

Лабораторная работа 2. Требования к программному обеспечению

Цель работы: овладеть навыками определения требований к программному обеспечению, руководствуясь действующими государственными стандартами, а также с использованием метода прецедентов унифицированного процесса разработки ПО.

Теоретические сведения

Правильное определение требований к программному обеспечению является важнейшим критерием успешной разработки.

Общие понятия

Требования Соммервил в [4] описывал следующим образом:

Требования — это возможности или условия, которым должна соответствовать система или проект.

Требования принято разделять по уровням:

1. Пользовательские требования — описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на нее.
2. Системные требования — детализированное описание системных функций и ограничений, которое иногда называют функциональной спецификацией. Она служит основой для заключения контракта между покупателем системы и разработчиками ПО.
3. Проектная системная спецификация — обобщенное описание структуры программной системы, которое будет основой для более детализированного проектирования системы и её последующей реализации. Эта спецификация дополняет и детализирует спецификацию системных требований.

А также классифицировать как:

1. Функциональные требования — это перечень функций, которые должна выполнять система, причем должно быть указано, как система реагирует на те или иные входные данные, как

она ведет себя в определенных ситуациях и т.д. В некоторых случаях указывается, что система не должна делать

2. Нефункциональные требования — описывают характеристики системы и её окружения, а не поведение системы. Здесь также может быть приведен перечень ограничений, накладываемых на действия и функции, выполняемые системой. Они включают временные ограничения, ограничения на процесс разработки системы, стандарты и т.д.
3. Требования предметной области — характеризуют ту предметную область, где будет эксплуатироваться система. Эти требования могут быть функциональными и нефункциональными.

Примеры требований

Приведем несколько примеров описания требований. Предположим, что нам необходимо написать программу для работы с файлами.

Пользовательские требования можно записать следующим образом:

1. ПО должно предоставить средство доступа к внешним файлам созданным в других программах.

В свою очередь системные требования будут выглядеть более подробно:

1. Пользователь должен иметь возможность определять тип внешних файлов;
2. Для типа внешнего файла должно иметься соответствующее средство, применимое к этому типу файлов.

Описание требований в контексте модели прецедентов

Описание требований удобно производить в контексте модели прецедентов. Что же такое прецеденты. В [3] дается следующее неформальное определение:

Прецеденты — это механизм упрощения этапа формулировки требований для всех заинтересованных лиц. По

существо это рассказы об использовании системы в процессе решения поставленных задач.

Для более полного понимания модели прецедентов запишем еще несколько определений (более подробно они описаны в [3]):

Исполнителем будем называть сущность, обладающую поведением, например, человека, компьютерную систему или организацию.

Сценарий — это специальная последовательность действий или взаимодействий между исполнителями и системой. Его иногда также называют экземпляром прецедента. Это один конкретный сценарий использования системы либо один проход прецедента, например, сценарий успешной покупки товара за наличный расчет, либо сценарий неудачного завершения покупки из-за прерванной транзакции по обработке данных кредитной карточки.

При описании прецедентов необходимо иметь в виду следующее: каждый прецедент должен приводить к осязаемому результату для его исполнителя. Таким образом при описании прецедентов необходимо постоянно задаваться вопросом «Как использование системы обеспечивает осязаемый для пользователя результат или решает его задачу?».

Примеры описания прецедентов

После описания основных определений приведем пример прецедента «Возврат товара», записанного в свободной форме:

Основной успешный сценарий.

Покупатель подходит к кассе с товарами, подлежащими возврату. Кассир использует POS-систему для регистрации каждого возвращаемого товара, после регистрации каждого товара кассир печатает квитанции о возврате, возвращает деньги покупателю.

Альтернативные сценарии.

Если в авторизации кредитной карточки отказано, кассир информирует об этом покупателя и предлагает ему другой способ оплаты покупки.

Если идентификатор товара в системе не обнаружен, система уведомляет об этом кассира и предлагает ему вручную ввести идентификационный код (возможно, штрих-код, и его сложно считать).

Если у системы возникают сложности при коммуникации с внешней системой вычисления налога и т.д.

Приведем пример описания одного из прецедентов для «Системы агрегирования информации из различных источников».

Прецедент «Создание источника».

Предусловия: Вход в систему

Основной успешный сценарий.

Пользователь открывает форму создания источника, выбирает его формат, вводит название, а также URL-адреса, по которым он будет доступен, сохраняет источник. Система показывает пользователю сообщение о начале сохранения источника. Система отвечает сообщением об удачном сохранении источника.

Альтернативные сценарии.

При создании, не все поля были заполнены и система информирует об этом пользователя.

Пользователем были указаны неверные адреса источника (источник недоступен), система уведомляет об этом пользователя.

Для описания сценариев в модели прецедентов обычно применяется подход «Черного ящика», то есть мы не раскрываем в прецедентах внутренние механизмы работы системы, используя только её внешние интерфейсы.

Задание к выполнению работы

В соответствии со своим вариантом выполнить следующие действия:

1. Описать требования к выбранному программному продукту в виде технического задания согласно ГОСТ 19.201-78.
2. Описать 3—4 прецедента, покрывающие основные функции описываемого программного продукта, включая альтернативные потоки. Использовать развёрнутый формат описания прецедентов.
3. Начертить диаграмму описанных прецедентов.
4. Оформить и распечатать отчёт по работе, включающий название, цель работы, а также результаты выполнения вышеприведённых пунктов.
5. Подготовить ответы на вопросы по материалам соответствующих лекций.

Варианты заданий

1. Сниффер;
2. Прокси-сервер;
3. Фаервол;
4. Система администрирования маршрутизатора;
5. Банковская система, основанная на методах слепой подписи;
6. Система мониторинга состояния сети;
7. Система управления защищенными паролями;
8. Keylogger;
9. Собственная тема.