последовательности действий, которые происходят внутри ВИ для достижения некой цели актера.

Шаблон полного описания варианта использования по А. Коберну

**Название** <краткая фраза в виде глагола в неопределённой форме совершенного вида, отражающая цель>

**Контекст использования** <уточнение цели, при необходимости – условия её нормального завершения>.

**Область действия** <ссылка на рамки проекта>. Например – подсистема бухгалтерского учёта.

**Уровень** <один из трёх: обобщённый, цели пользователя, подфункции>. Автор задаёт предопределённую трёхуровневую классификацию требований, в целом соответствующую классификации требований на бизнес-требования, требования пользователей и функциональные требования.

**Основное действующее лицо** <имя роли основного актора или его описание>.

**Участники и интересы** <список других акторов-участников прецедента с указанием их интересов>.

**Предусловие** <то, что ожидается, уже имеет место>.

**Минимальные гарантии** <что гарантируется акторам-участникам>. Например – в случае неудавшейся транзакции все данные, имевшиеся в системе до её начала, сохраняются неизменными.

**Гарантии успеха** <что получат акторы-участники в случае успешного достижения цели>.

**Триггер** <то, что «запускает» вариант использования, обычно – событие во времени>.

**Основной сценарий** <здесь перечисляются шаги основного сценария, начиная от триггера и вплоть до достижения гарантии успеха>.

Формат описания: <Номер шага> <Описание действия>

**Расширения** <здесь последовательно описываются все альтернативные сценарии>. Каждая из альтернатив привязана к шагу основного сценария.

Формат описания: <Номер шага.Номер расширения> <Условие>:<Действие или ссылка на подчинённый вариант использования>.

Любой из шагов основного сценария может иметь 1 или более ветвлений. Каждое ветвление оформляется в виде расширения. В блоке «Расширения» все расширения описываются последовательно.

В случае, если альтернативный сценарий не удаётся описать одной строкой – применяется следующий формат.

Начиная со строки, следующей после описания расширения, идёт описание его действий в формате основного сценария:

<Номер шага.Номер расширения.Номер шага расширения> <Действие>

Описание расширения заканчивается описанием выхода из расширения. Основные варианты выхода из расширения: возврат к очередному по номеру шагу основного сценария, окончание прецедента, переход к другому шагу основного сценария.

Обычно описание варианта использования оформляют табличным способом и он включает следующие категории:

1. уникальный идентификатор;
2. цель, кратко описывающая задачи пользователи в формате «глагол + объект», например «разместить заказ»;
3. действующие лица (участники);
4. триггер (с чего начинается ВИ)
5. список предварительных условий, которые должны быть удовлетворены до начала разработки варианта использования;
6. основной сценарий;
7. альтернативный сценарий (расширения);
8. вспомогательная информация (ограничения)

Пример #1:

|  |  |
| --- | --- |
| **ID** | **2 UC** |
| **Цель** | **Вступить в сообщество** |
| **Участники** | Зарегистрированные пользователи ВК |
| **Предусловие** | Пользователь должен быть зарегистрирован в социальной сети ВК |
| **Триггер** | Пользователь нажимает категорию меню "Мои сообщества" |
| **Основной сценарий** | 1. Пользователь осуществляет поиск сообщества (**1 UС**).  **2.** Пользователь осуществляет подписку на сообщество кликнув кнопку "Подписаться"  3. Система оповещает пользователя о вступлении в сообщество сообщением "Вы подписаны" |
| **Альтернативный сценарий** | **2А.** Пользователь переходит на страницу сообщества.  **2Б**. Пользователь кликает на кнопку "Подписаться" либо "Вступить в группу".  3А. Система оповещает пользователя о вступлении в сообщество одним из следующих сообщений "Вы подписаны на новости", "Вы состоите в группе". |
| **Расширения** | 3А\_И1. Сообщество является закрытым  3А\_И1-1. Пользователь переходит на страницу сообщества. 3А\_И1-**2.** Пользователь подает запрос на вступление в сообщество (кликает на кнопку "Подать заявку")  3А\_И1-**2а**. Система отправляет заявку пользователя администратору сообщества.  3А\_И1-**2б**. Администратор рассматривает поданную заявку и принимает ее либо отклоняет. |

Пример #2:

|  |  |
| --- | --- |
| **ID** | **К1.1** |
| **Цель** | Выполнить вход в систему |
| **Участники** | Пользователь |
| **Предусловие** | Пользователь загружает систему |
| **Триггер** | Окно входа в систему |
| **Основной сценарий** | 1. Пользователь вводит логин и пароль.  **2.** Пользователь запускает проверку  3. Система проверяет логин  4. Система проверяет пароль  5. Система предоставляет пользователю доступ |
| **Альтернативный сценарий** | **3А. Не найдена учетная запись с таким логином**  **3А.1** Система уведомляет об ошибке  **3А**.**2** Возврат сценария на пункт 1.  **4А.Пароль неверный**  **4А.1** Система увеличивает счетчик неудачных попыток входа.  4А.2. Система проверяет количество неудачных попыток входа |
| **Расширения** | 4А.1.а. Количество неудачных попыток больше установленного предела:  4А.1.а.1. Система уведомляет о блокировке  4А.1.а.2. Завершение сценария  4А.3. Система уведомляет об ошибке  4А.4. Возврат сценария на пункт 1 |

Пример #3:

|  |  |
| --- | --- |
| **ID** | **UC 3.** |
| **Цель** | Зарегистрироваться на сайте |
| **Участники** | Пользователь сайта |
| **Предусловие** | Пользователь находится на главной странице сайта |
| **Триггер** | Пользователь нажимает кнопку «Зарегистрироваться» |
| **Основной сценарий** | 1. Сайт отображает форму регистрации  **2.** Пользователь заполняет поля формы и подтверждает регистрацию.  3. Сайт подтверждает правильность заполнения формы  4. Сайт регистрирует пользователя и отправляет на его почтовый адрес письмо с подтверждением регистрации |
| **Альтернативный сценарий** | **3А. Есть ошибки заполнения формы**  **3А.1** Система уведомляет об ошибке  **3А**.**2** Возврат на шаг 2.  **3Б. Пользователь с таким почтовым адресом зарегистрирован**  **3Б.1** Сайт сообщает о том, что пользователь с таким почтовым ящиком уже зарегистрирован  **3Б.2**. Переход на шаг 2. |

**Не стоит использовать сценарий Use Cases:**

* Когда требуется описать сложный алгоритм
* Для описания пользовательского интерфейса

**Несколько рекомендаций по созданию сценария Use Cases:**

* Основной сценарий не больше 3 - 9 шагов
* Исключить элементы дизайна (если в них есть необходимость опишите в другом месте)
* Использовать один уровень детализации на всех шагах
* Избегать формализма (используйте тот шаблон, который необходим в данный момент )
* Не пытаться описать все расширения сразу
* Не использовать слово «если»

1. Порядок выполнения работы:
2. Изучить предлагаемый теоретический материал.
3. Разработать спецификацию требований к программе, разработанной ранее в рамках курсовой работы по дисциплине «Основы алгоритмизации и программирование» по аналогии с документами, приведенных в папке с практической работой.
4. Разработать пользовательские требования (в виде Use Cases) для основных функций к программе, разработанной ранее в рамках курсовой работы по дисциплине «Основы алгоритмизации и программирование».
5. Знать ответы на контрольные вопросы.
6. Подготовить отчет по практической работе.
7. Контрольные вопросы:
8. Дайте определение понятию «требование».
9. Перечислить основные уровни требований.
10. Что собой представляют нефункциональные требования?
11. Перечислите характеристики качественных требований.
12. Что такое «Вариант использования»?
13. Перечислить элементы описания «Вариантов использования».